INTERACTIVE CONTROL ENGINEERING

Practical PID Control

Interactive Control Engineering: Principles and Applications

Edited by John D. Hedengren

Control technology

1 Process Dynamics, Modeling, and Identification
2 Practical PID Control
3 Feedforward, Cascade, and Ratio Control
4 Override, Split-Range, and Constraint Control
5 Delays, Slow Measurements, and State Estimation
6 Multivariable and Plantwide Control
7 Nonlinear Dynamics, Oscillations, and Stabilization
8 Batch, Run-to-Run, and Hybrid Control
9 Model Predictive Control and Economic Optimization

Industry applications

10 Control in Upstream and Midstream Oil and Gas
11 Control in Petroleum Refining
12 Control in Petrochemicals and Polymers
13 Control in Chemical Plants
14 Control in Semiconductor Manufacturing
15 Control in Mining and Mineral Processing
16 Control in Iron, Steel, and Metals Production
17 Control of Energy Conversion and Storage
18 Control of Nuclear and Fusion Systems
19 Control of Biological Systems and Human Performance
20 Motion Control, Robotics, and Autonomous Systems

INTERACTIVE CONTROL ENGINEERING
Principles and Applications

Volume 2

Practical PID Control

Tuning, anti-windup and implementation that works on real loops

John D. Hedengren
Editor

Advanced Process Solutions, LLC

Interactive Control Engineering: Principles and Applications
Volume 2: Practical PID Control
Edited by John D. Hedengren

First edition, 2026

Copyright © 2026 Advanced Process Solutions, LLC. All rights reserved.

Produced by Advanced Process Solutions, LLC.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the prior written permission of the publisher, except for brief quotations in a review or scholarly work.

The simulators described in this book are teaching models. Their equations, parameters and results are simplified for instruction and are not validated for the design, operation or safety assessment of any real plant, vehicle or medical device. The editor and the publisher accept no liability for use of this material outside that purpose.

Names of products and companies are used for identification only and may be trademarks of their owners.

Every chapter has a companion simulator that runs in a web browser at https://apmonitor.com/apps/. This volume online, with chapter files and citation records: https://apmonitor.com/apps/books/v02-practical-pid/

How to cite this volume. John D. Hedengren, Ed., *Practical PID Control*, vol. 2 of *Interactive Control Engineering: Principles and Applications*. Advanced Process Solutions, LLC, 2026.

Typeset in Times and Helvetica-family type with LaTeX and the memoir class.

Preface

This book is one of 20 volumes of *Interactive Control Engineering: Principles and Applications*. Each chapter is about one plant, machine or system and one control lesson, and each has a simulator that runs in a web browser with nothing to install. The simulators are collected at https://apmonitor.com/apps/. Nine volumes are organized by control technology and eleven by industry or application area; the last pages of this volume list them all.

How to read this book with the apps

Chapter 1 teaches the theory this volume relies on and explains, once, the simulator framework that the apps share. Each chapter after it opens with the control question it answers, its learning objectives, what to read first, and the address of its app, also given as a code to scan. The chapter then develops the model, the controllers and the measured results. Every number reported was measured on the app, so a reader can open the app, load the named scenario and reproduce it.

Every chapter ends with exercises. Each is labeled with its kind and the time it should take, and none takes more than ten minutes. Concept checks and calculations can be done on paper. App challenges name a scenario, the settings and the run length. One exercise in each chapter, labeled *Review the agent*, presents a tuning, a model or a claim proposed by an AI assistant and asks what is wrong with it. Short answers to the odd-numbered exercises are at the back of the book, and worked solutions are available to instructors.

How AI agents were used

AI agents did much of the development work behind this series. Under the editor's direction they wrote and tested the simulators, drafted the papers from which the chapters are drawn, measured the reported results on the running apps, converted the papers to book form, and drafted the introductory and closing chapters and the exercises. Other agents then checked that work: they re-measured reported numbers, audited what each app displays against the state of its model, reviewed the papers as referees, and solved the exercises without seeing the answers. The editor set the scope, chose the plants and the lessons, reviewed the results and is responsible for the content.

That way of working is also a subject of the book. The last chapter of each volume describes what the agents got wrong while this volume was made and how the errors were caught, and it sets out what the editor believes this means for engineers: more of the routine work can be delegated, and more of the engineer's value lies in checking, supervising, choosing the problem and knowing the process.

About this volume

This volume is about the controller found in most industrial loops and the details that decide whether it works. Chapters 2 to 6 tune loops on process and vehicle plants: a cruise control whose gain moves with speed, a blending tank that meets its actuator limit, a heat exchanger identified from a step test, an integrating makeup loop and a pH loop that needs a scheduled gain. Chapters 7 to 11 apply the same loop to networks, traffic, markets and economic policy, where dead time, sampling and quantization set the limit. Chapter 12 commissions a servo axis whose mechanical resonance caps the gain.

A reader who is new to process models should read Volume 1 first. Plants that are unstable without control are the subject of Volume 7.

John D. Hedengren
2026

Contents

Preface v

Notation x

1 Introduction: PID Control in Practice 1

1.1 The Question This Volume Answers . . . 1
1.2 A Short History . . . 2
1.3 Notation and Conventions . . . 3
1.4 PID Theory on One Loop . . . 3
1.5 Instruments and Final Elements in This Volume . . . 13
1.6 The Simulator Framework . . . 14
1.7 Working With AI Agents While Studying . . . 18
1.8 Chapter Roadmap . . . 19
1.9 Exercises . . . 22

2 Automobile Cruise Control Simulator for Velocity-Form PID Under Quadratic Drag 24

2.1 Introduction . . . 25
2.2 Background . . . 26
2.3 Vehicle Model . . . 27
2.4 Dynamic Analysis . . . 31
2.5 Controller Design . . . 34
2.6 Results . . . 38
2.7 The Predictive Controller on the Shipped Cases . . . 43
2.8 Estimator Choice and Relay Tuning . . . 45
2.9 Technology in Practice . . . 46
2.10 Limitations . . . 48
2.11 Discussion . . . 49
2.12 Conclusion . . . 49
2.13 Exercises . . . 51

3 Stirred Blending Tank Simulator for Composition Control Against an Actuator Limit 55

3.1 Introduction . . . 56
3.2 Background . . . 57
3.3 Process Model . . . 58
3.4 Dynamic Analysis . . . 62
3.5 Controller Design . . . 64
3.6 Setpoint Reachability . . . 68
3.7 Results . . . 69
3.8 Technology in Practice . . . 78
3.9 Limitations . . . 80
3.10 Discussion . . . 81
3.11 Conclusion . . . 81
3.12 Exercises . . . 84

4 Steam-Heated Exchanger Laboratory with Hidden Parameters for Step Testing and Tuning 87

4.1 Introduction . . . 88
4.2 Background . . . 89

4.3 Process Model . . . 90
4.4 Identification from Step-Response Data . . . 95
4.5 Controller Design . . . 96
4.6 Dynamic Analysis . . . 101
4.7 Results . . . 101
4.8 Technology in Practice . . . 109
4.9 Limitations . . . 112
4.10 Discussion . . . 113
4.11 Conclusion . . . 113
4.12 Exercises . . . 116

5 Cooling Tower Makeup Simulator for Identification and Tuning of Integrating Loops 119
5.1 Introduction . . . 120
5.2 Background . . . 121
5.3 Process Model . . . 122
5.4 Dynamic Analysis . . . 126
5.5 Controller Design . . . 128
5.6 Results . . . 132
5.7 Technology in Practice . . . 138
5.8 Limitations . . . 140
5.9 Discussion . . . 141
5.10 Conclusion . . . 142
5.11 Exercises . . . 144

6 pH Neutralization Virtual Laboratory for Gain Scheduling and Measurement Linearization 147
6.1 Introduction . . . 148
6.2 A History of pH Control . . . 149
6.3 Process Model . . . 150
6.4 The Gain That Is Not a Number . . . 154
6.5 Controller Design . . . 156
6.6 Results . . . 157
6.7 Technology in Practice . . . 162
6.8 Limitations . . . 164
6.9 Discussion . . . 165
6.10 Conclusion . . . 166
6.11 Exercises . . . 167

7 Delay Differential Equations and the Stability Boundary in a TCP Congestion Simulator 171
7.1 Introduction . . . 172
7.2 Background . . . 173
7.3 Network Model . . . 174
7.4 Linearization and the Characteristic Equation . . . 178
7.5 Controller Design . . . 180
7.6 Numerical Solution of the Delay Equations . . . 181
7.7 Scenarios . . . 182
7.8 Results . . . 183
7.9 Technology in Practice . . . 188
7.10 Limitations . . . 190
7.11 Discussion . . . 191
7.12 Conclusion . . . 191
7.13 Exercises . . . 193

8 Ramp Metering Simulator for Holding a Freeway Merge Short of Breakdown 196
8.1 Introduction . . . 197
8.2 From Kinematic Waves to Ramp Meters . . . 198
8.3 The Freeway and Its Model . . . 199
8.4 The Meters as Implemented . . . 204
8.5 Ten Mornings and a Challenge . . . 207
8.6 Technology in Practice . . . 216
8.7 Limitations . . . 218
8.8 Conclusion . . . 219
8.9 Exercises . . . 220

9 Surge Pricing as Feedback on a Ride-Hailing Market With a Throughput Fold 224
9.1 Introduction . . . 225
9.2 From the Dispatched Taxi to the Surge Multiplier . . . 226
9.3 A Zone of Three Pools . . . 227
9.4 Two Speeds and a Fold . . . 231
9.5 Pricing on a Calendar . . . 233
9.6 Eleven Runs of One Zone . . . 235
9.7 Technology in Practice . . . 240
9.8 Limitations . . . 242
9.9 Conclusion . . . 242
9.10 Exercises . . . 244

10 Phillips's Stabilization Policy, Budget Lags and a Folding Public Debt 247
10.1 Introduction . . . 248
10.2 From Water Tanks to Debt Brakes . . . 249
10.3 The Machine, Its Budget Pipelines and the Debt . . . 250
10.4 What the Economy Does Before Anyone Decides . . . 254
10.5 Six Ways to Set the Budget . . . 255
10.6 Ten Cases on One Machine . . . 258
10.7 Technology in Practice . . . 264
10.8 Limitations . . . 266
10.9 Conclusion . . . 267
10.10 Exercises . . . 268

11 Inflation Targeting as Sampled, Quantized Control of a Lagged Economy 272
11.1 Introduction . . . 273
11.2 Stabilization Policy as Feedback, 1954 to Forecast Targeting . . . 274
11.3 An Economy of Stocks and Lags . . . 275
11.4 How the Economy Answers a Move . . . 278
11.5 How the Bank Decides . . . 280
11.6 Measured Runs of the Economy . . . 283
11.7 Technology in Practice . . . 288
11.8 Limitations . . . 291
11.9 Conclusion . . . 291
11.10 Exercises . . . 293

12 Commissioning a Servo Axis Whose Coupling Resonance Caps the Speed Loop 297
12.1 Introduction . . . 298
12.2 How the Servo Axis Acquired Its Cascade . . . 299
12.3 The Axis From Windings to Table . . . 300
12.4 The Drive as Implemented . . . 304

12.5 Ten Cases and a Move and Return . . . 310
12.6 Technology in Practice . . . 320
12.7 Limitations . . . 321
12.8 Conclusion . . . 322
12.9 Exercises . . . 324

13 Summary and Outlook **328**
13.1 What the Chapters Showed . . . 328
13.2 Choosing a Method . . . 331
13.3 Where Practice Is Heading . . . 332
13.4 Engineering With AI Agents . . . 333
13.5 Gaps in PID Practice . . . 336
13.6 Exercises . . . 340

Answers to Selected Exercises **342**

Glossary **347**

Index of Control Topics **350**

Index of Applications **353**

About the Editor **356**

Volumes in This Series **357**

Notation

The chapters share the symbols below. A symbol used by one chapter only is defined where it first appears, and each chapter states its own units.

Symbol	Meaning
SP	setpoint, the value a controller is asked to hold
PV	process variable, the measurement a controller acts on
CV	controlled variable
MV	manipulated variable; also OP, the controller output
DV	disturbance variable
e	control error, SP $-$ PV
K_c	controller gain
τ_I, τ_D	integral (reset) time and derivative time
K_p	process gain, the steady-state change in PV per unit change in MV
τ_p	process time constant
θ_p	process dead time
τ_c	closed-loop time constant chosen in a tuning rule
Δt	controller sample time
s	Laplace variable
x, u, y, d	state, input, output and disturbance of a state-space model
A, B, C	matrices of the linear model $\dot{x} = Ax + Bu$, $y = Cx$
Q, R	weights on outputs (or states) and on input moves in an optimal controller
$\hat{x}$	an estimate of x

Abbreviations used throughout: PID, proportional-integral-derivative; PI, proportional-integral; FOPDT, first-order plus dead time; MPC, model predictive control; MHE, moving horizon estimation; IAE, integral of the absolute error; RGA, relative gain array. Times are in seconds unless a chapter says otherwise. Controller output and valve position are in percent of span. A scenario or case number refers to the numbered list in the chapter's app.

CHAPTER 1

Introduction: PID Control in Practice



The chapters 2 cruise control 3 blending tank 4 steam heat exchanger 5 cooling tower 6 pH neutralization 7 router queue 8 ramp meter 9 ride-hailing price 10 budget and debt 11 inflation targeting 12 servo axis

Notice that the five units on the left are machines and process plant with valves and probes, while the next five are a network, a road, a market and two public institutions, and the servo at the right closes the row with a machine again. Illustration, not to scale.

IN PLAIN TERMS **One Simple Rule for Many Kinds of Loops**

The main idea. A feedback controller compares what it measures with what it wants and moves a handle to close the gap. The common rule adds three parts: how big the gap is now, how long it has lasted, and how fast it is changing.

The challenge. On a real plant the handle stops at its end, the reading is noisy or late, and the controller may act only once a minute or once a year. The plant can also respond more strongly on one day than on another.

Why it matters. The same rule runs car speed, tanks, internet routers, freeway ramps, prices and interest rates, so setting it well pays almost everywhere.

THREE WAYS TO READ THIS CHAPTER

Explore	7 min	This page, Section 1.2 with its timeline and two History boxes, the Did you know? boxes in Sections 1.1, 1.4.3 and 1.5, Figure 1.11, and Key points in Section 1.8. No equations.
Study	45 min	The whole chapter in order, with each number of the oven example worked, then Exercises 1 to 10.
Practice	10 min	This page, Section 1.4.5 with Table 1.1, Section 1.4.9, Section 1.5 with Figures 1.7 and 1.9, and Section 1.8.

1.1 The Question This Volume Answers

The proportional-integral-derivative (PID) controller is three terms and three numbers, and it is still the algorithm behind most industrial feedback loops [1]. The equation is rarely what makes a loop fail. A loop fails because its actuator reaches a limit, its measurement is noisy or late, its controller runs once a minute, or its process gain at today's operating point is not the gain at which it was tuned. This volume answers one question: how is a PID loop built and tuned so that it works on a real plant, with limits, noise, dead time, sampling and a gain that moves?

Eleven chapters answer it on eleven plants, each with an app that the reader runs in a browser. Chapters 2 to 6 are process and vehicle loops: a cruise control, a blending tank, a steam-heated exchanger, a cooling tower basin and a pH neutralizer. Chapters 7 to 11 carry the same algorithm outside the plant, to a network router, a freeway ramp, a ride-hailing market, a national budget and a central bank, where the controller acts on a calendar and in fixed steps. Chapter 12 closes the volume on a servo axis whose three nested loops run at three rates.

A reader who has worked through the volume can do the following on a loop that is new to the reader.

- Read a vendor's PID equation, name its form and convert its settings to another form.
- Choose the controller action from the sign of the process gain.

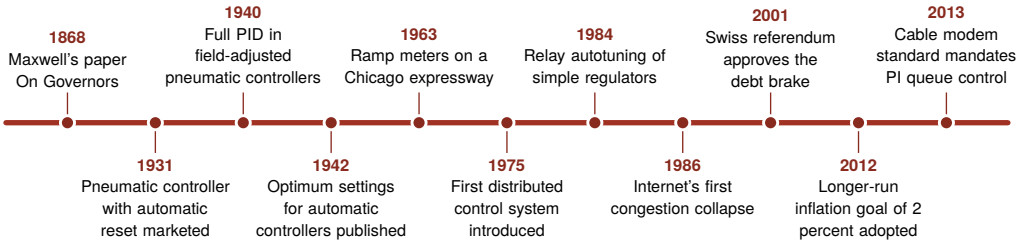


Figure 1.1 Milestones of PID control in practice. The dates are from [6] (1868), [7] (1931, 1940), [8] (1942), [9] (1963), [10] (1975), [11] (1984), [12] (1986), [13] (2001), [14] (2012) and [15] (2013). Notice that the three terms were settled in the process plant before 1942 and that the same feedback reached roads, networks and public policy later. Events are evenly spaced, not to a time scale.

- Fit a first-order or an integrating model with dead time to a step test, and tune from the fit with a named rule.
- Run a relay test and say what its result does and does not measure.
- Filter the derivative, protect the integral against windup and transfer between Manual and Auto without a bump.
- Say what a sample time, a dead time or a moving gain costs in stability margin, and decide when to schedule the controller gain.

This chapter develops each of these on one small example and names the chapter that uses it. Standard texts give fuller derivations [2–4].

> ✦ DID YOU KNOW? **A Hundred-Billion-Dollar Industry**
>
> A standard text of 2021 puts control and automation at a 100 billion dollar industry. Distributed control systems make up about 20 % of the market, and sensors, actuators, software and other components make up the rest [5].

1.2 A Short History

The milestones of Figure 1.1 follow one algorithm from the engine to the economy. In 1868 Maxwell described the governor as the part of a machine that keeps its velocity nearly uniform despite changes in the driving power or the resistance [6], the problem of Chapter 2.

Process plants took the three terms one at a time. A pneumatic controller with automatic reset, the action now called integral, was marketed in 1931, and derivative action, then called pre-act, was found the same year. By 1940 full PID was available in pneumatic controllers adjusted in the field [7]. A paper of 1942 gave practical ways to set the three knobs [7, 8], and pneumatic control then became the standard of the oil, gas and chemical industries [7].

The same feedback then left the plant. In 1963 ramp meters were set up on a Chicago expressway and run by hand by an officer in the field [9]. Chapter 8 closes that loop on detector reports. The first distributed control system was introduced in 1975 [10]. A relay test that tunes a simple regulator by itself was published in 1984 [11], and tuning from a model followed in 1986 [16] and in simple analytic rules in 2003 [17]. In October 1986 the Internet had the first of a series of congestion collapses [12], and in October 2013 a cable modem standard required a variant of a PI queue controller [15], the loop of Chapter 7.

> ⧖ HISTORY **By 1940: Three Knobs on a Pneumatic Controller**
>
> An early pneumatic proportional controller with a wide band was set with a gain low enough to limit hunting but high enough to keep the steady error small, and the setpoint was reset by hand so that the process stood on target despite the offset. A circuit that copied that manual correction, called automatic reset, is the integral action of today. Derivative action, then called pre-act, began as an accident: a capillary put into a feedback line by mistake gave the response a kick. By 1940 all three actions were available in pneumatic controllers adjusted in the field, three knobs turned largely by trial and error [7]. The offset of Equation (1.3) and the kick of Section 1.4.8 are the same two effects.

Public policy keeps a calendar of its own. Swiss voters approved a debt brake in a referendum of 2001, and it first applied to the federal budget of 2003 [13]. In January 2012 the monetary policy committee of the United States stated that inflation of 2 percent is most consistent with its mandate over the longer run [14]. Each rule gives a policy a published number to steer toward, the setpoint that Chapter 10 and Chapter 11 treat as the target of a loop.

> ⧖ HISTORY **1962 to 1975: From Relay Cabinets to Distributed Control**
>
> As instruments multiplied, bundles of pneumatic tubing ran into the control room. Wired transmitters replaced them, first with a signal of 10 to 50 mA and then, after the transistor, with the 4 to 20 mA range of the same ratio. A conference of 1962 discussed putting every action under the direct digital control of one computer, but process plants held back, with a large base of pneumatic and analog equipment and a fear that the one computer would fail. In 1968 the motor industry asked for a programmable replacement for its cabinets of clicking relays, and in 1975 the first distributed control system was introduced, still adjusting independent controllers [10]. Each loop of such a system is a digital PID that executes once per period, the sampled controller of Section 1.4.11.

1.3 Notation and Conventions

The chapters share the symbols of the notation page. The setpoint is SP, the process variable is PV and the manipulated variable is MV. In equations the measurement is y, the controller output is u and the error is $e = \mathrm{SP} - \mathrm{PV}$. The controller settings are the gain K_c, the integral time τ_I and the derivative time τ_D. The process model has a gain K_p, a time constant τ_p and a dead time θ_p. The closed-loop time constant chosen in a tuning rule is τ_c, and the control period is Δt.

A gain carries units. The process gain K_p is in PV units per MV unit, and the controller gain K_c is in MV units per PV unit, so the product $K_c K_p$ is a pure number. The MV is in percent of span unless a chapter says otherwise. Each chapter keeps the time unit of its plant: seconds for the cruise control and the router, milliseconds for the servo axis and minutes for the process units and the ramp meter. The ride-hailing zone runs in hours and the two economies in years. The running example of this chapter uses minutes, and frequencies are in radians per minute.

Three symbols need care across chapters. This chapter writes K_u and P_u for the ultimate gain and period, K' for the gain of an integrating process and $K_i = K_c/\tau_I$ for an integral gain. Chapter 2 uses K_u for the process gain from pedal to speed, and Chapter 5 writes K_I for the integrating gain of the basin. Each chapter defines its own symbols where they first appear.

1.4 PID Theory on One Loop

1.4.1 The Loop and the Running Example

The running example is a small electric oven. The MV is the heater power u in percent and the PV is the oven temperature y in °C. At rest the temperature is $y = 20 + 2u$, so the power range of 0 to 100 % covers 20 to 220 °C. The oven operates at $u = 50\,\%$ and $y = 120\,^{\circ}\mathrm{C}$. Its response to a change in power is first order with dead time, as shown in Equation (1.1).

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K_p\, \mathrm{e}^{-\theta_p s}}{\tau_p s + 1}, \qquad K_p = 2\,^{\circ}\mathrm{C}/\%, \quad \tau_p = 10\,\mathrm{min}, \quad \theta_p = 2\,\mathrm{min} \tag{1.1}$$

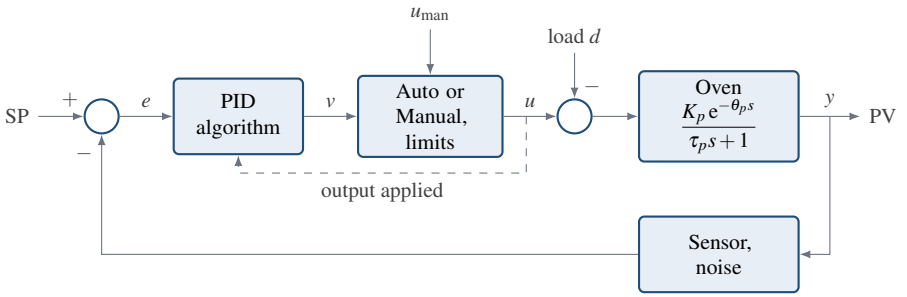


Figure 1.2 The Loop of the Running Example. The Dashed Signal Returns the Applied Output to the Algorithm for Anti-Windup and Tracking.

where Y and U are the transforms of the deviations of y and u from the operating point. Figure 1.2 shows the loop. A load d, such as a door left ajar, acts as a loss of heater power. The limits act on the output v that the algorithm computes, and the algorithm is told the output u that was applied. The controller executes every $\Delta t = 0.1$ min unless stated.

1.4.2 Three Forms of One Algorithm

Vendors write the same three terms in three ways. The ideal form, also called the standard or noninteracting form, is Equation (1.2).

$$u(t) = u_b + K_c \left[e(t) + \frac{1}{\tau_I} \int_0^t e\,dt' + \tau_D \frac{de}{dt} \right] \tag{1.2}$$

where u_b is the output bias. Most apps of this volume take their settings in this form.

Each term has one job. The proportional term acts on the present error, and alone it leaves an offset. After a setpoint step ΔSP the oven settles where the output change $K_c e$ produces the temperature change $K_p K_c e$, which gives Equation (1.3).

$$e_\infty = \frac{\Delta \text{SP}}{1 + K_c K_p} \tag{1.3}$$

With $K_c = 1.25\,\%/^\circ\text{C}$ the offset is $1/3.5 = 29\,\%$ of the step. The integral term keeps moving the output until the error is zero, and the derivative term opposes fast changes. Chapter 3 sets this offset beside one that a limit imposes, and Chapter 10 finds it in an economy's built-in stabilizers. The parallel form of Equation (1.4) gives each term its own gain.

$$u(t) = u_b + K_c\, e(t) + K_i \int_0^t e\,dt' + K_d \frac{de}{dt}, \qquad K_i = \frac{K_c}{\tau_I}, \quad K_d = K_c\, \tau_D \tag{1.4}$$

The ramp meter of Chapter 8 is set by such direct gains, and Chapter 2 writes its closed loop with K_i. The series form, also called interacting, multiplies a proportional-integral (PI) part by a derivative lead, as shown in Equation (1.5).

$$C(s) = K_c' \left(1 + \frac{1}{\tau_I' s} \right) \left(\tau_D' s + 1 \right) \tag{1.5}$$

Multiplying out gives the ideal settings that act on the loop, as shown in Equation (1.6).

$$K_c = K_c' \left(1 + \frac{\tau_D'}{\tau_I'} \right), \qquad \tau_I = \tau_I' + \tau_D', \qquad \tau_D = \frac{\tau_I' \tau_D'}{\tau_I' + \tau_D'} \tag{1.6}$$

The inverse exists only when $\tau_I \geq 4\tau_D$. A tuning used later in this section is $K_c = 3\,\%/^\circ\mathrm{C}$, $\tau_I = 4\,\mathrm{min}$ and $\tau_D = 1\,\mathrm{min}$ in the ideal form. Its parallel gains are $K_i = 0.75\,\%/(^\circ\mathrm{C\,min})$ and $K_d = 3\,\%\,\mathrm{min}/^\circ\mathrm{C}$, and its series settings are $K_c' = 1.5$, $\tau_I' = 2$ and $\tau_D' = 2$. The same three numbers typed into a series algorithm without conversion act as $K_c = 3.75$, $\tau_I = 5$ and $\tau_D = 0.8$, a gain 25 % higher than intended. With $\tau_D = 0$ the ideal and series forms are identical. No chapter of this volume runs a series algorithm. The form is given because settings taken from a chapter may be entered in an industrial controller that does.

A digital controller computes the terms at each execution k. The position form computes the whole output, as shown in Equation (1.7), and the velocity form computes its change, as shown in Equation (1.8).

$$u_k = u_b + K_c \left[e_k + \frac{\Delta t}{\tau_I} \sum_{j \leq k} e_j - \tau_D \frac{y_k - y_{k-1}}{\Delta t} \right] \tag{1.7}$$

$$u_k = u_{k-1} + K_c \left[(e_k - e_{k-1}) + \frac{\Delta t}{\tau_I} e_k - \tau_D \frac{y_k - 2y_{k-1} + y_{k-2}}{\Delta t} \right] \tag{1.8}$$

The two give the same output while the gains are constant and no limit is active. They differ in where the operating point is stored. The position form needs a bias and keeps a separate sum of the error. The velocity form has neither, because the last output is its memory. Every chapter of this volume runs a velocity form, and Section 1.6.2 gives the algorithm as the apps implement it.

1.4.3 Direct and Reverse Action

Feedback is negative only when the controller pushes against the error. The oven has a positive process gain, so a temperature above the setpoint must lower the heater power. The output falls as the PV rises, which is a reverse-acting controller. With the error written as $a(\mathrm{SP} - \mathrm{PV})$, it has $a = +1$. A loop that cools the oven with a damper would have $K_p < 0$ and would need $a = -1$, a direct-acting controller whose output rises with the PV. The test that never fails is the sign of the loop gain: $a K_c K_p$ must be positive. With the wrong action the loop gain is negative, each correction enlarges the error and the output runs to a limit.

Three chapters have a negative process gain. More marking shortens the router queue in Chapter 7, a higher price shortens the pickup time in Chapter 9 and a higher policy rate lowers inflation in Chapter 11. The two names are easy to confuse, so each of those chapters is best read by its error equation and not by the name.

> ✦ DID YOU KNOW? **A PID Behind a Monetary Rule**
>
> PID algorithms have been used in many fields, and one unconventional use is to explain popular monetary policy rules [5]. Chapter 11 runs a velocity-form PID and the Taylor rule in the same slot of one simulated bank, so the two can be compared on the same economy.

1.4.4 A Step Test and Its Fit

The model of Equation (1.1) is found from one experiment. The loop is put in Manual at a steady state, the MV is stepped by Δu and the PV is recorded until it settles. The gain is the PV change divided by Δu. Two times read from the record give the other two constants, as shown in Equation (1.9).

$$K_p = \frac{\Delta y}{\Delta u}, \qquad \tau_p = 1.5\,(t_{63} - t_{28}), \qquad \theta_p = t_{63} - \tau_p \tag{1.9}$$

where t_{28} and t_{63} are the times after the step at which the PV has covered 28.3 and 63.2 % of its change. A first-order response reaches those levels $\tau_p/3$ and τ_p after it starts to move.

Figure 1.3 shows the test on the oven. A step from 50 to 60 % raises the temperature from 120 toward 140 °C, so $K_p = 20/10 = 2\,^\circ\mathrm{C}/\%$. The PV passes 125.66 °C at $t_{28} = 5.33\,\mathrm{min}$ and 132.64 °C

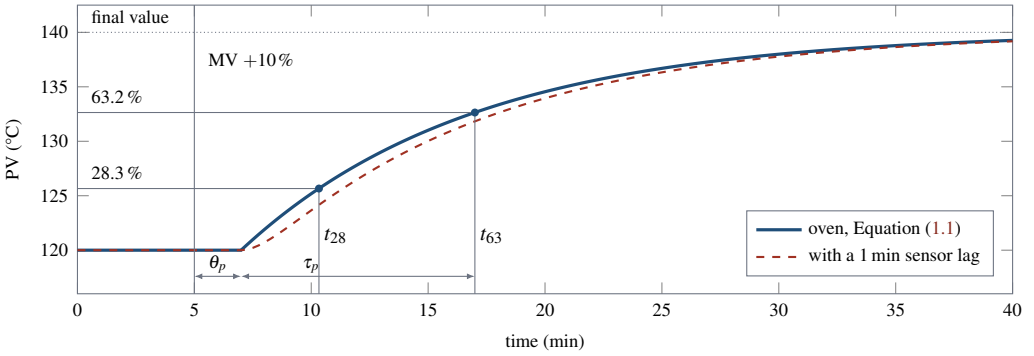


Figure 1.3 Step Test on the Oven With the Two-Point Fit. The MV Steps From 50 to 60 % at 5 min. The Solid Curve Is the Oven and the Dashed Curve Is the Same Test Read Through a 1 min Sensor Lag.

at $t_{63} = 12.00\,\text{min}$, which give $\tau_p = 10.0\,\text{min}$ and $\theta_p = 2.0\,\text{min}$. On a plant the step must stand clear of the noise and still be small enough to keep the process near its operating point. A second step in the opposite direction checks the fit.

A real response is never exactly first order, and the fit then returns effective constants. The dashed curve of Figure 1.3 adds a sensor lag of 1 min. The two-point reading of that record gives $\tau_p = 10.0\,\text{min}$ and $\theta_p = 3.0\,\text{min}$, and a least-squares fit gives 10.3 and 2.9 min. A small lag is read almost entirely as dead time. The half rule, which adds half of a neglected lag to each constant, would predict 10.5 and 2.5 min [17]. The fitted dead time is the number to tune with, because the loop feels the lag as delay. Chapter 4 fits a dead time of 0.94 min to a plant whose transport delay is 0.30 min, and Chapter 5 shows that a valve lag adds its full length to the apparent dead time of an integrating process. Chapter 2 shows the other hazard: on a nonlinear plant the fitted gain depends on the size of the step.

1.4.5 Tuning From the Model

Internal model control (IMC) tuning, also called lambda tuning, starts from the response the designer wants. The PV should follow a setpoint change as a first-order lag of time constant τ_c after the dead time, so the closed-loop transfer function is $T(s) = \mathrm{e}^{-\theta_p s}/(\tau_c s + 1)$. The controller that produces it is $C = T/[G(1-T)]$. Substituting Equation (1.1) and approximating the delay in the denominator by $\mathrm{e}^{-\theta_p s} \approx 1 - \theta_p s$ gives Equation (1.10).

$$C(s) = \frac{\tau_p s + 1}{K_p\,(\tau_c + \theta_p)\,s} = \frac{\tau_p}{K_p\,(\tau_c + \theta_p)}\left(1 + \frac{1}{\tau_p s}\right) \tag{1.10}$$

This is a PI controller [16]. The SIMC rule keeps its gain and shortens the integral time of a slow process so that loads are rejected sooner [17], as shown in Equation (1.11).

$$K_c = \frac{\tau_p}{K_p\,(\tau_c + \theta_p)}, \qquad \tau_I = \min\bigl(\tau_p,\, 4\,(\tau_c + \theta_p)\bigr) \tag{1.11}$$

The single choice is τ_c. A value equal to the dead time is fast with good margins, and a larger value is smoother. On the oven $\tau_c = \theta_p = 2\,\text{min}$ gives $K_c = 10/(2 \times 4) = 1.25\,\%/^{\circ}\text{C}$ and $\tau_I = \min(10, 16) = 10\,\text{min}$. Chapter 3, Chapter 4 and Chapter 6 tune with this rule.

The Ziegler–Nichols reaction-curve rule is older and far more aggressive [8]. It was designed for a decay ratio of one quarter, which accepts a large first overshoot. It is Equation (1.12).

$$K_c = \frac{1.2\,\tau_p}{K_p\,\theta_p}, \qquad \tau_I = 2\,\theta_p, \qquad \tau_D = 0.5\,\theta_p \tag{1.12}$$

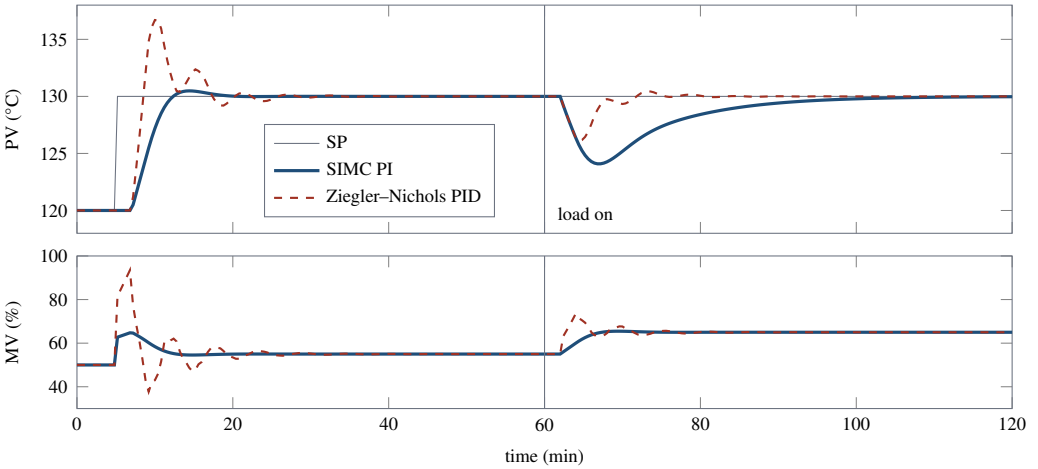


Figure 1.4 Two Tunings of the Oven Loop. The Setpoint Steps From 120 to 130 °C at 5 min and a Load of 10 % of Heater Power Enters at 60 min. Solid: SIMC PI. Dashed: Ziegler–Nichols PID With a Derivative Filter of 0.1 min.

Table 1.1 The Two Tunings of Figure 1.4. Settling Is Into ±0.2 °C. Margins Are Defined in Section 1.4.12.

	SIMC PI	Ziegler–Nichols PID
K_c (%/°C), τ_I, τ_D (min)	1.25, 10, 0	3, 4, 1
Setpoint step: overshoot (%)	4.8	67.7
Setpoint step: settling time (min)	12.5	20.2
Setpoint step: IAE (°C min)	43.7	54.8
Load step: largest deviation (°C)	5.9	3.9
Load step: IAE (°C min)	79.7	16.1
MV travel over the run (%)	37	220
Gain margin, phase margin	3.14, 61°	1.33, 39°

On the oven it gives $K_c = 3\,\%/^\circ\mathrm{C}$, $\tau_I = 4\,\mathrm{min}$ and $\tau_D = 1\,\mathrm{min}$, the tuning converted in Section 1.4.2. Chapter 4 and Chapter 5 load both rules from the reader's own fit.

Figure 1.4 runs the two tunings through a setpoint step of 10 °C and a load of 10 % of heater power, and Table 1.1 gives the measures. The comparison is a trade and not a ranking. The Ziegler–Nichols tuning rejects the load with a fifth of the integrated error, because its integral gain K_c/τ_I is six times larger. It pays with 68 % overshoot on the setpoint step, six times the heater travel and a gain margin of 1.33, so a 33 % rise in process gain makes it unstable. Such thin margins are the known weakness of the rule [18]. The SIMC tuning tolerates a tripled gain.

Both rules divide by the dead time, so both inherit its reading error. A dead time read as 1 min doubles the Ziegler–Nichols gain to 6.0. The SIMC gain rises by one third, to 1.67, when τ_c is held at 2 min, because the dead time is added to τ_c before it divides. It also doubles if τ_c is tied to the misread value.

1.4.6 An Integrating Process

Some processes do not settle after a step. The basin level of Chapter 5 ramps for as long as makeup and losses differ. The model is Equation (1.13), and the step test reads a slope and a delay in place of a gain and a time constant.

$$G(s) = \frac{K' \mathrm{e}^{-\theta_p s}}{s}, \qquad K' = \frac{\text{change in slope of the PV}}{\Delta u} \tag{1.13}$$

An oven with no heat loss is the integrating variant of the running example. It keeps the initial slope of the real oven, $K' = K_p/\tau_p = 0.2\,^{\circ}\mathrm{C}/\mathrm{min}$ per percent, and the dead time of 2 min. The SIMC settings follow from Equation (1.11) as τ_p grows without bound at a fixed K_p/τ_p, as shown in Equation (1.14).

$$K_c = \frac{1}{K'(\tau_c + \theta_p)}, \qquad \tau_I = 4(\tau_c + \theta_p) \tag{1.14}$$

With $\tau_c = \theta_p$ the gain is again 1.25 %/°C and $\tau_I = 16\,\mathrm{min}$. The minimum in Equation (1.11) therefore tunes any process with $\tau_p > 4(\tau_c + \theta_p)$ as an integrator.

Two properties separate this loop from the self-regulating one. Proportional control alone tracks a setpoint with no offset, since the process supplies the integrator. A load d still leaves an offset of d/K_c, which is 8 °C for a 10 % load, so integral action is kept. The integral time must be long, because the controller and the process together put two integrators in the loop. On a setpoint step of 5 °C the overshoot is 29 % at $\tau_I = 16\,\mathrm{min}$, 51 % at 8 min and 90 % at 4 min, and at 2 min the loop is unstable. Shortening the integral time speeds up a self-regulating loop and destabilizes this one. Chapter 5 measures the same effect on its basin.

1.4.7 The Ultimate Gain and the Relay Test

A second way to characterize a loop needs no model. Under proportional control the loop oscillates steadily at one gain, the ultimate gain K_u, with the ultimate period P_u. The frequency $\omega_u = 2\pi/P_u$ is where the process phase reaches −180°, as shown in Equation (1.15).

$$\omega_u \theta_p + \arctan(\omega_u \tau_p) = \pi, \qquad K_u = \frac{1}{|G(j\omega_u)|} = \frac{\sqrt{1 + \omega_u^2 \tau_p^2}}{K_p} \tag{1.15}$$

For the oven $\omega_u = 0.844\,\mathrm{rad}/\mathrm{min}$, $P_u = 7.44\,\mathrm{min}$ and $K_u = 4.25\,\%/^{\circ}\mathrm{C}$. The integrating variant has $K_u = \pi/(2K'\theta_p) = 3.93$ and $P_u = 4\theta_p = 8\,\mathrm{min}$, the formulas Chapter 5 uses.

Raising the gain until a plant oscillates is slow and risky. The relay test finds the same point safely [11]. The controller is replaced by a relay that holds the MV at $u_0 + \delta$ while the PV is below the setpoint and at $u_0 - \delta$ while it is above. The loop settles into a bounded cycle near P_u, and the swing $\pm\alpha$ of the PV gives the gain, as shown in Equation (1.16).

$$K_u \approx \frac{4\,\delta}{\pi\sqrt{\alpha^2 - \varepsilon^2}} \tag{1.16}$$

where ε is the half-width of a switching band added so that noise does not chatter the relay, zero for an ideal relay. Figure 1.5 shows the test on the oven with $\delta = 10\,\%$ and no band. The cycle has a period of 7.6 min, 2 % longer than P_u, and a swing of ±3.75 °C, so Equation (1.16) returns $K_u = 3.39$. That is 20 % below the true value. The equation assumes a sine wave, and this PV is closer to a triangle, whose fundamental is only 0.81 of its peak. Chapter 12 avoids the error by taking the fundamentals of both signals.

Table 1.2 lists the rules that turn K_u and P_u into settings, with their values at the oven's true ultimate point. The Tyreus–Luyben settings [19] land close to the SIMC tuning, with a phase margin of 69°, against 36° for the Ziegler–Nichols PI.

A relay test measures one point of the frequency response and nothing else. Chapter 2 and Chapter 3 find that the switching band and the noise, not the plant, set their cycle, and Chapter 12 finds that its relay measures a coupling resonance and not the motion the loop must control.

1.4.8 Derivative Action and Its Filter

The derivative term adds phase lead, and it multiplies whatever changes fastest in its input. Two precautions are standard. The first takes the derivative of the measurement and not of the error, as Equations (1.7) and (1.8) already do. With the Ziegler–Nichols tuning a setpoint step of 10 °C

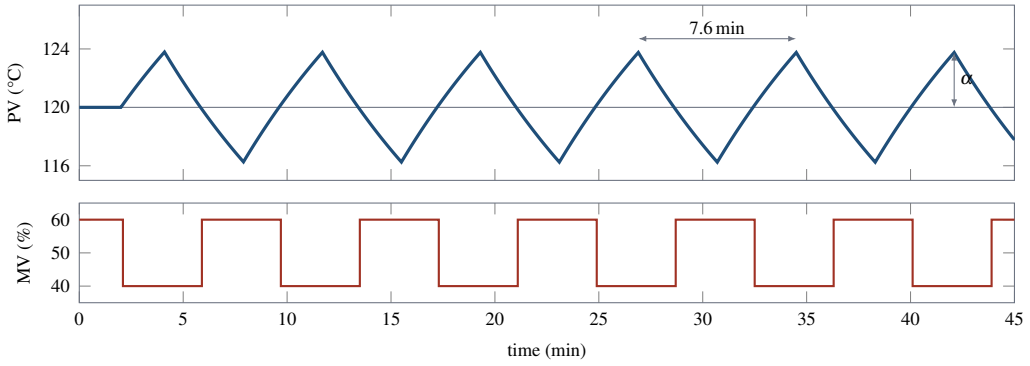


Figure 1.5 Relay Test on the Oven. The Relay Switches the Heater Between 40 and 60 % Each Time the PV Crosses 120 °C.

Table 1.2 Tuning Rules Built on the Ultimate Point, and Their Values for the Oven ($K_u = 4.25$, $P_u = 7.44$ min)

Rule	Settings	On the oven	Used in
Ziegler–Nichols PI	$0.45K_u$, $P_u/1.2$	1.91, 6.2 min	Chapter 12
Ziegler–Nichols PID	$0.6K_u$, $P_u/2$, $P_u/8$	2.55, 3.7, 0.93 min	Chapter 5
Tyreus–Luyben PI	$K_u/3.2$, $2.2P_u$	1.33, 16.4 min	Chapter 2, Chapter 3

Table 1.3 Derivative Filter on the Ziegler–Nichols PID With Measurement Noise of 0.2 °C

Filter time constant τ_f (min)	none	0.1	0.25	0.5
Standard deviation of the MV (%)	11.3	5.6	2.8	1.8
Phase margin (°)	41.0	39.3	34.9	23.4

differentiated through the error would ask for $K_c\tau_D\Delta\mathrm{SP}/\Delta t = 300\,\%$ of output for one execution. Differentiating the measurement removes that derivative kick and leaves the proportional kick of 30 %.

The second precaution is a filter, because measurement noise changes at every execution. A first-order filter of time constant τ_f turns the term into Equation (1.17).

$$D(s) = -\frac{K_c\,\tau_D\,s}{\tau_f s + 1}\,Y(s) \tag{1.17}$$

Its gain at high frequency is limited to $K_c\tau_D/\tau_f$. Without the filter the gain per execution is $K_c\tau_D/\Delta t$, which is 30 %/°C here. Table 1.3 shows the trade on the oven with noise of standard deviation 0.2 °C. A filter of $\tau_D/10$ halves the heater movement for 2° of phase margin. Longer filters quiet the heater further and take margin faster, so the filter belongs to the tuning and is not an afterthought [20]. The SIMC PI moves the heater with a standard deviation of a quarter of a percent on the same noise, which is why many loops run without derivative action.

Chapter 2 and Chapter 5 measure derivative action on a noisy speedometer and a noisy level. Chapter 10 finds that a derivative term helps its lagged loop on clean data and hurts it on published data, and Chapter 7 explains why derivative action earns little when dead time dominates.

1.4.9 Windup and Three Remedies

An actuator at a limit breaks the loop. The error persists, the integral keeps growing, and when the PV finally arrives the stored integral holds the output at the limit long after it should have left [21]. A setpoint step from 120 to 200 °C shows it on the oven with the SIMC PI. The new setpoint needs 90 %

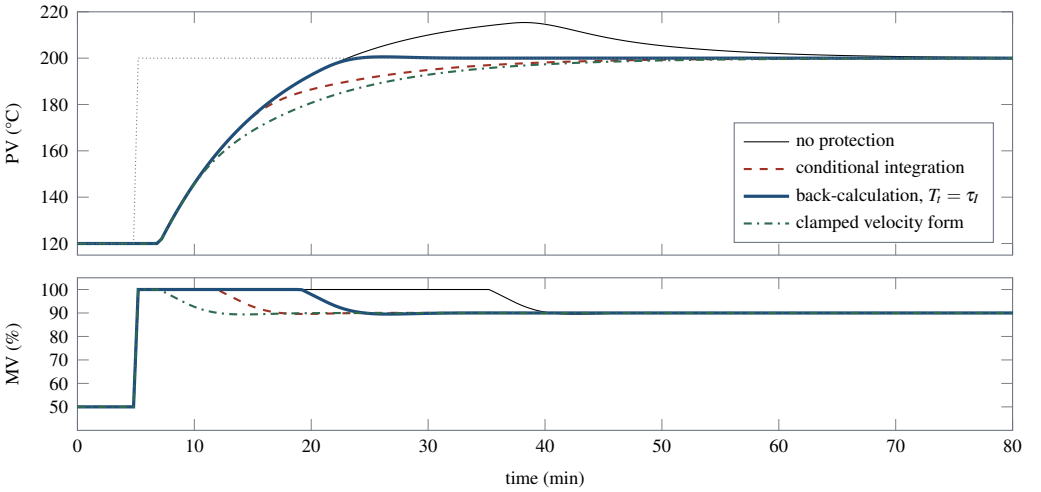


Figure 1.6 Windup on a Setpoint Step From 120 to 200 °C at 5 min, SIMC PI. Thin Solid: No Protection. Dashed: Conditional Integration. Thick Solid: Back-Calculation. Dash-Dotted: Clamped Velocity Form. The Dotted Line Is the Setpoint.

of power, and at the step the proportional term alone raises the computed output to 150 %. Three remedies are in common use [4, 22].

- *Conditional integration* stops the integral while the output is limited and the error would drive it further into the limit.
- *Back-calculation* feeds the difference between the applied and the computed output back to the integral through a tracking time constant T_t, as in Equation (1.26).
- *A clamped velocity form* limits the running output of Equation (1.8) itself, so that nothing is stored beyond the limit.

Figure 1.6 and Table 1.4 compare them with an unprotected integral. Without protection the integral term reaches 80 %, the heater stays at full power for 30 min, and the oven overshoots by 15.4 °C. All three remedies cut the overshoot to less than 1 °C. They differ in when the output leaves the limit. The clamped velocity form leaves after 2 min, as soon as the PV starts to rise, and then creeps to the setpoint. Back-calculation with $T_t = \tau_I$ holds full power for 14 min and settles in less than half the time of the others. A shorter T_t is not better here: at $T_t = 2$ min the settling time grows from 18 to 42 min.

The setting $T_t = \tau_I$ has a simple reading. With the output at a limit u_{lim} and no derivative term, the proportional parts of Equation (1.26) cancel and leave Equation (1.18).

$$\frac{\mathrm{d}(u_b + I)}{\mathrm{d}t} = \frac{u_{\text{lim}} - (u_b + I)}{\tau_I} \tag{1.18}$$

The stored output $u_b + I$ relaxes toward the limit with time constant τ_I, whatever the error, and the computed output exceeds the limit only by the proportional term $K_c e$. On the oven the stored output has reached 88 % when the heater leaves full power, close to the 90 % the new setpoint needs. Chapter 2, Chapter 3 and Chapter 7 derive the same relaxation for their loops.

Clamping the running output has a second cost. Every increment that crosses the limit is discarded and never restored. With integral action off, the clamped form settles this step at 64.3 % of power where the position form gives 78.6 %, because the 50 % discarded at the limit is lost. Noise does the same in small steps. With a setpoint of 225 °C, which is out of reach, and noise of 0.5 °C, the clamped PI averages 99.3 % of power and holds the oven 1.4 °C below what full power reaches.

Table 1.4 The Four Runs of Figure 1.6. Times Are From the Setpoint Step. Settling Is Into ±1 °C.

Integral protection	Leaves the limit (min)	Peak PV (°C)	Settling (min)	IAE (°C min)
None	30.3	215.4	61.8	960
Conditional integration	7.2	200.0	41.2	779
Back-calculation, $T_t = \tau_I$	14.2	200.6	18.0	642
Clamped velocity form	2.1	200.0	44.9	884

Table 1.5 Effect of the Control Period on the Oven Loop. The Tuning in the Last Two Rows Is the SIMC PI With $K_c = 1.25$ and $\tau_I = 10$ min.

Control period Δt (min)	0.1	0.5	1	2
Ultimate gain, exact sampled loop	4.15	3.79	3.39	2.76
Ultimate gain with $\theta_p + \Delta t/2$ in Equation (1.15)	4.16	3.82	3.47	2.95
Phase margin (°)	60.7	57.9	54.6	48.1
Overshoot of a setpoint step (%)	4.8	8.2	12.8	23.4

Back-calculation stays at 100 %. For this reason the apps limit only the output they emit and correct only the integral, as Section 1.6.2 describes.

Every chapter meets a limit. Chapter 3 is built on one, and it computes the overshoot that back-calculation avoids. Chapter 4 and Chapter 6 let the reader switch the protection off, and the second adds a rate limit, which winds an integral up far from either stop. Chapter 11 winds its integral up at the floor of the policy rate.

1.4.10 Bumpless Transfer

A switch from Manual to Auto should not move the actuator. It does when the algorithm starts from a state that does not match the output in force. Suppose the oven is in Manual at 50 % with an error of 4 °C for 20 min. A position form whose integral kept running in Manual has stored $(K_c/\tau_I)\,e\,t = 10\,\%$, and its first automatic output is $50 + 5 + 10 = 65\,\%$, a bump of 15 %. Resetting the integral to zero at the switch still leaves the proportional bump of 5 %. The remedy is tracking: while in Manual the integral is set at every execution so that the computed output equals the manual one. The first automatic output is then 50 %, and the heater moves by the integral increment of 0.05 % per execution. The conditioning technique treats windup and transfer as this one problem [23].

A change of gain is a second source of bumps. Doubling K_c while the error is 4 °C moves the output by 5 % at once in the position form. Equation (1.8) does not move it, because it applies the new gain only to later changes of the error. The apps form each increment from the whole proportional term, so a new gain acts at once, as Chapter 2 and Chapter 3 note. The chapters on markets and economies apply tracking to a different switch: Chapter 9, Chapter 10 and Chapter 11 keep an idle policy tracking the active one.

1.4.11 Sample Time

A digital controller reads the PV once per period and holds its output until the next execution. On average the held output is half a period old, so sampling acts on the loop much like an added dead time of $\Delta t/2$ [17]. Table 1.5 tests that rule on the oven against the exact sampled loop. The ultimate gain falls from 4.25 to 2.76 as the period grows to 2 min, and the SIMC PI loses 13° of phase margin and overshoots by 23 %. The rule predicts each ultimate gain within 7 %.

The remedy is to tune for the dead time the loop has. With $\theta_p + \Delta t/2 = 3$ min in Equation (1.11) the gain at $\Delta t = 2$ min falls to 0.83, the phase margin returns to 60° and the overshoot to 4.9 %. By the same rule a period below a fifth of the dead time costs this tuning less than 3°. A long period is not always a choice. The ramp meter of Chapter 8 acts on one detector report a minute, and the

bank of Chapter 11 meets eight times a year. The budget of Chapter 10 is decided once a year, which halves the gain its loop can use. Chapters 9 to 11 also round the output to a fixed step, which can leave a small sustained cycle such as the one Chapter 11 measures. The drive of Chapter 12 runs three loops at three rates and counts the phase each hold costs.

1.4.12 Margins With Dead Time

Two numbers say how far a loop is from instability [5]. Let $L(j\omega) = C(j\omega)\,G(j\omega)$ be the frequency response around the loop. The gain margin is the factor by which the loop gain can rise before the loop oscillates. The phase margin is the phase lag that can be added at the crossover frequency ω_c. Both are defined in Equation (1.19).

$$\mathrm{GM} = \frac{1}{|L(j\omega_{180})|}, \qquad \mathrm{PM} = 180^\circ + \angle L(j\omega_c), \qquad |L(j\omega_c)| = 1 \tag{1.19}$$

where ω_{180} is the frequency at which the phase of L is -180°. Dead time is the reason margins run out. A delay θ_p leaves the magnitude alone and subtracts $\omega\theta_p$ radians of phase, without limit as the frequency rises.

The SIMC PI with $\tau_I = \tau_p$ makes the loop simple enough to solve by hand. The integral zero cancels the process lag and $L(s) = \mathrm{e}^{-\theta_p s}/[(\tau_c + \theta_p)s]$, which gives Equation (1.20).

$$\omega_c = \frac{1}{\tau_c + \theta_p}, \qquad \mathrm{PM} = \frac{\pi}{2} - \frac{\theta_p}{\tau_c + \theta_p}, \qquad \mathrm{GM} = \frac{\pi}{2}\,\frac{\tau_c + \theta_p}{\theta_p} \tag{1.20}$$

with the phase margin in radians. For $\tau_c = \theta_p$ the phase margin is 61.4° and the gain margin is $\pi = 3.14$, whatever τ_p and K_p are. The loop also tolerates a dead time up to 3.14 times the fitted one. Solving the phase margin for the crossover gives the bound of Equation (1.21).

$$\omega_c\,\theta_p \le \frac{\pi}{2} - \mathrm{PM} \tag{1.21}$$

A phase margin of 60° limits the crossover to $0.52/\theta_p$, 45° to $0.79/\theta_p$ and 30° to $1.05/\theta_p$. This is the rule of thumb that a loop cannot be made faster than about $1/\theta_p$ [24]. It is exact for a PI controller whose integral time cancels the lag. Derivative action buys a little more: the Ziegler–Nichols PID of Table 1.1 crosses at $1.26/\theta_p$ with a phase margin of 39°. No tuning removes the delay itself. Volume 5 treats the predictors that work around it.

Chapter 7 is this subsection on a router, where the dead time is the round-trip time and a longer path takes the phase margin from 68° to zero with no setting touched. Chapter 12 reads both margins from a measured frequency response, and Chapter 4 has a dead time that shortens as throughput rises.

1.4.13 Scheduling a Gain

Every tuning rule above sets K_c in inverse proportion to K_p. When the process gain moves with the operating point, a fixed controller gain makes the loop gain $K_c K_p$ move with it. The gain margin says how much rise is tolerated: a factor of 3.14 for the SIMC PI with $\tau_c = \theta_p$. Suppose the oven's gain at another operating point is two or three times the fitted 2 °C/%. With K_c fixed at 1.25 the phase margin falls from 61° to 33 and 4°, and the overshoot of a setpoint step rises from 4.8 to 52 and 103 %. At half the fitted gain the loop is stable and slow, reaching 63 % of a step in 8.1 min against 4.6 min.

Gain scheduling removes the variation by changing the controller gain with a measured variable x that tells where the process is [25], as shown in Equation (1.22).

$$K_c(x) = K_c^0\,\frac{K_p^0}{K_p(x)} \tag{1.22}$$

where K_c^0 is the gain tuned at the point where the process gain is K_p^0. On the oven the scheduled gains are 0.625 and 0.417 at two and three times the fitted gain, and the step response returns to 4.8 %

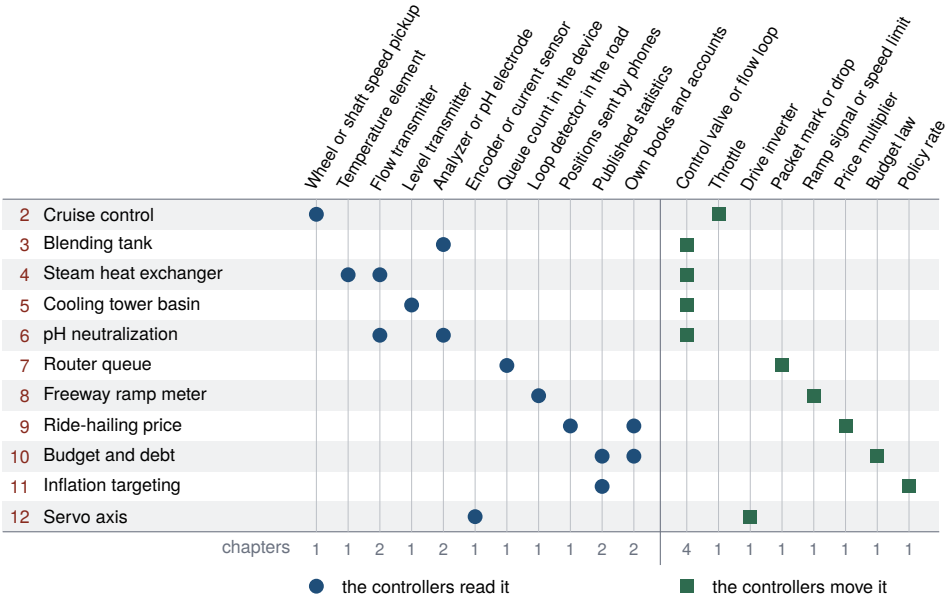


Figure 1.7 What the controllers of each chapter read and move: a round mark where the chapter's controllers read the source and a square mark where they move the element, as its variables table and its technology section name them. A value computed from other readings, such as the pickup time of Chapter 9, is marked under the source it is computed from. Notice that four chapters read no field instrument at all.

overshoot in each case. The schedule needs a gain curve and a measurement to enter it with, and it is only as good as both.

Chapter 6 is the extreme case. Its process gain changes by a factor of 1.96×10^4 across one valve stroke, far beyond any gain margin, and its schedule follows the slope of an assumed titration curve. Chapter 2 is the mild case. Its gain falls by a factor of three with speed, but its time constant falls with it, and Equation (1.11) depends on the ratio τ_p/K_p, which stays fixed. The proportional gain of Chapter 3 is constant for the same reason. In Chapter 9 the gain from price to pickup time grows twentyfold as the market nears a fold.

1.5 Instruments and Final Elements in This Volume

Figure 1.7 lists what the controllers of each chapter read and what they move. Seven chapters read a field instrument, from a wheel-speed pickup to a detector loop cut into the road. The other four read a count inside a router, positions sent by phones, published statistics or their own books, and they move a packet decision, a price, a budget or a policy rate. A control valve or a flow loop is the final element of four chapters, more than any other.

A control valve with excess stem friction moves in jerks and does not answer small reversals of its signal, an effect called stiction. In valves without a positioner to correct the stem, the resulting hysteresis can exceed 10 % [26]. A positioner is a closed-loop controller in its own right [26], the inner loop of Figure 1.8. With integral action a sticking valve makes the loop hunt above and below the setpoint, and the only cure is to reduce the friction [26]. The dead band of Chapter 4 stands for it.

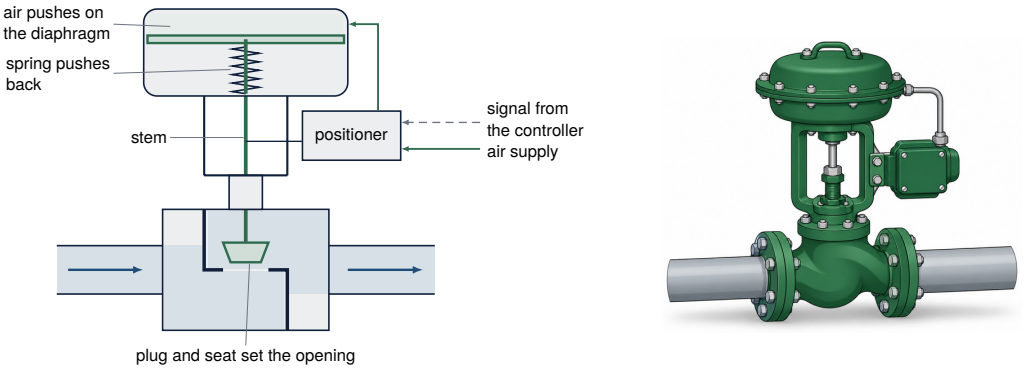


Figure 1.8 Principle of a globe control valve with its positioner. Notice that the positioner compares the controller's signal with the stem position, so the PID above it sees a valve that goes where it is sent until friction or the supply runs out (Chapter 4, Chapter 5 and Chapter 6).

> **DID YOU KNOW? A Zero That Is Not Zero**
>
> The 3 to 15 psi pneumatic and the 4 to 20 mA electric signal standards both begin above zero. This live zero tells a true reading of 0 % from a failed signal, such as a leaking tube or a severed cable [26].

A pH electrode is a glass bulb whose voltage grows by 59.17 mV at 25 °C for each tenfold change in hydrogen ion activity, one pH unit. The voltage also depends on temperature, so many probes carry a resistance thermometer [26]. That reading is the process variable of Chapter 6, whose process gain spans four decades.

Four chapters have no transmitter. Output is compiled from surveys and records: the first estimate of a quarter appears about a month after it ends and is revised several times, and a monthly price index is released in the middle of the following month [27]. A phone's satellite position is typically good to 4.9 m under open sky and worse near buildings [28]. A policy rate is moved by administered rates, among them the rate on reserve balances, set to support the committee's target range [29]. Late data, rounding and a calendar are the instruments of these loops, and Section 1.4.11 prices them.

PID control appears in dedicated devices and in plants with thousands of controllers, alone, inside distributed systems and in embedded systems, and higher controllers give setpoints to the PID layer [5]. Model predictive control sits one level above the PID loops that move the valves [30]. Figure 1.9 places the volume's loops on their platforms. Industrial loops run with cycle times from milliseconds to minutes [31]. The apps go further, from the 0.1 ms current loop of Chapter 12 to the budget of Chapter 10, decided once a year.

1.6 The Simulator Framework

Every chapter of this book has an app that runs in a web browser, collected with the others at https://apmonitor.com/apps/. About 240 apps exist across the series. All but a few are built on one shared framework, so the engine, the PID algorithm, the trends and the data export behave the same way in every chapter. The exceptions are standalone studios, such as the cascade studio and the two Temperature Control Lab apps, which follow the same ideas in their own code. This section describes the framework once and states where the standalone apps differ. Table 1.6 lists what a reader meets.

1.6.1 The Simulation Engine

Each app carries a plant model written as ordinary differential equations $\frac{\mathrm{d}x}{\mathrm{d}t} = f(x,u,d,p)$ in the states x, the inputs u, the disturbances d and the parameters p. The engine integrates the model with the classical fourth-order Runge–Kutta method on a fixed step h set for the fastest dynamics of the

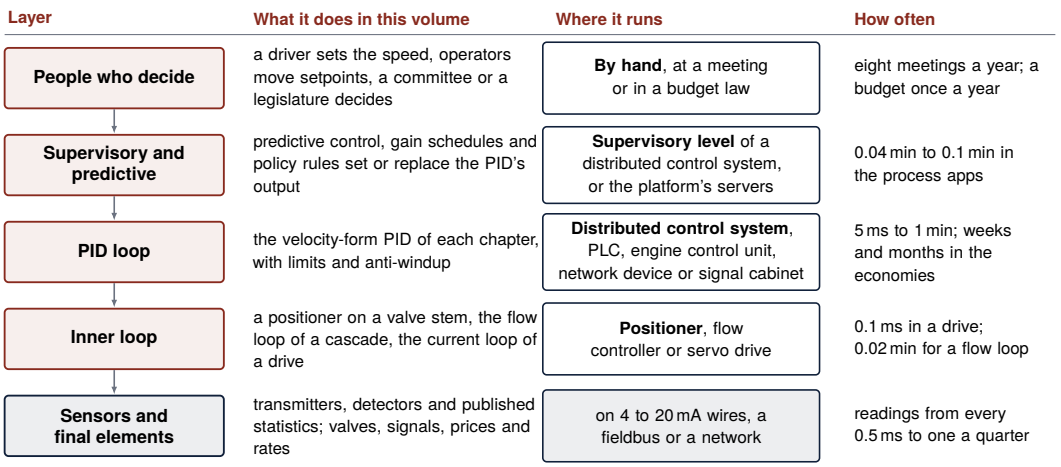


Figure 1.9 The control layers of this volume, the platform that hosts each and how often each acts. Platforms are from [5], [30] and [31], and the app periods from the chapters' technology sections. Notice that the economies sit in the same layers as the plants, with a meeting or a budget law in place of a computer.

Table 1.6 What Every App Provides. Standalone Apps Are Described in Their Own Chapters.

Element	What it does
Run, Pause, Reset	Start, hold and restart the run. Reset keeps the reader's settings.
Speed	Simulated time per second of real time. It changes the pace and never the result.
Manual and Auto	Hand each actuator to the reader or to the controller, without a bump.
Scenario list	Numbered cases, each with a briefing, a completion rule and a reload button.
Control panel	Setpoints, manual outputs, tuning, options and disturbances, at a Beginner or an Advanced level.
Trends	SP and PV above the MV for each loop, with limits, event marks and a cursor.
Scorecard	IAE, overshoot, settling time, MV travel and time at a limit for each loop.
Learn card	Predict, Try It, two hints and Explain, opened by the reader in that order.
Show Model	Equations, live variable values, editable parameters and assumptions.
Export CSV	Every logged sample of the run as a text file with a header row.
Instructions	A separate page with background, an operating guide and the model's limitations.

app. Some apps supply their own step, such as explicit Euler for a discontinuous right-hand side or an exact update for a model driven by events.

Three clocks run at separate rates. The plant advances every step h. The controller executes every control period Δt, equal to h unless the app declares a slower controller, and holds its outputs between executions. The log takes one sample at a fixed interval of simulated time for the trends, the scorecard and the exported file. The order of work in one step is shown in Figure 1.10. The controller also executes once when a scenario is loaded, so every run starts from an initialized controller.

Drawing the screen is kept apart from these clocks. A speed of 5 means five units of the app's time base per second of real time. Speed, frame rate and pauses change how fast a run is shown and never what it computes, so a phone and a workstation produce the same trajectory and the same file.

Noise comes from a deterministic pseudorandom generator. Its seed is declared by the app, or by a scenario that needs its own noise record, and it is restored whenever a scenario is loaded or Reset is pressed. The framework gives the reader no seed setting, so *default seed* in this book means the seed the app ships with. A run repeats bit for bit when the same scenario is loaded and the same settings are made before Run. A change made by hand during a run lands on a different step each time, which

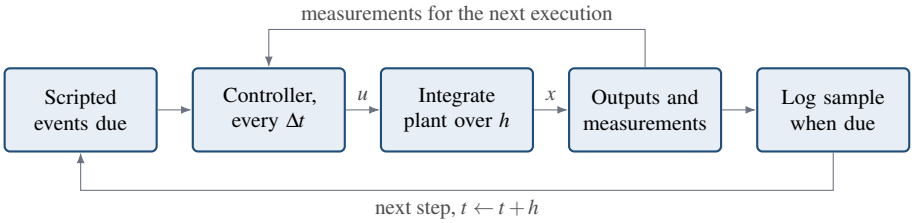


Figure 1.10 Work of the Engine in One Integration Step

is one reason numeric answers carry a tolerance. The standalone simulators draw noise from the browser's own generator, so a noisy run there repeats in its statistics and not sample for sample.

Run and Pause start and hold the clock. Reset restarts the run at $t = 0$, paused, and keeps the reader's settings: every setpoint, tuning value, option, model parameter, disturbance slider and mode the reader changed is put back before the plant and the controller are initialized. Scripted events replay, and a one-shot disturbance button pressed during the run is cleared. The reload button beside the scenario list restores the scenario's own settings instead.

1.6.2 The Velocity-Form PID Algorithm

One PID implementation serves the PID loops of every framework app. It is the velocity (incremental) form: each execution adds the change in the proportional and derivative terms to a running sum, and the integral is a separate term, as shown in Equations (1.23) to (1.26) for execution k and control period Δt.

$$e_k = a\,(\mathrm{SP}_k - \mathrm{PV}_k), \qquad P_k = K_c\,e_k, \qquad D_k = -a\,K_c\,\tau_D\,\frac{\mathrm{PV}_k^f - \mathrm{PV}_{k-1}^f}{\Delta t} \tag{1.23}$$

$$S_k = S_{k-1} + (P_k - P_{k-1}) + (D_k - D_{k-1}) \tag{1.24}$$

$$v_k = u_b + S_k + I_{k-1}, \qquad u_k = \mathrm{sat}(v_k) \tag{1.25}$$

$$I_k = I_{k-1} + \left[\frac{K_c}{\tau_I}\,e_k + \frac{u_k - v_k}{T_t}\right]\Delta t \tag{1.26}$$

where $a = \pm 1$ is the controller action, PV^f is the measurement after an optional first-order filter, u_b is the output bias, S is the running sum of proportional and derivative increments, v is the output before limiting, u is the output sent to the actuator, I is the integral term and T_t is the tracking time constant, equal to τ_I unless a loop sets another value. The function sat clamps v_k to the actuator range and then applies a rate limit if the loop declares one.

The limit acts on the emitted output u_k only. The sum S_k is never clamped, so it equals $P_k + D_k$ and the output before limiting is the position form $u_b + K_c e_k + D_k + I_{k-1}$. Anti-windup touches the integral alone. While the output is limited, the last term of Equation (1.26) drives I toward the value that reproduces the output applied, which is back-calculation [21]. With no integral action, Equation (1.26) is skipped. Clamping the running sum would instead discard each increment that crosses a limit and keep each one that returns, so symmetric noise would push the output off the limit even in a proportional-only loop.

The derivative acts on the measurement, so a setpoint step gives a proportional kick and no derivative kick. A change of K_c acts on the whole error at the next execution, as in the position form. The action $a = +1$ suits a process whose PV rises with the MV, a reverse-acting controller in the ISA convention, and $a = -1$ gives the direct-acting one.

In Manual the controller keeps executing and sets $I_k = u_{\mathrm{man}} - u_b - S_k$, so the output it would send equals the manual value u_{man}. The first automatic output differs from it by one period's change in P and D. In Auto the engine copies each controller output to the manual station, so a switch to Manual

is also bumpless. The standalone studios with PID loops use velocity-form code of their own, with the limit on the output and anti-windup on the integral alone, in which a gain change applies only to later increments.

1.6.3 The Predictive Control and Estimator Options

More than 100 apps offer model predictive control (MPC) as an option beside the structure they open with, which stays the default. Volume 9 develops the theory. In most of them the controller plans with a small linear model in discrete time as shown in Equation (1.27).

$$x_{k+1} = A\,x_k + B\,u_k + E\,w_k + c, \qquad y_k = C\,x_k \tag{1.27}$$

where w is a measured or scheduled disturbance and c is a constant left by linearization. The model comes from a step test at the design point, from the app's balances linearized there or again at every decision, or from the model the reader fits. It never reads the true states or parameters of the run, so a parameter the reader changes in Show Model is model mismatch, as on a real plant.

At each decision the controller minimizes the objective of Equation (1.28) over a horizon of N steps.

$$J = \sum_{k=1}^{N} (y_k - r_k)^{\top} Q\,(y_k - r_k) + \sum_{j} \Delta u_j^{\top} R\,\Delta u_j + \sum_{i} \rho_i\, s_i^2 \tag{1.28}$$

where r is the setpoint, Q weights the tracking error, R is the move suppression on each input change Δu_j and ρ_i is the price of a slack $s_i \geq 0$ that lets an output band be broken when it cannot be held. Actuator limits are hard constraints. The quadratic program is solved in the browser by a dual active-set method [32]. The first move is applied, and the plan is made again at the next decision.

The plan starts from an estimate of the model state. About half of these apps use a steady-state Kalman filter on the model with an added disturbance state, which absorbs what the model did not predict and removes steady offset without an integral term [33]. Others correct the model output with a filtered bias to the same end. About a dozen apps also offer moving horizon estimation (MHE), which solves the estimation problem over a window of recent samples and can hold bounds that a Kalman filter cannot, such as a concentration that is never negative [34].

With the MPC selected, the tuning fields of the loops it replaces leave the Control panel. In most of these apps the Advanced level shows up to three settings, each a multiple of the shipped value: the prediction horizon, the move suppression and the band price. The trend draws the accepted plan ahead of the present time, and a list of constraint prices shows which limits bind. A plan that is not optimal, not finite or outside its limits is rejected, and a fallback, in most apps the shipped controller, holds the actuator until a plan succeeds. The Model Fitting and Tuning Studio builds the same kind of controller from the reader's own data in its tab, MPC from the fit.

1.6.4 Scenarios, Progress and Data

An app teaches through scenarios, the numbered cases at the top of the Control panel, nine to thirteen in most apps. Loading one sets the model parameters, the controller settings, the disturbances, the mode, the initial state and any scripted events. The information button beside the list opens a briefing in four parts: what the case is, why it matters, what to do and one detail for the practicing engineer. The Learn card holds the lesson back, with Predict and Try It before the run, hints on request and Explain afterward.

Each scenario has a completion rule that the app judges from the run on the simulated clock: a run length, the scripted events passed, a setting made by the reader, a state held for a stated time or a target met. Choosing a case earns nothing. A completed case is marked in the list, and a badge row shows the count and a level, Explorer after one case, Practitioner at half and Completed at all. Progress is kept in the reader's browser with no account. A certificate of progress or of completion can be issued from it, as a record of self-paced work and not an assessed credential.

One case, usually the last, is the graded challenge. It states a target, such as an error band to hold through a sequence of upsets. Changing the model parameters or the disturbances voids the attempt in most apps, and a record made with Reveal suggested tuning says so.

Show Model opens the equations, live values of the variables, the editable parameters and the assumptions. The trends plot SP and PV above the MV for each loop, with actuator limits and marks at setpoint moves, disturbances, mode changes and scripted events. The scorecard reports for each loop the integral of the absolute error (IAE), overshoot, settling time, MV travel and share of time at a limit, with the app's own measures, and it does not combine them into one score. IAE and travel accumulate from Reset, so two runs are compared at the same clock reading.

Export CSV saves the log as a text file with a header row of column names that carry their units and one row per logged sample. In most apps the time column is in seconds, minutes or hours according to the run length. The log holds the most recent 60 000 samples unless the app sets another size. Apps with the MPC option begin the file with comment lines that start with #, which a script must skip.

An app exercise in this book names the scenario by its number and name as the app shows them, every setting that differs from the scenario's defaults, the run length in simulated time, the seed (the default seed unless stated), what to read and where (a trend, a scorecard tile or a column of the exported file) and a tolerance on a numeric answer. Settings are made before Run unless the exercise gives a time.

1.7 Working With AI Agents While Studying

A reader of this book is likely to have an AI assistant within reach, and the apps make its work easy to test. This section gives the editor's view of how to use one while working through a chapter. In short, the assistant may do the algebra and the scripting, and the reader keeps the checking.

What to delegate. Tasks with a clear input and a result that can be tested are good candidates. An assistant can derive the linearization of a chapter's balance at an operating point and the transfer function that follows from it. It can write a short script that reads the exported CSV file and fits a first-order-plus-dead-time model to a step test. It can propose a first tuning from that fit with a named rule, convert units, explain a term in a model or outline a report. The assistant knows nothing about the app until it is told. Give it the equations and parameter values from Show Model, the column names of the exported file and the scenario that produced the data, as described in Section 1.6.4. Ask it to state its assumptions and the units of every result.

What to check, and how. An assistant returns a fluent answer whether or not the answer is right, so each result needs a test that it could fail. Four tests cover most of the errors a chapter will produce.

- *Units.* Every term of an equation carries the same units, and a process gain is in PV units per MV unit. A gain computed with the valve as a fraction is 100 times the gain of a loop whose valve is in percent.
- *Signs.* Decide from the physics which way the PV moves when the MV rises. Compare that with the sign of the fitted gain and with the controller action of Section 1.6.2.
- *Steady state.* Set the derivatives to zero. The model must return the operating point that Show Model reports, and a linearization must give zero deviation there.
- *A run on the app.* Load the named scenario at the default seed, enter the proposed numbers and run. The trend and the scorecard decide. A tuning that reads well on paper and holds the valve at a limit for half the run has failed.

A fitted model gets one more test: compare it with data it was not fitted to, such as a second step in the opposite direction. When a check fails, report the failure to the assistant with the evidence and ask for the cause, then check the correction the same way.

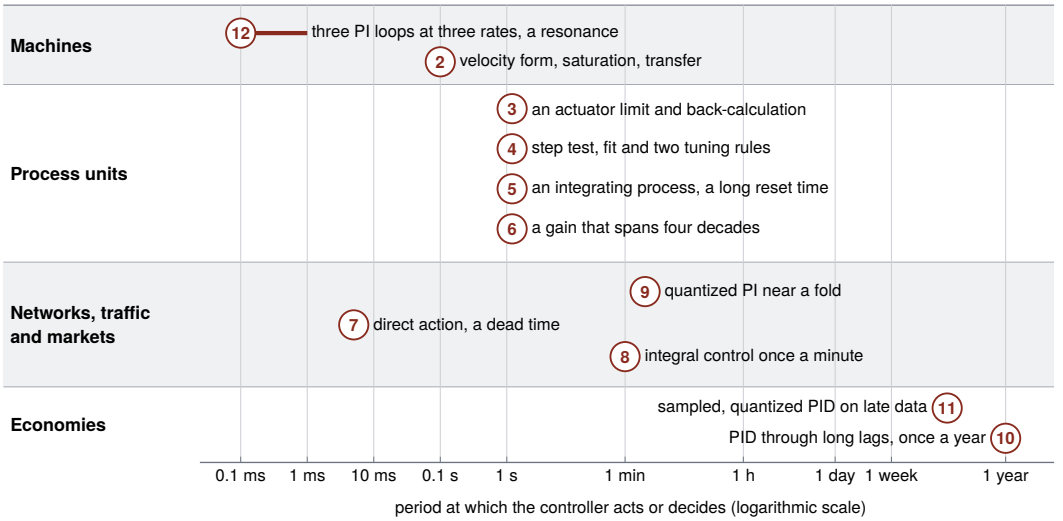


Figure 1.11 Method map of the volume. Each chapter is placed by the period at which its controller acts or decides and by the kind of plant, with its PID lesson from Table 1.7. Periods are those the chapters state for their apps. Notice that one algorithm spans eleven decades of period.

How the exercises train the habit. One exercise in each chapter is marked *Review the agent*. It presents a tuning, a model or a claim as an AI assistant might propose it, with a flaw, and asks the reader to find the flaw in about five minutes. The flaws are ordinary ones: a unit, a sign, a steady state that does not close, a result measured on the wrong scenario or at a different run length, or a limit that was ignored. Each is found with one of the tests above. The purpose is not distrust. An engineer who can say why an answer is right can hand an assistant much larger tasks and still stand behind the result. That larger picture, in which agents do most of the routine work and the engineer supervises and checks, is the subject of Chapter 13.

1.8 Chapter Roadmap

Table 1.7 lists the chapters with the lesson each adds to this one. The level says how much of Section 1.4 a chapter assumes. The time is an estimate for reading the chapter once and running each case of its app once, at about eight minutes a case. Figure 1.11 places the same chapters by the period at which their controllers act, from a tenth of a millisecond to a year.

Chapters 2 to 5 form a first course and are best read in order: the algorithm, a limit, a tuned loop with dead time and an integrating loop. Chapter 6 builds on the tuning of Chapter 4. Chapter 7 needs Section 1.4.12 and stands alone otherwise. Chapters 8 to 11 can be read in any order after Chapters 2 and 3, and Sections 1.4.9 and 1.4.11 prepare them. Chapter 12 uses nearly every subsection of this chapter and is best read last. A reader who has taken a first control course can take a shorter path through Chapters 4, 5, 6, 7 and 12.

Table 1.7 Roadmap of the Volume

Ch.	Plant	The PID lesson	Level	Time
2	Cruise control under quadratic drag	The velocity form, transfer and saturation. Gain and time constant move together with speed.	introductory	2.5 h
3	Stirred blending tank	An actuator limit and back-calculation. An offset that feedback cannot remove.	introductory	2.5 h
4	Steam-heated exchanger with hidden parameters	Step test, fit and two tuning rules on a loop with dead time. A gain that falls with load.	intermediate	4 h
5	Cooling tower basin makeup	An integrating process: slope and delay from a step, long reset time.	intermediate	2.5 h
6	pH neutralization tank	A gain that spans four decades. Scheduling against linearizing the measurement.	advanced	2.5 h
7	Router queue under TCP	Direct action and a dead time that sets the phase margin.	advanced	2.5 h
8	Freeway on-ramp meter	Integral control once a minute, just below a fold in the plant's map.	intermediate	2.5 h
9	Ride-hailing price	PI on a calendar with a quantized, rate-limited output. A gain that grows near a fold.	intermediate	2.5 h
10	National budget and public debt	PID through long lags and an annual decision. The gain the calendar allows.	advanced	2.5 h
11	Central bank policy rate	Sampled, quantized PID on late data. Windup at a lower bound.	advanced	2.5 h
12	Machine-tool servo axis	Three PI loops at three rates. A resonance that caps the gain, and what a relay test measures.	advanced	2.5 h

KEY POINTS

- A PID loop seldom fails because of its equation. It fails at an actuator limit, on noisy or late data, on a slow calendar or at a process gain that has moved since tuning.
- Both model-based tuning rules divide by the dead time, so an error in reading the dead time carries straight into the gain.
- An actuator at a limit breaks the loop, so the apps limit only the output they emit and correct only the integral.
- When the process gain moves with the operating point, a gain schedule holds the loop gain, and it is only as good as its gain curve and its measurement.

Where else this applies. Cascade in Volume 3, overrides in Volume 4, dead time in Volume 5, predictive control in Volume 9.

References

[1] K. J. Åström and T. Hägglund, "The future of PID control," *Control Engineering Practice*, vol. 9, no. 11, pp. 1163–1175, 2001.

[2] ——, *Advanced PID Control*. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2006.

[3] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, III, *Process Dynamics and Control*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.

[4] A. Visioli, *Practical PID Control*, ser. Advances in Industrial Control. London: Springer, 2006.

[5] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed. Princeton, NJ: Princeton University Press, 2021.

[6] J. C. Maxwell, "On governors," *Proceedings of the Royal Society of London*, vol. 16, pp. 270–283, 1868.

[7] M. Pitt, "Taking a look back at control: Part 1," *The Chemical Engineer*, no. 988, pp. 51–54, Oct. 2023, thechemicalengineer.com/features/taking-a-look-back-at-control-part-1.

[8] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the ASME*, vol. 64, no. 8, pp. 759–765, 1942.

[9] L. Jacobson, J. Stribiak, L. Nelson, and D. Sallman, “Ramp management and control handbook,” Federal Highway Administration, U.S. Department of Transportation, Washington, DC, Tech. Rep. FHWA-HOP-06-001, Jan. 2006.

[10] M. Pitt, “Taking a look back at control: Part 2,” *The Chemical Engineer*, no. 989, Nov. 2023, thechemicalengineer.com/features/taking-a-look-back-at-control-part-2.

[11] K. J. Åström and T. Hägglund, “Automatic tuning of simple regulators with specifications on phase and amplitude margins,” *Automatica*, vol. 20, no. 5, pp. 645–651, 1984.

[12] V. Jacobson, “Congestion avoidance and control,” in *Symposium Proceedings on Communications Architectures and Protocols (SIGCOMM ’88)*. ACM, 1988, pp. 314–329.

[13] T. Beljean and A. Geier, “The Swiss debt brake — has it been a success?” *Swiss Journal of Economics and Statistics*, vol. 149, no. 2, pp. 115–135, 2013.

[14] “Statement on longer-run goals and monetary policy strategy,” Board of Governors of the Federal Reserve System, Federal Open Market Committee. Adopted effective January 24, 2012; as reaffirmed effective January 27, 2026, 2026, accessed 2026-11-20. [Online]. Available: https://www.federalreserve.gov/monetarypolicy/files/FOMC_LongerRunGoals.pdf

[15] R. Pan, P. Natarajan, F. Baker, and G. White, “Proportional integral controller enhanced (PIE): A lightweight control scheme to address the bufferbloat problem,” RFC Editor, RFC 8033, 2017.

[16] D. E. Rivera, M. Morari, and S. Skogestad, “Internal model control: PID controller design,” *Industrial & Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252–265, 1986.

[17] S. Skogestad, “Simple analytic rules for model reduction and PID controller tuning,” *Journal of Process Control*, vol. 13, no. 4, pp. 291–309, 2003.

[18] K. J. Åström and T. Hägglund, “Revisiting the Ziegler–Nichols step response method for PID control,” *Journal of Process Control*, vol. 14, no. 6, pp. 635–650, 2004.

[19] B. D. Tyreus and W. L. Luyben, “Tuning PI controllers for integrator/dead time processes,” *Industrial & Engineering Chemistry Research*, vol. 31, no. 11, pp. 2625–2628, 1992.

[20] T. Hägglund, “A unified discussion on signal filtering in PID control,” *Control Engineering Practice*, vol. 21, no. 8, pp. 994–1006, 2013.

[21] K. J. Åström and L. Rundqwist, “Integrator windup and how to avoid it,” in *Proc. 1989 American Control Conference*, 1989, pp. 1693–1698.

[22] M. V. Kothare, P. J. Campo, M. Morari, and C. N. Nett, “A unified framework for the study of anti-windup designs,” *Automatica*, vol. 30, no. 12, pp. 1869–1883, 1994.

[23] R. Hanus, M. Kinnaert, and J.-L. Henrotte, “Conditioning technique, a general anti-windup and bumpless transfer method,” *Automatica*, vol. 23, no. 6, pp. 729–739, 1987.

[24] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. Chichester, UK: John Wiley & Sons, 2005.

[25] W. J. Rugh and J. S. Shamma, “Research on gain scheduling,” *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[26] T. R. Kuphaldt, *Lessons in Industrial Instrumentation*, version 2.33 ed. Open textbook, Creative Commons Attribution 4.0 license, ibiblio.org/kuphaldt/socratic/sinst, 2024.

[27] B. Bok, D. Caratelli, D. Giannone, A. Sbordone, and A. Tambalotti, “Macroeconomic nowcasting and forecasting with big data,” Federal Reserve Bank of New York, Staff Report 830, 2017. [Online]. Available: https://www.newyorkfed.org/medialibrary/media/research/staff_reports/sr830.pdf

[28] GPS.gov, National Coordination Office for Space-Based Positioning, Navigation, and Timing, “GPS accuracy,” 2026, accessed 2026-11-20. [Online]. Available: https://www.gps.gov/gps-accuracy

[29] “Interest on reserve balances,” Board of Governors of the Federal Reserve System, 2026, accessed 2026-11-20. [Online]. Available: https://www.federalreserve.gov/monetarypolicy/reserve-balances.htm

[30] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[31] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, “Guide to operational technology (OT) security,” National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-82r3, 2023.

[32] D. Goldfarb and A. Idnani, “A numerically stable dual method for solving strictly convex quadratic programs,” *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[33] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model-predictive control," *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[34] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE Transactions on Automatic Control*, vol. 48, no. 2, pp. 246–258, 2003.

1.9 Exercises

1.1 Concept check · 2 min

A reactor is cooled by a jacket valve. Opening the valve lowers the reactor temperature, which is the PV. The controller forms its error as $e = a\,(\mathrm{SP} - \mathrm{PV})$ with a positive gain K_c. Which setting and name are right?

(a) $a = +1$, reverse-acting, so that the output falls as the temperature rises

(b) $a = +1$, direct-acting, because the process gain is negative

(c) $a = -1$, direct-acting, so that the output rises as the temperature rises

(d) $a = -1$, reverse-acting, because the error has changed its sign

1.2 Theory and calculation · 5 min

A series (interacting) PID controller holds the settings $K_c' = 2\,\%/^\circ\mathrm{C}$, $\tau_I' = 5\,\mathrm{min}$ and $\tau_D' = 1\,\mathrm{min}$. Find the ideal-form settings K_c, τ_I and τ_D that give the same controller, and the parallel gains K_i and K_d.

1.3 Theory and calculation · 6 min

A dryer is in Manual at a steady outlet temperature of 60 °C. The heater output is stepped up by 8 % and the temperature settles at 72 °C. It passes 63.4 °C at 4.0 min after the step and 67.6 °C at 9.0 min. Fit a first-order-plus-dead-time model with the two-point method.

1.4 Theory and calculation · 8 min

Tune a controller for the model $K_p = 1.5\,^\circ\mathrm{C}/\%$, $\tau_p = 7.5\,\mathrm{min}$ and $\theta_p = 1.5\,\mathrm{min}$.

(a) Find the SIMC PI settings for $\tau_c = \theta_p$. *Checkpoint:* $K_c = 1.67\,\%/^\circ\mathrm{C}$, $\tau_I = 7.5\,\mathrm{min}$.

(b) Find the Ziegler–Nichols reaction-curve PID settings. *Checkpoint:* $K_c = 4.0\,\%/^\circ\mathrm{C}$, $\tau_I = 3.0\,\mathrm{min}$, $\tau_D = 0.75\,\mathrm{min}$.

(c) Give the phase margin and the gain margin of the loop of part (a). *Checkpoint:* 61.4° and 3.14.

1.5 Concept check · 3 min

A tank level is an integrating process and is held by a PI controller. After the integral time was shortened to speed the loop up, the level rolls slowly about its setpoint and the valve never rests. Which change is the best first correction?

(a) Shorten the integral time further, so the integral catches the error sooner

(b) Lengthen the integral time toward $4\,(\tau_c + \theta_p)$ and keep the gain

(c) Add derivative action with no filter, so the valve leads the level

(d) Change the controller from reverse to direct action

1.6 Theory and calculation · 6 min

A relay test with no switching band moves a heater by $\delta = \pm 5\,\%$. The PV settles into a cycle of $\pm 1.6\,^\circ\mathrm{C}$ with a period of 6.0 min.

(a) Estimate the ultimate gain. *Checkpoint:* $K_u = 3.98\,\%/^\circ\mathrm{C}$.

(b) Give the Tyreus–Luyben PI settings from that estimate. *Checkpoint:* $K_c = 1.24\,\%/^\circ\mathrm{C}$, $\tau_I = 13.2\,\mathrm{min}$.

(c) The recorded PV looks like a triangle wave. Is the true ultimate gain above or below the estimate, and by how much at most? *Checkpoint:* Above, by up to a factor of 1.23, about 4.9.

1.7 Theory and calculation · 6 min

A PI controller with $K_c = 1.25\,\%/°\mathrm{C}$ and $\tau_I = 10\,\mathrm{min}$ has no windup protection. Its output is held at the 100 % limit for 12 min while the error averages 20 °C.

(a) By how much does the integral term grow during that time? *Checkpoint:* 30 %.

(b) The PV then passes the setpoint and stays 5 °C above it. How long does the integral take to give back what it stored? *Checkpoint:* 48 min.

1.8 Concept check · 3 min

A PI controller with $K_c = 2\,\%/°\mathrm{C}$ and reverse action is in Manual with its output at 40 %. The PV is 6 °C below the setpoint. In Manual the algorithm tracks: at each execution it sets its integral so that the output it computes equals the manual output. The loop is switched to Auto. Which value is closest to the first automatic output?

(a) 40 %, the manual output

(b) 52 %, the manual output plus the proportional term

(c) 28 %, the manual output less the proportional term

(d) 100 %, because the integral grew while the loop was in Manual

1.9 Theory and calculation · 7 min

The oven of this chapter has $K_p = 2°\mathrm{C}/\%$, $\tau_p = 10\,\mathrm{min}$ and $\theta_p = 2\,\mathrm{min}$. It runs the SIMC PI with $\tau_c = \theta_p$, so $K_c = 1.25\,\%/°\mathrm{C}$ and $\tau_I = 10\,\mathrm{min}$.

(a) The controller is moved to a system that executes once a minute. Use the half-period rule to estimate the phase margin with the settings unchanged. *Checkpoint:* About 54°.

(b) Retune with the effective dead time and τ_c equal to it. *Checkpoint:* $K_c = 1.0\,\%/°\mathrm{C}$, $\tau_I = 10\,\mathrm{min}$.

1.10 Concept check · 3 min

The oven loop runs the SIMC PI with $\tau_c = \theta_p$. At a new operating point the process gain is twice the fitted value, and the time constant and the dead time are unchanged. What follows?

(a) The loop is unstable, because the loop gain has doubled

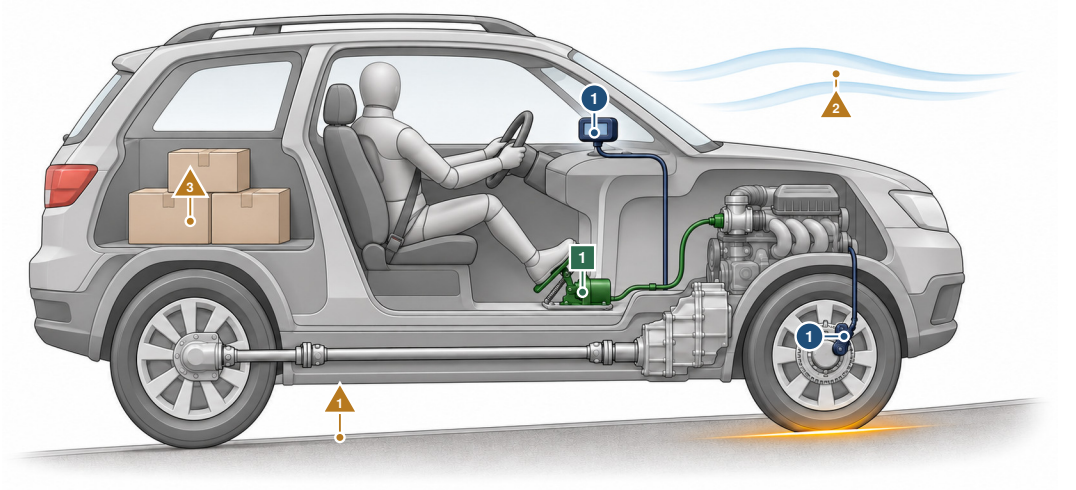
(b) Nothing changes, because integral action absorbs a change of process gain

(c) The loop becomes slower, because the closed-loop time constant doubles

(d) The loop is stable with a thin phase margin, and half the controller gain restores it

CHAPTER 2

Automobile Cruise Control Simulator for Velocity-Form PID Under Quadratic Drag



The controller moves 1 a drive that moves the gas pedal

The controller reads 1 the speedometer: a pickup at the wheel and its readout

What disturbs it 1 the slope of the road 2 a headwind 3 passengers and cargo

Notice that the controller never measures the hill, the wind or the cargo, and learns of each only from the speed the car loses. Illustration, not to scale.

IN PLAIN TERMS Holding a Car's Speed as the Road Changes

The main idea. A cruise controller holds a car at a chosen speed by moving the gas pedal. It reads the speed and presses harder when a hill, a headwind or a heavy load slows the car.

The challenge. Air pushes back harder the faster the car goes, so one push on the pedal buys less speed at high speed than at low speed. When the pedal is already on the floor, the controller must not keep asking for more, or the car shoots past its target once the load eases.

Why it matters. A test made at one speed tells only part of the story at another speed.

THREE WAYS TO READ THIS CHAPTER

Explore	12 min	the first page, the fact on how common PID control is (Section 2.1), the history of governors (Section 2.2), the history of tuning by experiment (Section 2.5.6), the fact on the offset and the try-it segments of cases 4 and 6 (Section 2.6) and the key points
Study	70 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	20 min	the first page, the results (Section 2.6), Technology in Practice (Section 2.9), Limitations (Section 2.10) and the open challenges

The control question Aerodynamic drag grows with the square of road speed, so the process gain and the time constant of a cruise control loop both fall as the car goes faster. What does a step test at one cruise speed predict about the loop at another, and what does a velocity-form PID do when the pedal reaches its stop?

Learning objectives

- Derive the steady pedal map and the linearized gain and time constant from the momentum balance.
- Compute the natural frequency and damping ratio of the PI loop, and explain why only the damping moves with speed.
- Identify a first-order model from a pedal step test and tell its secant gain from the tangent gain.
- Predict the offset of a proportional loop and the integrated error of a PI loop for a road grade.
- Decide whether a setpoint is reachable, and describe how back-calculation bounds windup when it is not.

Prerequisites Sections 1.4.2 and 1.6.2 (velocity form), 1.4.4 (step test), 1.4.9 (windup) and 1.4.13 (a gain that moves with the operating point). Sections 1.4.7 and 1.6.3 are needed only for the relay test and the predictive controller.

The app https://apmonitor.com/apps/speed/

See also Volume 1, Chapter 2 and Volume 1, Chapter 3 for step-test identification on tanks whose model moves with the level or the lineup. Volume 7, Chapter 3 for a plant that the setpoint chooses, Volume 7, Chapter 4 for braking at the friction peak and Volume 7, Chapter 14 for cruise loops in a string.



2.1 Introduction

Speed regulation on a road vehicle is longitudinal vehicle control in its simplest form, a single loop closed around a road speed measurement with the throttle as its actuator [1]. A driver holding a fixed throttle operates open loop against a load that changes with every rise in the pavement and every gust of wind, and the correction is applied by a foot with no calibration and a reaction time of several hundred milliseconds. What separates this loop from other single-input single-output regulators is the form of that load.

Aerodynamic drag on a road vehicle is proportional to the square of the air speed relative to the body and is the dominant resistance at highway speeds [2]. A quadratic load makes the plant nonlinear in a form compact enough to treat in closed form. The steady pedal position required to hold a speed grows as the square of that speed, which means the incremental speed produced by one extra percent of pedal falls in inverse proportion to the speed already being held. A step test performed at 10 m/s and a step test performed at 35 m/s on the same vehicle therefore return process gains that differ by a factor of three. A process gain that moves with the operating point is the standard motivation for gain scheduling [3], which Section 1.4.13 introduces. The inertia term of this plant carries the same factor, so the identified time constant falls by the same three, and the two changes very nearly cancel inside the loop.

Simulators used to teach cruise control usually linearize the drag term or present the vehicle as a first-order lag with fixed constants [4]. However, a fixed first-order lag removes the property that makes the loop worth studying, because every step test then returns the same answer and no evidence remains that the operating point matters. This chapter works with a browser cruise control simulation that retains the quadratic drag term in full. It quantifies what a step test at one cruise speed does and does not predict about the loop at another.

The experiments built on this model exercise an identifiable set of process dynamics and control principles [4, 5]. The pedal ladder and the step tests show first-order self-regulating dynamics and the difference between a secant gain and a tangent gain on a nonlinear plant. The three disturbances separate a load change that moves the steady state from one that moves only the dynamics. Switching integral action off shows the offset that proportional action alone leaves and its dependence on the open-loop gain, and integral action removes that offset at a fixed cost in integrated error. The saturated

runs show actuator limits and anti-windup in a sampled controller. The physical fundamentals are equally compact, because the model is one momentum balance on a rigid body, the driving forces are a tractive force set by the actuator and two resistances set by the operating point, and the single time scale is the ratio of the mass to the slope of the drag curve. Gain variation with throughput is a routine commissioning problem in engine, compressor and pump control, where a loop tuned at one operating point oscillates at another, so the question this chapter poses is one a practicing engineer answers repeatedly [6].

> **DID YOU KNOW? PID Runs Most Loops**
>
> More than 95 % of all industrial control problems are solved by PID control, and many of those controllers are in fact PI controllers, because derivative action is often not included [6]. The derivative time of this chapter's cruise loop also defaults to zero.

The chapter is organized as follows. Section 2.2 places the speed loop inside adaptive cruise control and recalls the longitudinal force balance. Section 2.3 states the one-state vehicle model, with the force balance, pedal limits, instrument chain and integration scheme in enough detail to be rebuilt from scratch. Section 2.4 treats the operating-point dependence of this plant in closed form. It shows that the ratio of process gain to time constant is the thrust-to-mass ratio and carries no speed at all. As a consequence, a proportional-integral (PI) loop on this vehicle runs at a natural frequency fixed by the integral gain, the thrust parameter and the mass, with the whole speed dependence pushed into the damping ratio. Section 2.5 describes the velocity-form PID, a linear model predictive controller offered beside it, a bounded moving horizon estimator that can replace that controller's filter, and a relay experiment that writes PI constants into the PID. Section 2.6 reports runs in which one disturbance is applied at three cruise speeds and at two vehicle masses under a single set of gains, followed by saturation, setpoint and noise experiments and the shipped cases, each read out of the exported record. Section 2.7 compares the predictive controller with the PID on the shipped cases and separates what a disturbance estimate and a map of the road ahead each contribute. Section 2.8 measures the estimator and the relay against the shipped choices. Section 2.9 sets the loop in its real sensor, final element and controller platform, Sections 2.10 and 2.11 state the limitations of the model and its economic and safety context, Section 2.12 summarizes the results, and the exercises follow in Section 2.13.

2.2 Background

Speed regulation on a road vehicle started as a mechanical limit rather than as a setpoint. A governor acting on the throttle linkage caps the engine speed, and therefore the road speed in a given gear on a given grade, but it holds nothing constant, because the road load never enters the mechanism. Electronic cruise control replaced the cap with a regulator by closing a loop around a road speed measurement and positioning the throttle to hold a driver-selected value. The structure of that loop is a single-input single-output regulator on a plant whose two principal disturbances, grade and wind, are not measured, so integral action is the only route to zero offset [1].

The limitations of a speed regulator that cannot see the road ahead drove the next thirty years of work. Conventional cruise control holds a speed regardless of what is in front of the vehicle, and a comparison of that behavior against a controller that also regulates the spacing to a lead vehicle set out the requirements for an autonomous intelligent cruise control [7]. Adaptive cruise control, which switches between a speed loop and a spacing loop depending on which constraint is active, has since become the most widely deployed longitudinal automation on production vehicles, and the development of its sensing, its control laws and its switching logic has been reviewed in detail [8]. Chaining such vehicles together raises a question that a single loop does not, because a string of controllers each following its predecessor can amplify a disturbance as it propagates backward. The PATH program built and instrumented platoons to study exactly that, together with the lateral

Table 2.1 Classification of Variables. Nominal Values Are Those of the 25 m/s Cruise Point on a Flat Road With the Default 200 kg Load.

Symbol	Role	Nominal	Range
v	state, true road speed	25.00 m/s	≥ 0
$\hat{v}$ (SI)	CV, measured speed	25.00 m/s	≥ 0
v_{SP}	setpoint	25.0 m/s	0 to 45, step 0.5
u	MV, pedal position	15.31 %	−50 to 100, step 1
m_L	DV, passengers and cargo	200 kg	0 to 800, step 10
s	DV, road grade	0 %	−10 to +10, step 0.5
w	DV, headwind	0 m/s	−20 to +20, step 1
σ_n	DV, speedometer noise	0.08 m/s	0 to 0.50
v_r	relative air speed	25.00 m/s	—
x	output, distance traveled	0 m	≥ 0
saturated	output, pedal-at-limit flag	false	—

automation needed to make them practical [9]. Volume 7, Chapter 14 treats the string stability of such a platoon. In every one of these architectures the speed loop of this chapter is the inner loop, and its tuning sets what the outer loop is allowed to ask for.

The modeling side of the problem was settled earlier and has changed less. The longitudinal force balance on a vehicle is a momentum balance with three resistances, aerodynamic drag, rolling resistance and the component of weight along the road, and the tractive force available to oppose them is set by the powertrain and limited by tire adhesion [1]. Drag is written as one half the air density times a drag coefficient times a frontal area times the square of the relative air speed, and the measurement of that coefficient in wind tunnels and by coastdown testing is a mature discipline [2]. Combining the road load with a powertrain model gives the fuel consumption of a driving cycle, which is the form in which the same balance appears in propulsion system design [10]. The simulator of this chapter uses this balance with the rolling resistance term deliberately removed, for the reason given in Section 2.3.4.

> ⧖ HISTORY **Moderators and Governors, 1868**
>
> In 1868 James Clerk Maxwell set out the dynamical theory of the governor, the part of a machine that keeps its speed nearly uniform against changes in the driving power or the resistance. He separated two kinds. In the first, the regulator adds a resistance that depends on the speed itself, and more driving power leaves the speed permanently higher. Such regulators, he wrote, should be called moderators rather than governors. In the second kind, the speed error moves an independent piece that keeps adding resistance for as long as the speed stays above its normal value, and the machine returns to the same normal speed whatever the change in power or resistance [11]. The moderator acts like the proportional loop of this chapter, the governor like the loop with integral action.

2.3 Vehicle Model

The simulated vehicle is a single rigid mass traveling along a road of adjustable grade. The controlled variable is the road speed reported by the speedometer and the manipulated variable is the gas pedal position, which spans full braking to full throttle on one continuous scale. Cargo mass, road grade and headwind stay under operator control whether the loop is open or closed, and none of the three is measured, so all three are unmeasured load disturbances, although the predictive controller of Section 2.5.4 can be given the grade changes a case schedules, as a map would supply them. The loop has a single state, a single actuator and no cascade. Figure 2.1 shows each symbol at the point on the vehicle where the equations apply it.

Each signal is assigned its role in Table 2.1. Speed is carried in m/s, both pedal position and grade in percent, and time in seconds, all of them SI. A single differential equation is integrated and a single

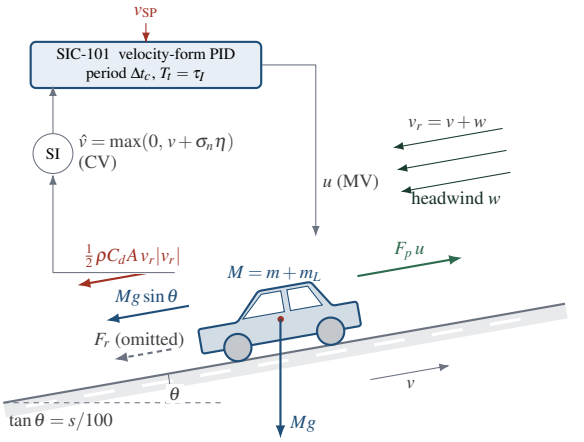


Figure 2.1 Longitudinal force balance on the simulated vehicle. The pedal produces a tractive force proportional to its position through the thrust parameter F_p, aerodynamic drag acts on the relative air speed v_r rather than on the road speed v, and a grade of s percent adds the component $Mg\sin\theta$ of the total weight along the road. Rolling resistance F_r is drawn for completeness and is not part of the model. Controller SIC-101 reads the noisy speedometer output $\hat{v}$ and positions the pedal.

algebraic bound is imposed on its solution, because clipping the road speed at zero after every step is what keeps the vehicle from rolling backward.

2.3.1 Momentum Balance

The vehicle is a point mass driven by a tractive force proportional to pedal position and resisted by aerodynamic drag and by the along-road component of its own weight, as shown in Equation (2.1).

$$M\frac{\mathrm{d}v}{\mathrm{d}t} = F_p\,u \;-\; \tfrac{1}{2}\rho\, C_d A\, v_r |v_r| \;-\; M g \sin\theta, \qquad v \geq 0 \tag{2.1}$$

where $M = m + m_L$ is the total moving mass in kg, m is the empty vehicle mass, m_L is the passenger and cargo load, F_p is the thrust parameter in N per percent of pedal, u is the pedal position in percent, ρ is the air density in kg/m^3, C_d is the drag coefficient, A is the frontal area in m^2, g is gravitational acceleration and θ is the road inclination. Both sides carry units of N. The relative air speed and the road inclination are the two algebraic definitions of Equation (2.2).

$$v_r = v + w, \qquad \theta = \arctan\!\left(\frac{s}{100}\right) \tag{2.2}$$

where w is the headwind in m/s, positive against the vehicle, and s is the road grade in percent rise, positive uphill. Writing the drag term as $v_r|v_r|$ rather than v_r^2 keeps the sign of the relative air speed, so a tailwind stronger than the road speed pushes the vehicle along instead of holding it back. Grade is entered as a percentage rather than as an angle because that is how road signs and survey data report it, and the arctan makes the two definitions agree exactly rather than to first order.

Grouping the three drag constants into a single coefficient $k = \frac{1}{2}\rho C_d A$ shortens every expression that follows. With the default values $k = 0.735\,\mathrm{kg/m}$, so the drag on the flat at 25 m/s is 459.4 N and the propulsive power needed to hold that speed is 11.5 kW. Setting the derivative in (2.1) to zero on a flat road with no wind gives the two steady-state maps of Equation (2.3).

$$\bar{u} = \frac{k\,\bar{v}^2}{F_p}, \qquad \bar{v} = \sqrt{\frac{F_p\,\bar{u}}{k}} \tag{2.3}$$

Equation (2.3) is the whole source of the nonlinearity. Neither expression contains the mass, so cargo changes how quickly the vehicle reaches a speed and not which speed it reaches. Holding 25 m/s on the flat requires $\bar{u} = 15.31\,\%$, holding 35 m/s requires 30.01 %, and the pedal at 100 % would settle at 63.9 m/s, which is well above the 45 m/s top of the setpoint range.

Pause. Try Exercise 2.2 before reading on.

2.3.2 Actuator and Measurement

The pedal is one signed variable rather than two, with negative values representing the brake. The applied position is clipped to $u \in [-50, 100]$ percent, so the tractive force is bounded by 3000 N at full throttle and −1500 N at full brake. On the flat with the default mass this gives 4.29 m/s^2 of acceleration from rest and −2.80 m/s^2 of deceleration from 25 m/s, the second of which is about 28.5 % of gravity and is therefore a weak brake by the standards of a real vehicle. Braking is modeled as a force independent of speed and independent of load transfer, and the clip at $v \geq 0$ is what stops the vehicle rather than the brake releasing.

What the controller sees is never the state. Equation (2.4) is the speedometer output that reaches it.

$$\hat{v} = \max(0,\ v + \sigma_n\,\eta) \tag{2.4}$$

with σ_n the standard deviation in m/s and η a fresh standard normal deviate, generated by a Box-Muller pair from the application's seeded random source once for each execution of the controller. The default $\sigma_n = 0.08\,\mathrm{m/s}$ is 0.29 km/h and is white at the 10 Hz execution rate. The instrument chain has no lag, no quantization, no bias and no dependence on tire radius. The vehicle animation and its speed readout are driven by the true state v while the speed trend and the controller are driven by $\hat{v}$, so the contribution of the instrument is visible against the physics during a run.

Table 2.2 carries every constant of the plant and the PID that a reimplementation needs, and Section 2.5.4 gives the remaining ones of the predictive controller. Parameter entries are bounded before they reach the model, because m is floored at 1 kg, F_p at machine epsilon and C_d, A, ρ and σ_n at zero, and the cargo load is floored at zero before the two masses are added. The default vehicle is light and has a large drag area, because $m = 500\,\mathrm{kg}$ and $C_dA = 1.2\,\mathrm{m}^2$ together compress the whole speed range into a response that can be observed in real time, and all six vehicle constants are operator adjustable so that a more representative vehicle can be entered. The top of the integral-time range, 120 s, switches integral action off. Every case starts at steady operation, so $v(0)$ is the steady speed of the setpoint in force in automatic, or of the pedal in force in manual, and $u(0)$ is the pedal that holds it.

2.3.3 Numerics and Clocks

The app supplies its own integration step in place of the Runge–Kutta default of the framework (Section 1.6.1). Explicit Euler on a fixed 0.01 s step advances Equation (2.1). A higher-order integrator returns nothing on this right-hand side, because the pedal is held constant between controller executions and the speed is clipped at zero, and those two discontinuities set the error floor rather than the smooth drag term. Five Euler steps make one simulation step of 0.05 s, and the odometer that drives the animation accumulates $\Delta t\, v$ inside the same loop so that the road markings and the wheels scroll at the true road speed. Two simulation steps make one controller execution of 0.1 s, and one row joins the historian every 0.1 s, holding the true speed at its own time stamp, the measured speed that the controller execution at that stamp acts on, and the pedal held over the 0.1 s just ended. The engine keeps these clocks apart from the drawing of the screen, as Section 1.6.1 describes, so the speed multiplier changes the pace of a session and not its result. The noise generator restarts from the same seed whenever a case is loaded or Reset or a setting is changed before Run, so a session at 10× repeats a session at 1× sample for sample with the speedometer noise on or off, and the exported record advances by one control period per row.

Table 2.2 Model and Controller Parameters. The First Block Is Operator Adjustable, the Second Is Fixed in the Source.

Symbol	Meaning	Default	Range	Unit
Operator adjustable				
m	empty vehicle mass	500	300–2000	kg
C_d	drag coefficient	0.24	0.10–0.60	—
A	frontal area	5.0	2.0–8.0	m^2
ρ	air density	1.225	0.80–1.50	kg/m^3
F_p	thrust parameter	30	10–60	N/%
σ_n	speedometer noise	0.08	0–0.50	m/s
m_L	passengers and cargo	200	0–800	kg
s	road grade	0	−10–+10	%
w	headwind	0	−20–+20	m/s
v_{SP}	speed setpoint	25	0–45	m/s
u	manual pedal	15.31	−50–+100	%
K_c	proportional gain	6	0–30	% s/m
τ_I	integral time	12	0.05–120	s
τ_D	derivative time	0	0–10	s
T_p	MPC prediction horizon	80	10–160	s
κ	MPC move suppression	0.3	0–3	—
—	MPC estimator	Kalman	Kalman, MHE	—
—	speed multiplier	1	0.5–10	—
Fixed in the source				
g	gravitational acceleration	9.80665	—	m/s^2
Δt	integration step (Euler)	0.01	—	s
—	simulation step	0.05	—	s
Δt_c	controller sample period	0.1	—	s
u_b	controller bias	$u(0)$	—	%
T_t	anti-windup tracking time	τ_I	—	s
T_s	MPC decision period	0.5	—	s
N_e	MHE window	30	—	executions
h	relay amplitude	7.5	—	%
ε	relay switching band	0.24	—	m/s
—	historian rate	10	—	samples/s
$v(0)$	initial road speed	$\bar{v}$	—	m/s

2.3.4 Assumptions

Only the mechanisms that alter a tuning decision survive into the model. Each item below names what has been left out and the operating region in which the omission starts to matter.

1. *Rolling resistance is absent.* The only speed-dependent resistance is aerodynamic. A rolling resistance coefficient of 0.01 on the default vehicle would contribute $f_r Mg = 68.6\,\text{N}$, which equals the aerodynamic drag at 9.7 m/s and is 15 % of it at 25 m/s. The omission therefore matters most at low speed, where it makes the simulated vehicle coast much further than a real one, and it removes the one road-load term that does not vanish as the vehicle stops.

2. *Thrust is linear in pedal position and instantaneous.* There is no engine map, no gearbox, no torque converter, no turbocharger lag and no traction limit, so F_p is the same at 5 % pedal and at 100 % pedal and at every speed. A real powertrain delivers a tractive force that falls with speed in each gear and steps at every shift, which adds a second operating-point dependence on top of the drag one and introduces dead time of the order of a few hundred milliseconds [10].

3. *Braking is negative thrust.* Negative pedal produces a force of $F_p u$ with no dependence on speed, no separation between engine braking and friction braking, no wheel lock and no anti-lock intervention. The floor at $v \geq 0$ is what brings the vehicle to rest.

4. *Constant drag coefficient.* C_d does not vary with Reynolds number, with yaw angle or with the presence of the headwind. A crosswind component changes both the effective frontal area and the drag coefficient of a real body, and measured coefficients move appreciably with yaw angle [2]. The headwind input therefore models a purely head-on or tail-on wind, which is the only case in which the relative air speed is a scalar sum.

5. *Point mass on a rigid road.* There is no pitch, no suspension, no load transfer, no tire slip and no rotational inertia of the wheels and driveline. Wheel and driveline inertia typically add several percent to the effective mass in the lowest gears and would appear in the model as an increase in M that depends on gear.

6. *Grade and wind are step-changeable and spatially uniform.* The road has one slope at a time, applied everywhere at once, and the wind has one speed. A real grade arrives progressively as the vehicle enters it, which is a ramp rather than a step and is easier for the loop to reject.

7. *Ideal speedometer apart from white noise.* The sensor has no lag, no bias, no quantization and no dependence on tire radius or wheel slip, and the noise is white and Gaussian. A wheel-speed measurement on a real vehicle is biased by tire wear and inflation and is quantized by the tooth count of the encoder, and neither error is represented.

8. *Omitted entirely.* Fuel consumption, engine speed, gear selection, road curvature, a lead vehicle, spacing control and every lateral degree of freedom are missing. None of them alters the drag-driven gain variation this chapter is built around, which is why each was left out.

2.4 Dynamic Analysis

2.4.1 Linearization About a Cruise Point

A first-order Taylor expansion of Equation (2.1) around a steady cruise point $(\bar{v}, \bar{u}, \bar{w}, \bar{s} = 0)$ produces the deviation model of Equation (2.5).

$$\tau \frac{\mathrm{d}v'}{\mathrm{d}t} + v' = K_u u' + K_w w' + K_s s' \tag{2.5}$$

where the primes denote deviations from the cruise point and the four constants are given by Equation (2.6).

$$\tau = \frac{M}{2k\bar{v}_r}, \quad K_u = \frac{F_p}{2k\bar{v}_r}, \quad K_w = -1, \quad K_s = -\frac{g\,\tau}{100} \tag{2.6}$$

with τ the process time constant in s, K_u the process gain in m/s per percent of pedal, K_w dimensionless and K_s in m/s per percent of grade. The three gains are not interchangeable and each carries a different piece of the physics. K_u and τ both carry the factor $1/(2k\bar{v}_r)$, so both shrink as the cruise speed rises. K_w is exactly -1 and is not an approximation at all, because the drag term depends on $v + w$ alone, so at a fixed pedal position the sum $v + w$ is fixed and a headwind removes its own magnitude from the road speed whatever the speed, the mass or the drag parameters happen to be. K_s inherits the time constant, so a hill costs more speed on a heavy vehicle and on a slow one.

Evaluated at 25 m/s with the default 700 kg, the linearized vehicle carries $K_u = 0.816\,\mathrm{m/(s\,\%)}$ against $\tau = 19.05\,\mathrm{s}$, and evaluated at 35 m/s the same vehicle carries $K_u = 0.583\,\mathrm{m/(s\,\%)}$ against $\tau = 13.61\,\mathrm{s}$. The two descriptions differ only in the speed at which each was taken.

🕮 **Pause.** Try Exercise 2.1 before reading on.

2.4.2 Invariance of the Gain to Time Constant Ratio

The drag coefficient, the frontal area, the air density and the cruise speed all cancel when the gain in (2.6) is divided by the time constant, as shown in Equation (2.7).

$$\frac{K_u}{\tau} = \frac{F_p}{M} = 0.0429\,\mathrm{m/(s^2\,\%)} \tag{2.7}$$

The right-hand side of Equation (2.7) is Newton's second law with the drag term deleted, and it is the initial slope of the speed response to a pedal step. Every step test on this vehicle therefore leaves the starting point at the same acceleration per percent of pedal, and what separates one test from another is the final speed and the time taken to arrive at it. Measuring that initial slope returns the thrust-to-mass ratio alone, and waiting for the response to settle returns the drag.

2.4.3 Closed-Loop Dynamics and Steady-State Error

Substituting the deviation model (2.5) into a proportional-integral control law and clearing the denominator gives the characteristic equation of the loop, as shown in Equation (2.8).

$$M\lambda^2 + (2k\bar{v}_r + F_pK_c)\,\lambda + F_pK_I = 0 \tag{2.8}$$

where λ is the complex frequency in 1/s and $K_I = K_c/\tau_I$ is the integral gain of the parallel form (Section 1.4.2), 0.5 %/m at the default $K_c = 6\,\%\,\mathrm{s/m}$ and $\tau_I = 12\,\mathrm{s}$. Reading the natural frequency and the damping ratio off Equation (2.8) gives Equation (2.9).

$$\omega_n = \sqrt{\frac{F_pK_I}{M}}, \qquad \zeta = \frac{2k\bar{v}_r + F_pK_c}{2\sqrt{F_pK_IM}} \tag{2.9}$$

Equation (2.9) is the central result for tuning this loop. The natural frequency does not contain the cruise speed at all, so the closed loop runs at $\omega_n = 0.146\,\mathrm{rad/s}$ with the default gains whether the vehicle is doing 10 or 35 m/s. The damping ratio does contain the cruise speed, and it rises linearly with it, from 0.950 at 10 m/s through 1.058 at 25 m/s to 1.129 at 35 m/s. A threefold change in process gain would ordinarily leave a controller commissioned at one end of the range unusable at the other, which does not happen on this plant. What transfers between operating points is the speed of the response, and what does not transfer is the damping and, as shown next, the offset that proportional-only control leaves behind.

Two closed-loop invariants follow directly and are worth stating separately because both are exact rather than approximate. A load force d acting along the road is rejected in steady state by a pedal increment d/F_p, whatever the speed and whatever the mass, because thrust is linear in pedal position. The integral of the error over the whole rejection is then fixed by the integral gain alone, as shown in Equation (2.10).

$$\int_0^\infty e\,\mathrm{d}t = \frac{d}{F_p\,K_I} \tag{2.10}$$

where $e = v_{\mathrm{SP}} - \hat{v}$ in m/s and the integral carries units of m. Equation (2.10) holds because the only term in the velocity-form recursion that can supply a permanent output change is the integral increment, so the accumulated error is whatever it takes to produce d/F_p of pedal. The corresponding offset when integral action is switched off is given by Equation (2.11).

$$e_\infty \simeq \frac{d}{2k\bar{v}_r + F_pK_c} \tag{2.11}$$

Equation (2.11) is the only one of the four that is approximate, because the drag term has been linearized, and it shares its denominator with Equation (2.8). A hill therefore costs a proportional loop less speed at high cruise speed than at low, which is the opposite of what the falling process gain alone would suggest, and is again a consequence of the drag itself providing part of the restoring action.

Table 2.3 Step Tests With a 2 % Pedal Increment From 2, 15 and 30 % and $\sigma_n = 0$. The Predicted Columns Come From Equation (2.6) Evaluated at the Starting Speed.

$\bar{v}$ (m/s)	m_L (kg)	$\hat{K}_u$ (m/(s %))	K_u	$\hat{\tau}$ (s)	τ	R^2
9.04	200	1.871	2.259	41.09	52.70	0.99968
24.74	200	0.799	0.825	18.42	19.25	0.99999
34.99	200	0.574	0.583	13.30	13.61	1.00000
24.74	0	0.799	0.825	13.16	13.75	0.99999
24.74	600	0.799	0.825	28.95	30.24	0.99999

2.4.4 Identification From Step Tests

Pedal step tests run by hand in manual mode, with the speedometer noise turned off, are collected in Table 2.3. Section 1.4.4 gives the step test and its fit in general. The application accepts the pedal in whole percent, so each test sets the pedal before Run at the whole percent nearest the value that (2.3) predicts for 10, 25 and 35 m/s (2, 15 and 30 %), which starts the case at the steady speed of that pedal, adds 2 % 10 s into the run and waits 600 s. The gain is the observed speed change divided by the pedal change and the time constant is the time to 63.2 % of that change, both taken from the true speed in the exported comma-separated file, and R^2 belongs to a two-parameter first-order fit to the same record. An unstepped run from the same start does not move over the window, so no drift correction is needed.

The identified gain falls by a factor of 3.26 between 9.04 and 34.99 m/s against the factor of 3.87 predicted by the ratio of starting speeds, and the identified time constant falls by 3.09 against the same prediction. Every fitted value falls short of the linearized prediction, and the shortfall widens as the vehicle slows, from 1.6 % of the gain at 34.99 m/s to 17 % of it at 9.04 m/s. The mechanism is the same one the table exists to demonstrate. A 2 % pedal step at 9.04 m/s raises the speed by 3.74 m/s, which is 41 % of the starting value, so the plant the response is traveling through has a lower gain and a shorter time constant than the plant the linearization describes, and the secant necessarily lands below the tangent. At 34.99 m/s the same step is a 3 % excursion and the two agree to within 2.3 %. The residual behaves the same way, with R^2 rising from 0.99968 to 1.00000 as the fractional excursion shrinks.

Pause. Try Exercise 2.3 before reading on.

The last two rows isolate the mass. Loading 600 kg of cargo instead of none more than doubles the identified time constant, from 13.16 s to 28.95 s, a ratio of 2.201 against the mass ratio of 2.200, while the identified gain is unchanged at 0.799 m/(s %) in all three rows. That result is Equation (2.3) restated as a measurement, because the steady-state map contains no mass and the dynamic term contains nothing else. The gravity-drained tank of Volume 1, Chapter 2 has the same structure, a gain and a time constant that move together with the operating point at a fixed ratio.

Keeping three descriptions of the same vehicle apart matters more here than it usually does. Equation (2.1) is the simulation model, nonlinear and exact inside its own assumptions. The pair $(\hat{K}_u, \hat{\tau})$ from Table 2.3 is the identified control model, good near one cruise speed and one mass and nowhere else. The PID of Section 2.5 holds no model whatever, only three tuning constants and a sample period, so the operating point has to be tracked by whoever chose those numbers. The predictive controller of Section 2.5.4 carries such a pair inside the loop, fitted once at 25 m/s and never refitted.

The guide supplied with the application quotes $K_u \approx 0.76\,\mathrm{m/(s\,\%)}$ and $\tau \approx 17.3\,\mathrm{s}$ for a 5 % pedal step at 25 m/s, beside the tangent values 0.82 and 19 s of Equation (2.6). Raising the pedal of the first case from 15.31 to 20 %, the nearest step the whole-percent pedal allows, reproduces them at 0.762 and 17.34 s, and the 2 % step of Table 2.3 does not. Both pairs are correct measurements of

the same vehicle. The difference between them is the size of the excursion, and it is the cheapest available demonstration that a secant gain on a nonlinear plant is a property of the test as much as of the process.

2.5 Controller Design

2.5.1 Velocity Form

The PID is the velocity-form algorithm of the framework, which Section 1.6.2 gives in general, and this subsection writes it in the symbols of the speed loop. The algorithm produces a change in pedal position each cycle rather than a position, which is the velocity or incremental form [12], with the proportional term acting on the error and the derivative term on the measurement. Equation (2.12) defines the error as setpoint minus measurement, and Equations (2.13) to (2.15) give the three increments.

$$e_k = v_{\mathrm{SP}} - \hat{v}_k \tag{2.12}$$

$$\Delta u_k^P = K_c\,(e_k - e_{k-1}) \tag{2.13}$$

$$\Delta u_k^I = \frac{K_c}{\tau_I}\, e_k\, \Delta t_c \tag{2.14}$$

$$\Delta u_k^D = -\,K_c \tau_D\, \frac{\hat{v}_k - 2\hat{v}_{k-1} + \hat{v}_{k-2}}{\Delta t_c} \tag{2.15}$$

with K_c measured in % s/m, τ_I and τ_D in seconds, and a sample period Δt_c of 0.1 s. The derivative acts on the unfiltered measurement. Combining the increments, clipping the demand to the pedal travel and correcting the integral gives the new output, as shown in Equation (2.16).

$$\begin{aligned} s_k &= s_{k-1} + \Delta u_k^P + \Delta u_k^D \\ c_k &= u_b + s_k + I_{k-1} \\ u_k &= \min(100,\ \max(-50,\ c_k)) \\ I_k &= I_{k-1} + \Delta u_k^I + \frac{\Delta t_c}{T_t}\,(u_k - c_k) \end{aligned} \tag{2.16}$$

Here s_k accumulates the proportional and derivative increments without limit, c_k is the output the algorithm demands, u_b is the bias, and I_k carries the integral contribution, so the integral used at scan k was accumulated through scan $k-1$. The last line is back-calculation with the tracking time $T_t = \tau_I$, and it is the only place the clip enters the recursion. Away from both limits $u_k = c_k$, the correction vanishes, and the recursion is algebraically identical to the positional law with the same three constants. On a limit it is not, which is the purpose of the correction. Keeping s_k unclipped leaves the proportional expression telescoping exactly, and Section 1.4.9 shows what clipping the running sum would cost. In the notation of Equations (1.23) to (1.26), s_k is the running sum S_k and c_k is the output before limiting v_k. The code forms the proportional increment as the difference of the whole terms $K_c e$ at consecutive scans, so a gain change moves the pedal by the change in gain times the current error and is bumpless only on a settled loop (Section 1.4.10). The loop is reverse acting in the ISA sense, because pressing the pedal raises the speed, so a speed above setpoint has to back the pedal off (Section 1.4.3). No action switch is offered.

2.5.2 Initialization and Transfer Between Modes

A velocity-form recursion has no bias term of its own, so where the recursion starts decides the operating point it holds. Whenever a case is loaded or Reset, or a setting is changed before Run, the application puts the vehicle at steady operation, at the steady speed of the setpoint in automatic or of the manual pedal in manual, sets u_b to the pedal that holds it, and sets s and the integral to zero. A case that starts in automatic therefore applies the whole proportional term $K_c e$ at its first

execution, which is the positional law with the steady pedal as its bias, and on a settled start that error is the noise of the first reading alone. Transfer between manual and automatic is bumpless in both directions by the tracking of Section 1.6.2. In manual mode the controller keeps executing and sets $I_k = u_{\mathrm{man}} - u_b - s_k$ after every scan, so its demanded output equals the pedal the driver holds. The first automatic execution therefore moves the pedal only by the change in the proportional and derivative terms since the last manual scan. Telescoping Equation (2.16) from the transfer gives Equation (2.17) for constant gains and an output off both limits.

$$u_k = b + K_c\, e_k + \frac{K_c}{\tau_I}\,\Delta t_c \sum_{j=k_0}^{k-1} e_j - K_c\,\tau_D\,\frac{\hat{v}_k - \hat{v}_{k-1}}{\Delta t_c} \tag{2.17}$$

where k_0 is the first automatic execution and $b = u_0 - K_c e_0 + K_c \tau_D (\hat{v}_0 - \hat{v}_{-1})/\Delta t_c$ is fixed by the pedal u_0, the error e_0 and the speed change at the last manual scan. Equation (2.17) is the positional law with an effective bias b set by the moment of transfer rather than by an operator entry. Setting τ_I to the 120 s top of its range switches integral action off and freezes I_k, which leaves a proportional law whose bias is whatever $u_b + I_k$ held at that moment, and whose offset the integral no longer removes. With integral action on, $u_b + I_k$ settles on the flat at 25 m/s at the 15.31 % of Equation (2.3), which is the bias that a positional law needs at this operating point [4].

2.5.3 Saturation and Windup

Anti-windup is the back-calculation term of Equation (2.16) [13], one of the remedies that Section 1.4.9 compares. While the pedal sits on a stop u_{lim}, the correction drives $u_b + I_k$ toward the value that reproduces the applied pedal. Because $T_t = \tau_I$, the integral increment $(K_c/\tau_I) e_k \Delta t_c$ cancels the part of the correction that comes from the proportional term when $\tau_D = 0$, and Equation (2.18) is what remains, the sampled counterpart of Equation (1.18).

$$u_b + I_k - u_{\mathrm{lim}} = \left(1 - \frac{\Delta t_c}{\tau_I}\right)\left(u_b + I_{k-1} - u_{\mathrm{lim}}\right) \tag{2.18}$$

The integral contribution therefore relaxes to the stop itself with time constant τ_I, removing $1/120$ of the excess each scan at the default $\tau_I = 12\,\mathrm{s}$, and the demanded output settles at $u_{\mathrm{lim}} + K_c e_k$. The pedal leaves the stop only when the error changes sign, which is when the speed crosses the setpoint, and the integral then has to shed the difference between the stop and the pedal the new operating point needs. Windup is bounded at the stop rather than removed, and the application offers no switch that turns the correction off. Measurements on a pedal held at 100 % for 290 s are reported in Section 2.6.

Once the loop has settled, the integral term is the only place the operating point lives, which is why a setpoint change leaves the accumulated disturbance rejection in place. The new setpoint reaches the recursion as a step in the error, the proportional increment moves the pedal by K_c times that step at the next execution, and the derivative increment, which acts on the measurement, adds nothing. Both behaviors are measured in Section 2.6.

2.5.4 Model Predictive Control

A linear model predictive controller (MPC) can drive the pedal in automatic mode instead of the PID, which stays the default in every case except case 9. It follows the general arrangement of Section 1.6.3, and this subsection gives the model, the estimator and the settings it carries for the vehicle. At each decision, $T_s = 0.5\,\mathrm{s}$ or five controller executions apart, the MPC computes a pedal trajectory over a horizon and applies only its first move, keeping it to the next decision [14]. The prediction model is Equation (2.19), fitted with the speedometer noise off to the averaged responses to pedal steps of $\pm 3\,\%$ about the design point of the shipped cases, $\bar{v}_m = 25\,\mathrm{m/s}$ on the flat with the default load and $\bar{u}_m = 15.31\,\%$.

$$\tau_m \frac{\mathrm{d}\xi}{\mathrm{d}t} = -\xi + K_m\,(u - \bar{u}_m + \delta + p) \tag{2.19}$$

Here ξ is the speed deviation of the model in m/s, whose predicted speed is $v_m = \bar{v}_m + \xi$, $K_m = 0.820\,\mathrm{m/(s\,\%)}$ and $\tau_m = 19.23\,\mathrm{s}$ are the fitted gain and time constant, and δ and p are an unmeasured disturbance and the grade read from a map, both in percent of pedal. The fit finds no dead time and a root-mean-square residual of 0.0011 m/s, and its pair lies 0.5 and 1.0 % above the tangent values of Equation (2.6).

The estimate $\hat{\delta}$ is supplied by a steady-state Kalman filter built on Equation (2.19), sampled with a zero-order hold at Δt_c and extended with δ as a random walk, the input-disturbance arrangement that removes offset from a predictive controller [15]. At every execution, whichever controller holds the pedal, the filter runs the two steps of Equation (2.20).

$$\begin{aligned} \bar{\xi}_k &= \phi\,\hat{\xi}_{k-1} + \gamma\left(u_{k-1} - \bar{u}_m + \hat{\delta}_{k-1} + p_{k-1}\right) \\ \begin{bmatrix}\hat{\xi}_k \\ \hat{\delta}_k\end{bmatrix} &= \begin{bmatrix}\bar{\xi}_k \\ \hat{\delta}_{k-1}\end{bmatrix} + \begin{bmatrix}\ell_\xi \\ \ell_\delta\end{bmatrix}\left(\hat{v}_k - \bar{v}_m - \bar{\xi}_k\right) \end{aligned} \tag{2.20}$$

In Equation (2.20) $\phi = e^{-\Delta t_c/\tau_m} = 0.99481$ and $\gamma = K_m(1-\phi) = 0.004\,254\,\mathrm{m/(s\,\%)}$, u_{k-1} and p_{k-1} are the pedal and the map over the period just ended, $\hat{v}_k$ is the speedometer reading shared with the PID, and $\ell_\xi = 0.162$ and $\ell_\delta = 3.53\,\%\,\mathrm{s/m}$ are the steady-state gains obtained with a measurement variance $r = \sigma_n^2$, a state noise variance of $10^{-6}r$ and a disturbance noise variance $q_\delta = 10\,r/K_m^2 = 0.0952\,\%^2$. An unmapped grade, headwind or cargo appears in $\hat{\delta}$ as an equivalent pedal deficit, which every plan carries to the end of its horizon, and a steady load therefore leaves no offset.

With the map on, p is the pedal that the grade of the coming road is worth at the design mass and thrust, as shown in Equation (2.21).

$$p(t) = -\frac{\bar{M}g}{\bar{F}_p}\,\sin\!\left(\arctan\frac{s_{\mathrm{map}}(t)}{100}\right) \tag{2.21}$$

Here $\bar{M} = 700\,\mathrm{kg}$, $\bar{F}_p = 30\,\mathrm{N/\%}$ and $s_{\mathrm{map}}(t)$ is the grade the case schedules for time t, never a grade the operator sets during a run, so a 5 % grade gives $p = -11.43\,\%$. The filter applies p as a known input, which leaves $\hat{\delta}$ only what the map omits, and each plan reads p over its whole horizon, as look-ahead controllers for heavy trucks read the slope of the coming road from map data [16]. The map is exact and is converted at the mass of the default vehicle, so case 9 is a teaching bound on what a preview can deliver.

At each decision a quadratic program is solved on Equation (2.19) sampled at T_s, over a horizon $T_p = 80\,\mathrm{s}$ of $N = 160$ steps, in which twelve pedal positions μ_j, held over move blocks of $(1,1,1,1,2,2,3,4,6,9,15,115)$ steps, minimize Equation (2.22) subject to the hard limits $-50 \le \mu_j \le 100$.

$$J = \frac{T_s}{\tau_n}\sum_{k=1}^{N}\left(\frac{v_{m,k} - v_{\mathrm{SP}}}{y_s}\right)^2 + \kappa\sum_{j=1}^{12}\left(\frac{\mu_j - \mu_{j-1}}{u_s}\right)^2 \tag{2.22}$$

Here $v_{m,k}$ is the speed predicted k steps ahead, μ_0 the pedal now, $\tau_n = 19.2\,\mathrm{s}$, $y_s = 1\,\mathrm{m/s}$ the settling band of the scorecard, $u_s = 150\,\%$ the pedal span and $\kappa = 0.3$ a dimensionless move suppression, named λ in the application and renamed here because λ is the complex frequency of Equation (2.8). No speed band, rate limit or terminal weight enters. The decisions are departures from $\bar{u}_m$, the Hessian carries a ridge of 10^{-9} relative to the two weights, and a dual active-set method [17] returns the optimum. One rule, common to the seven single-loop studio apps, set κ and q_δ before any comparison, as the largest κ within 10 % of the best true-value IAE with no more pedal travel than the PID on cases 3 and 4.

The plan is discarded if the solver finds no optimum or if the move it would apply is not a finite number inside the pedal travel. The PID is the fallback of Section 1.6.3. It follows the applied pedal at every execution, so it takes the pedal without a bump until a plan is accepted again. Forced at

20.5 s of case 9 with the noise off, while the plans are adding pedal for the grade, a rejection moves the pedal by 0.092 %, the size of each of the next three PID moves, and the plan that returns at 21.1 s moves it by 3.8 %. The MPC reads the speedometer, the pedal it applied, the setpoint, the pedal limits and, with the map on, the grade schedule of the case, and nothing of the plant, so the vehicle of Equation (2.1), the instrument of Equation (2.4) and the model of Equation (2.19) stay distinct.

2.5.5 Moving Horizon Estimation

The state from which each plan starts has a second possible source. Under the MPC, a moving horizon estimator (MHE) can replace the filter of Equation (2.20) as that source, fitting the same sampled model to the latest readings while holding the states inside declared bounds [18]. The filter itself runs on in either case. With the augmented state $x_j = [\xi_j \;\; \delta_j]^{\mathsf{T}}$ and the window $j = s, \dots, k$, $s = k - N_e$, every execution solves Equation (2.23) for x_s and the process noise $\nu_s, \dots, \nu_{k-1}$.

$$\begin{aligned} \min \; & \|x_s - \bar{x}_s\|^2_{\bar{P}^{-1}} + \sum_{j=s}^{k-1} \|\nu_j\|^2_{Q^{-1}} + \frac{1}{r} \sum_{j=s}^{k} (\hat{v}_j - \bar{v}_m - \xi_j)^2 \\ \text{s.t. } & x_{j+1} = \begin{bmatrix} \phi & \gamma \\ 0 & 1 \end{bmatrix} x_j + \begin{bmatrix} \gamma \\ 0 \end{bmatrix} (u_j - \bar{u}_m + p_j) + \nu_j \\ & \bar{v}_m + \xi_j \geq 0 \end{aligned} \tag{2.23}$$

In Equation (2.23), $Q = \operatorname{diag}(10^{-6} r, q_\delta)$ and r are the variances of the filter, u_j and p_j are the pedal and the map applied from execution j to $j+1$, $\bar{x}_s$ is the estimate the MHE formed at $s-1$ advanced one step through the model, and $\bar{P}$ is the steady a-priori covariance of the filter, held fixed as the arrival cost. With $N_e = 30$ the window covers 3 s and 31 readings, so a solve has 62 decisions and 31 bound rows. A dual active-set method [17] returns each solution. The only bound is the $v \geq 0$ of Equation (2.1), the one physical limit that the declared values place on the speed, while δ stays unbounded because nothing declares its range in pedal units. A window that ends without an optimum, with a non-finite estimate or with one outside the bound is rejected, and its estimate is replaced by the Kalman update of the one-step MHE prediction, clipped to the bound and marked in the log. A vehicle constant changed by the operator reaches the MHE only through the speedometer, exactly as it reaches the filter.

2.5.6 Relay Autotuning

The PID constants can also be measured on the running loop instead of derived from a step test, by the relay feedback experiment of [19], which Section 1.4.7 introduces. The application offers it as an action taken from the present state in automatic, never as a mode in which a case runs. The relay takes the pedal from the PID and holds it $h = 7.5\,\%$, a twentieth of its travel or 0.9 of the distance to the nearer stop if that is less, above or below the pedal averaged over the preceding 6 s, one tenth of the trend window, switching each time the speedometer reading crosses the setpoint by more than $\varepsilon = 3\sigma_n = 0.24\,\mathrm{m/s}$, where σ_n is the declared noise and not the value set for the run. A cycle counts only when each half lasts four executions or more. When two counted cycles agree within 5 % in period and in swing $\pm a$, the describing function of a relay with hysteresis and the Tyreus–Luyben rule of Table 1.2 set the constants of Equation (2.24) and $\tau_D = 0$, where K_{cu} is the ultimate gain in % s/m, D the share of each period during which the pedal sits high and P_u the measured period.

$$K_{cu} = \frac{4h \sin(\pi D)}{\pi \sqrt{a^2 - \varepsilon^2}}, \qquad K_c = \frac{K_{cu}}{3.2}, \qquad \tau_I = 2.2\, P_u \tag{2.24}$$

The first relation is Equation (1.16), in which the relay amplitude and the swing are written δ and α, with the factor $\sin(\pi D)$ added for a cycle whose two halves are unequal. The experiment gives up after 30 cycles or 120 s. The PID tracks the applied pedal throughout and resumes from it with

whatever constants the experiment left. Where the swing is small against the band, the band and not the loop fixes the cycle, and the application flags the result once $\psi = \arcsin(\varepsilon/a)$, the angle separating the observed cycle from the -180° point, reaches 10°.

> ⧖ HISTORY **Tuning by Experiment, 1942 to 1984**
>
> The first tuning rules for PID controllers were presented by Ziegler and Nichols in 1942, after extensive experiments with pneumatic simulators and a differential analyzer. One method reads a bump test of the open loop, and the other raises the proportional gain until the loop starts to oscillate. The rules had a huge impact, and controller makers adopted them for routine use [6]. In 1984 it was still common experience that many industrial regulators were poorly tuned, partly because simple and robust tuning methods had not been available. A relay placed in the loop then generated its own test signal and found the critical gain and frequency, a method simple to run on a microprocessor [19]. Relay autotuning, started by pushing a button, is now used in many commercial PID controllers [6].

2.6 Results

The application is used as Section 1.6.4 describes for every framework app. Manual mode positions the pedal directly, automatic mode closes the loop of Section 2.5, and each run exports as a comma-separated file with the measured and the true road speed, the pedal, the setpoint and the three disturbances at ten rows per second. Ten cases ship with the application, each starting at steady operation, and Table 2.4 lists them in selector order with the requirement for completing each. Every experiment below, up to Section 2.6.6, starts from the first of them, in which the vehicle cruises at 25 m/s in manual with the pedal at 15.31 %, and is set up from the operator inputs of Tables 2.1 and 2.2 before the run starts, with every setting on a step the application accepts. A setting chosen before Run moves the starting steady state rather than acting as a step, so each step or upset below is applied after the run has started, and the application gives the same run whether a setting is entered before Run or carried through Reset. The scorecard reports the integrated absolute error (IAE), overshoot, settling time, pedal travel and share of time on a limit, but it integrates the whole run, so every number quoted here is computed from the exported record over the window named, or at the 0.05 s simulation step where stated. The sequence appears in Table 2.5. Runs used for identification and for the deterministic offsets set the speedometer noise to zero, which the application allows.

2.6.1 Manual Operation and the Steady-State Map

Equal increments of the pedal produce speed increments that shrink as the vehicle goes faster. With the pedal set to zero before Run, so that the first case starts at rest, manual settings of 5, 10 and 15 % applied in turn from $t = 1$ s, each held for 150 s, reach 13.97, 20.19 and 24.74 m/s against the 14.29, 20.20 and 24.74 m/s of Equation (2.3), so the second 5 % increment buys 4.55 m/s against 6.22 m/s for the first. The first plateau is 0.31 m/s short of its resting value because the approach from rest carries the longest time constant of the three, 33.3 s against 23.6 and 19.2 s. The same three plateaus show the time constant moving, because each approach is quicker than the one before it although the pedal steps that produced them are the same size. Figure 2.2 shows the approach from rest at 15 % of pedal. The vehicle reaches 24.64 m/s 119 s after the pedal is applied.

The two disturbances that a step test cannot distinguish are separated by holding the pedal still. With the pedal set before Run to 15 %, the whole percent nearest the 15.31 % of Equation (2.3), the vehicle starts and rests at 24.7436 m/s, a 5 m/s headwind takes it to 19.7436 m/s, a 5 m/s tailwind takes it to 29.7436 m/s, and removing the wind returns it to 24.7436 m/s. The unity disturbance gain of Equation (2.6) is reproduced to four decimal places in both directions. Adding 400 kg of cargo to the same standing experiment leaves the resting speed at 24.7436 m/s, because the mass appears nowhere in (2.3).

Table 2.4 Scenarios and Completion Requirements

	Scenario	Kind	Required to complete
1	Reference operating point	Experiment	In manual at the steady cruise of 25 m/s with the pedal at 15.31 %, run for 15 s and confirm that nothing moves.
2	Open-loop actuator step	Experiment	From the same cruise in manual, the pedal steps from 15.31 to 35.31 % at 5 s. Run through the step to 35 s, by which time the speed is 36.56 m/s and heading for 37.96 m/s.
3	Closed-loop setpoint step	Experiment	In automatic at 25 m/s, the setpoint steps to 29.5 m/s at 5 s. Run until the speed has stayed within $\pm 1\,\mathrm{m/s}$ of the new setpoint for 3 s.
4	Disturbance rejection	Experiment	In automatic at 25 m/s, the road grade steps from 0 to 5 % at 5 s. Run through the step to 35 s.
5	Recover by hand	Exploration	In manual at 25 m/s with the same grade step at 5 s, set the pedal by hand after the step to hold the speed, once settled, within half of its largest error since the step, or within $\pm 1\,\mathrm{m/s}$, for 3 s in manual.
6	Conservative tuning	Experiment	In automatic at 25 m/s with $K_c = 2.4\,\%\,\mathrm{s/m}$ and $\tau_I = 30\,\mathrm{s}$, the setpoint steps to 28.6 m/s at 5 s. Run through the step to 35 s.
7	Aggressive tuning	Experiment	As case 6 with $K_c = 12\,\%\,\mathrm{s/m}$ and $\tau_I = 6\,\mathrm{s}$.
8	Actuator limit	Demonstration	In automatic at 25 m/s with the pedal thrust lowered to 10 N/% (pedal 45.94 %), the setpoint steps to 40 m/s at 5 s. Run until the pedal reaches its limit and for at least 20 s.
9	MPC with the hill on the map	Experiment	In automatic at 25 m/s under the predictive controller with the map on, a 5 % grade begins at 20 s. Run 60 s from $t = 0$ under the MPC, then reset and run the same 60 s under the PID.
10	Return to target	Challenge	Starting from $K_c = 1.2\,\%\,\mathrm{s/m}$ with the road grade stepped from 0 to 5 % at 5 s, retune so that the measured speed stays within $\pm 2\,\mathrm{m/s}$ of 25 m/s over the last third of a 30 s run (19.8 to 30 s), with the model parameters and disturbances as given.†

†With the shipped K_c the vehicle is still at 21.77 m/s when judging starts and the worst measured error is 3.52 m/s (3.26 m/s in the true speed), so the case fails. The suggested $K_c = 6\,\%\,\mathrm{s/m}$ and $\tau_I = 12\,\mathrm{s}$ pass with a worst measured error of 0.918 m/s (0.806 m/s true), the same whether revealed before Run or carried through Reset. Raising K_c alone also passes, from 2.76 % s/m with the application's own noise sequence. Model runs with four other noise seeds give 3.35 to 3.43 m/s shipped and 0.84 to 0.93 m/s suggested.

Table 2.5 Experiment Sequence and the Mechanism Each One Exposes

Experiment	Setup	What it exposes
Pedal ladder	manual 5, 10, 15 % from rest	equal pedal steps buy less speed each time
Step test	2 % step from 2, 15 and 30 %	$\hat{K}_u$ and $\hat{\tau}$ both scale as $1/\bar{v}$
Cargo at fixed pedal	m_L 200 to 600 kg in manual	no steady-state effect at all
Wind at fixed pedal	w stepped to $\pm 5\,\mathrm{m/s}$	$K_w = -1$ exactly, at any speed
Hill, PI	s 0 to 4 %, $\tau_I = 12\,\mathrm{s}$	invariant $\int e\,dt$ and moving damping
Hill, P only	same upset, τ_I at its 120 s top (integral off)	offset of Equation (2.11)
Pedal on the stop	v_{SP} 25 to 40 m/s, 8 % grade, 15 m/s wind, 600 kg	windup bounded at the stop
Setpoint change on a hill	v_{SP} 25 to 25.5 m/s	integral history survives the change
Setpoint step on the flat	v_{SP} 25 to 28 m/s, τ_D 0 and 0.8 s	no derivative kick
Noise and derivative	τ_D 0 and 0.2 s at 25 m/s	pedal activity against speed variation

2.6.2 Grade Rejection at Three Cruise Speeds and Two Masses

Table 2.6 reports the identical load disturbance, a road grade opened from 0 to 4 %, applied to the same controller at three cruise speeds and at two masses. Each run chooses automatic, the setpoint at the cruise speed, the cargo and the default tuning before Run, so it starts at the steady state of that setpoint, and the grade is applied in one step 10 s into the run. The speedometer noise is turned off for these runs, so the offsets quoted are the deterministic ones.

Table 2.6 Response to a Road Grade Step From 0 to 4 % With $K_c = 6\,\%\,\mathrm{s/m}$, $\tau_D = 0$ and $\sigma_n = 0$, Measured Over the 300 s Following the Upset. Deviations and Times Are Read From the True Speed, IAE From the Measured Speed.

	$m_L = 200\,\mathrm{kg}$			$m_L = 600\,\mathrm{kg}$
$\bar{v}$ (m/s)	10	25	35	25
disturbance force d (N)	274.4	274.4	274.4	431.1
ω_n (rad/s)	0.1464	0.1464	0.1464	0.1168
ζ from Eq. (2.9)	0.950	1.058	1.129	0.844
Integral action, $\tau_I = 12\,\mathrm{s}$ ($K_I = 0.5\,\%/\mathrm{m}$)				
peak deviation (m/s)	1.029	0.957	0.914	1.389
predicted	1.019	0.949	0.907	1.378
time of peak (s)	6.9	6.7	6.5	9.0
predicted	6.9	6.7	6.6	9.0
IAE (m)	18.29	18.29	18.29	29.19
from Eq. (2.10)	18.29	18.29	18.29	28.74
pedal increment (%)	9.146	9.146	9.146	14.372
return within 0.05 m/s (s)	36.6	41.3	44.2	41.5
Proportional only, integral switched off at the upset				
final offset (m/s)	1.417	1.271	1.190	—
from Eq. (2.11)	1.409	1.266	1.185	—
pedal travel (%)	8.5	7.6	7.1	—

The linearized description of Section 2.4 accounts for every entry in Table 2.6 to within 1 %. The peak deviation under integral action falls from 1.029 m/s at 10 m/s to 0.914 m/s at 35 m/s, an 11 % improvement produced entirely by the rise in damping ratio from 0.950 to 1.129, and the predicted peaks of 1.019 and 0.907 m/s sit 0.8 to 1.0 % below the measurements. That remainder is the 0.1 s sample period, because a model run of the same loop executed every 0.01 s comes within 0.3 % of the predictions. The time of the peak barely moves, from 6.9 to 6.5 s, which is the invariant natural frequency of (2.9) showing itself directly. The integrated error is 18.29 m in all three of the 200 kg runs, matching Equation (2.10) to the digits reported, and the steady pedal increment is 9.146 % in all three, matching d/F_p. Figure 2.3 draws the measured rows of Table 2.6 as bars.

The proportional-only rows run the other way from the intuition that a falling process gain gives a worse loop. In these runs τ_I is set to the 120 s top of its range at the moment of the upset, which switches integral action off and freezes the integral at its settled value as the bias of a proportional law. The offset shrinks as the vehicle goes faster, from 1.417 m/s at 10 m/s to 1.190 m/s at 35 m/s, because the drag term $2k\bar{v}_r$ in the denominator of Equation (2.11) is itself a proportional feedback supplied by the vehicle itself and grows with speed. At 10 m/s that term contributes 14.7 N s/m against the 180 N s/m of the controller, and at 35 m/s it contributes 51.45 N s/m. The controller is doing 92 % of the work in the first case and 78 % in the second.

Cargo is the disturbance that changes the loop rather than the setpoint. The same grade step applied to the same controller with 600 kg aboard produces a peak deviation of 1.389 m/s against 0.957 m/s with 200 kg, because the added mass drops ω_n from 0.1464 to 0.1168/s and ζ from 1.058 to 0.844 while also raising the disturbance force from 274.4 to 431.1 N. The measured IAE of 29.19 m exceeds the 28.74 m of Equation (2.10) by 1.6 % for a reason worth stating, because $\zeta < 1$ in this run alone and the speed therefore overshoots the setpoint on the way back, so the absolute error integrated by the metric exceeds the signed error fixed by (2.10). Integrating the absolute value of the linearized response at $\zeta = 0.844$ gives 29.16 m against the signed 28.74 m, which accounts for the difference to within 0.1 %.

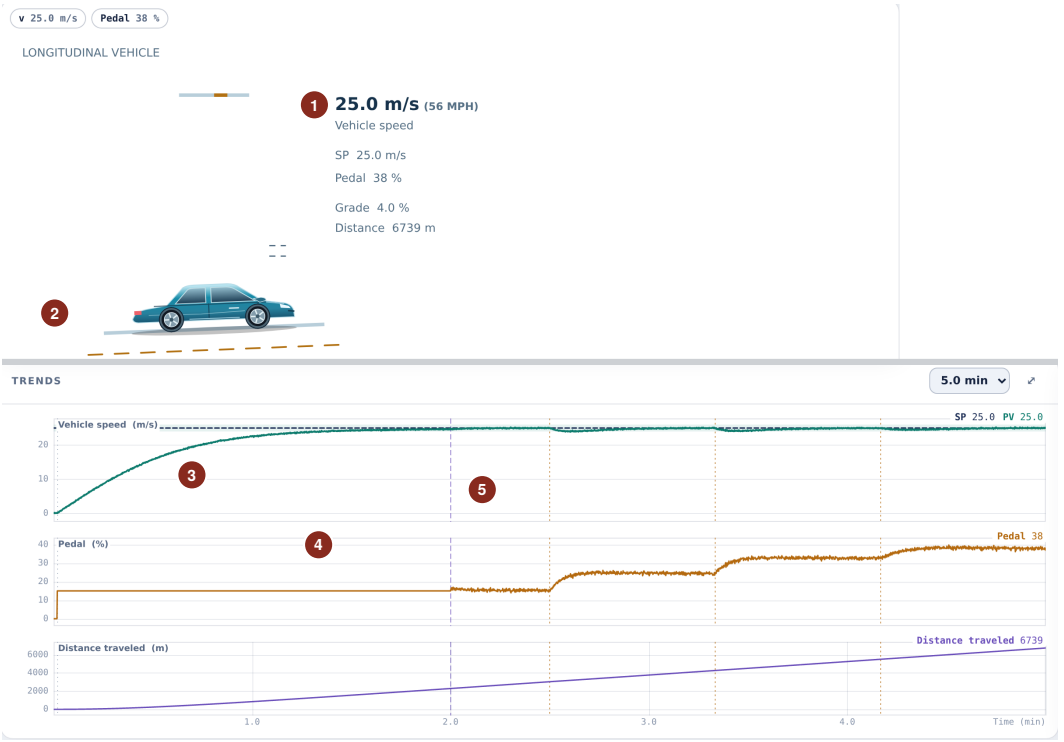


Figure 2.2 Start-up, transfer to automatic and three load disturbances in the first case with the default speedometer noise of 0.08 m/s, captured at $t = 300$ s and cropped to the vehicle view (top) and the trends (bottom). The pedal is set to zero before Run, so the case starts at rest, and to 15 % at $t = 1$ s. The vehicle reaches 24.64 m/s by $t = 120$ s, where the loop is switched to automatic with $v_{SP} = 25\,\mathrm{m/s}$, $K_c = 6\,\%\,\mathrm{s/m}$ and $\tau_I = 12$ s, and the pedal continues from the 15 % the driver held. A 4 % grade at $t = 150$ s, a 6 m/s headwind at $t = 200$ s and 400 kg of extra cargo at $t = 250$ s pull the true speed down to 24.03, 24.14 and 24.50 m/s, and after each the speed returns to setpoint while the pedal settles about the 24.46, 32.69 and 37.92 % the force balance requires. At $t = 300$ s the measured speed is 24.98 m/s and the pedal 37.87 %. (1) The speed readout with the setpoint, the pedal, the grade and the distance. (2) The animated vehicle on the grade, driven by the true speed. (3) The speed trend with the setpoint dashed, where the speedometer noise remains visible throughout. (4) The pedal trend. (5) The event markers, the switch to automatic dashed and the three upsets dotted.

> ✦ **DID YOU KNOW?** **Faster Car, Smaller Offset**
>
> From the simulator: with integral action switched off, the same 4 % grade leaves an offset of 1.417 m/s at 10 m/s but only 1.190 m/s at 35 m/s, although the process gain is lower at the higher speed. The drag itself supplies part of the restoring action (Table 2.6).

2.6.3 Actuator Saturation

A cruise setpoint of 40 m/s with 600 kg of cargo, an 8 % grade and a 15 m/s headwind demands more force than the powertrain can supply. The run starts in automatic at 25 m/s under that load, with the pedal at the 67.875 % that holds it, and the setpoint is raised to 40 m/s 10 s into the run. The pedal reaches its 100 % stop at that execution and stays there for the 290 s before the wind is removed at $t = 300$ s, and the vehicle settles at 38.956 m/s, which is what solving $F_p \cdot 100 = k(v+w)^2 + Mg\sin\theta$

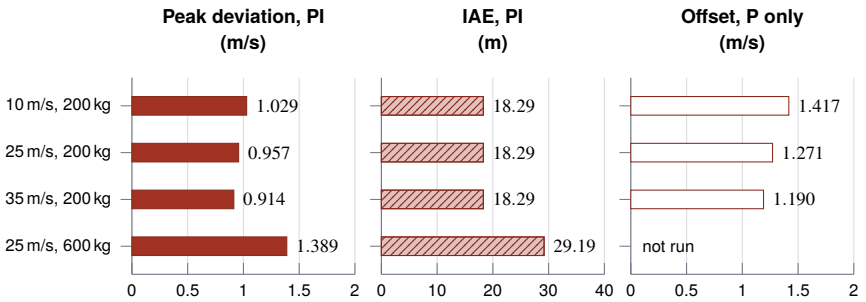


Figure 2.3 The grade step of Table 2.6 at three cruise speeds and two loads under one set of gains: the peak deviation (solid bars) and the integrated absolute error (hatched bars) under integral action, and the final offset with integral action switched off (open bars). Notice that the integrated error does not move with the cruise speed, that the peak and the offset both shrink as the car goes faster, and that cargo, not speed, is what makes the loop worse.

for the road speed returns. The controller is not at fault and the offset of 1.044 m/s is not a tuning failure, because the setpoint lies outside the set of speeds this road load allows. Meanwhile $u_b + I$ relaxes through Equation (2.18) from the starting 67.875 % to the stop itself, within 0.001 % of it 124 s after the setpoint change and at 100.000 % when the wind is removed.

Removing the headwind at $t = 300$ s does not release the pedal at once. It leaves the stop at the scan 1.2 s later, the first at which the measured speed exceeds 40 m/s, and is at 91.4 % at the scan 3 s after the removal and at 66.58 % at the scan 28 s after it. The speed overshoots to 42.896 m/s, which is 7.2 % above the setpoint, 9.9 s after the removal, comes back within 0.05 m/s of the setpoint after 54.1 s, and rests at 40.000 m/s with the pedal back at 67.875 %, because the air speed is 40 m/s in both steady states. The overshoot is the vehicle's own inertia carrying it through the 1024 N of excess thrust that existed at the instant the wind stopped, extended by the 1.2 s on the stop and by the 32 % of pedal the integral has to shed once the speed has crossed the setpoint. A model run of the same controller with the back-calculation removed, which the application does not offer, winds $u_b + I$ up to 323.2 % during the 290 s on the stop, holds the pedal down for 36.0 s after the wind is removed and carries the vehicle to 52.69 m/s. The difference between the two behaviors is the content of the last line of Equation (2.16).

2.6.4 Setpoint Changes

Raising the setpoint from 25.0 to 25.5 m/s at $t = 300$ s, while the loop is holding 25 m/s against a 4 % grade that arrived 10 s into the run, moves the pedal from 24.458 to 27.458 % on the next control execution, a step of 3.000 % which is the proportional gain acting on the 0.5 m/s change in error. The speed never falls below the old setpoint, peaks at 25.521 m/s, comes within 0.05 m/s of the new setpoint after 7.2 s, and rests at the 25.077 % of pedal that the force balance on the grade requires at the new speed. The 9.146 % of pedal accumulated against the hill survives the setpoint change, because the velocity form applies the new setpoint to the next increment and leaves the integral where it was.

On the flat, a setpoint step from 25 to 28 m/s 10 s into the run with the default tuning moves the pedal by $K_c \Delta e = 18$ % at the next execution, from 15.31 to 33.31 %, which is also the highest the pedal goes. The speed overshoots to 28.107 m/s and stays within 0.05 m/s of the new setpoint from 30.3 s after the change. Adding derivative action with $\tau_D = 0.8$ s ($K_D = K_c \tau_D = 4.8\,\%\,\mathrm{s}^2/\mathrm{m}$) leaves that first increment unchanged, because the derivative acts on the measurement and the setpoint never enters it, so the application has no derivative kick to show. The derivative instead opposes the acceleration the proportional step started, the speed overshoots further to 28.170 m/s, the approach takes 37.1 s, and the pedal travel over the 200 s following the change rises from 32.2 to 33.5 %.

2.6.5 Measurement Noise and Derivative Action

With the default 0.08 m/s of speedometer noise and the default tuning, a settled loop at 25 m/s holds the pedal to a standard deviation of 0.482 % against the $K_c\sigma_n = 0.480\,\%$ that the proportional term alone contributes, and the true speed varies by only 0.010 m/s because a 19 s time constant is an effective low-pass filter on the pedal activity. A derivative time of 0.2 s ($K_D = 1.2\,\%\,\mathrm{s}^2/\mathrm{m}$) nearly quadruples the pedal standard deviation to 1.793 % and raises the pedal travel from 5.5 to 24.7 % per second while the true speed variation rises to 0.011 m/s rather than falling. Each figure is taken over 600 s of a settled run with the application's own noise sequence. Model runs with four other noise seeds, which the application does not expose, move the pedal statistics by up to 3 % and the true speed variation by up to 13 %, and the rise in true speed variation with derivative action appears in every one of them. The short sample period is what makes derivative action expensive on this loop, because $K_D/\Delta t_c = 12\,\%\,\mathrm{s/m}$ is already twice the default proportional gain. Derivative action has nothing to recover on a first-order plant with no dead time, which is why τ_D defaults to zero.

2.6.6 The Shipped Cases Under the PID

The experiments above are built on case 1, because a settled loop is what the analysis of Section 2.4 describes. The other cases of Table 2.4 add results on the vehicle as shipped. Every case starts at steady operation, cruising at 25 m/s, so each scripted event at 5 s lands on a settled loop. In case 4 the 5 % grade pulls the true speed down by 1.21 m/s, to its lowest at 12.1 s, and the loop has it back within $\pm 1\,\mathrm{m/s}$ by 16.7 s.

> **TRY IT IN THE APP** **4 · Disturbance Rejection**
>
> Load the case and press Run. Watch the speed trend when the road turns uphill at 5 s. The true speed falls by 1.21 m/s, reaches its lowest at 12.1 s, and the loop has it back within 1 m/s of the setpoint by 16.7 s, while the pedal trend climbs to carry the hill.

In case 5 the same grade with the pedal untouched slows the vehicle to 14.70 m/s by 60 s on its way to 12.59 m/s, while a pedal of 27 % holds 25.2 m/s. The completion rule of case 5 credits only a pedal that holds the speed in the band once settled. A pedal of 26 or 27 % set at 5.1 s is credited at 8.1 s. A pedal of 20 %, which would let the speed fall toward 18.7 m/s, is never credited.

Cases 6 and 7 scale both tuning terms. The conservative case rises from 10 to 90 % of its step in 40.15 s with no overshoot, and the aggressive one in 2.9 s with 8.9 % overshoot of the true speed. The default tuning on the same step gives 7.0 s and 3.5 %. The scorecard counts 88 and 455 % of pedal travel over the 35 s run, against 224 % for the default tuning. Case 8 holds the pedal on its stop from the execution at 5 s, and the speed climbs toward the 36.89 m/s that full pedal holds on the weaker engine, 3.11 m/s short of the setpoint. The challenge is judged on the case as given, so a changed model parameter or disturbance voids the attempt.

> **TRY IT IN THE APP** **6 · Conservative Tuning**
>
> Run the case and time the climb to the new setpoint: 10 to 90 % of the step takes 40.15 s, with no overshoot. Then load 7 · Aggressive tuning and run it. The same step takes 2.9 s with 8.9 % overshoot of the true speed, and the scorecard counts 455 % of pedal travel against 88 %.

2.7 The Predictive Controller on the Shipped Cases

Unlike the experiments of Section 2.6, which start from case 1, Table 2.7 runs the shipped cases 3, 4, 8 and 9 over 60 s and the challenge over 30 s, with every value read on the 0.1 s log as the scorecard reads it. Each controller is selected before the run starts, so every run of a case meets the same events and noise. The PID keeps the shipped $K_c = 6\,\%\,\mathrm{s/m}$, $\tau_I = 12\,\mathrm{s}$ and $\tau_D = 0$, except that the challenge ships $K_c = 1.2\,\%\,\mathrm{s/m}$ and suggests $K_c = 6$. The MPC keeps the settings of Section 2.5.4, with the

Table 2.7 Five Cases Under the Shipped PID and Under the Predictive Controller, With the Same Speedometer Noise in Every Run of a Case

Case	Controller	IAE (m)	IAE_v	e_{max} (m/s)	OS	$t_{0.3}$ (s)	Travel (%)	Limit (%)
3, setpoint	PID	18.76	17.23	—	0.145	8.3	363.3	0
	MPC	7.10	4.48	—	0.172	1.5	286.6	0.8
4, grade	PID	23.61	22.67	1.205	—	26.1	334.6	0
	MPC	4.82	2.22	0.389	—	2.0	138.5	0
8, limit	PID	335.66	335.33	—	—	—	84.0	91.7
	MPC	335.85	335.54	—	—	—	62.2	91.7
9, mapped hill	PID	23.59	22.28	1.192	—	25.7	335.3	0
	MPC, no map	4.87	2.45	0.423	—	2.3	139.3	0
	MPC	4.25	1.66	0.111	—	0	131.4	0
10, challenge	PID, $K_c = 1.2$	64.69	64.42	3.256	—	—	35.8	0
	PID, $K_c = 6^a$	19.94	19.69	1.205	—	—	176.3	0
	MPC	2.76	1.40	0.389	—	2.0	83.0	0

IAE uses the speedometer reading, as the scorecard does, and IAE_v the true speed. e_{max} is the largest true deviation after the grade begins, OS the largest true overshoot of a new setpoint, $t_{0.3}$ the time from the first event until the true speed stays within 0.3 m/s of setpoint (a dash if it never does within the run), Travel the summed absolute pedal change and Limit the time on a pedal stop as a share of the run. [a]Suggested, with $\tau_I = 12$ s.

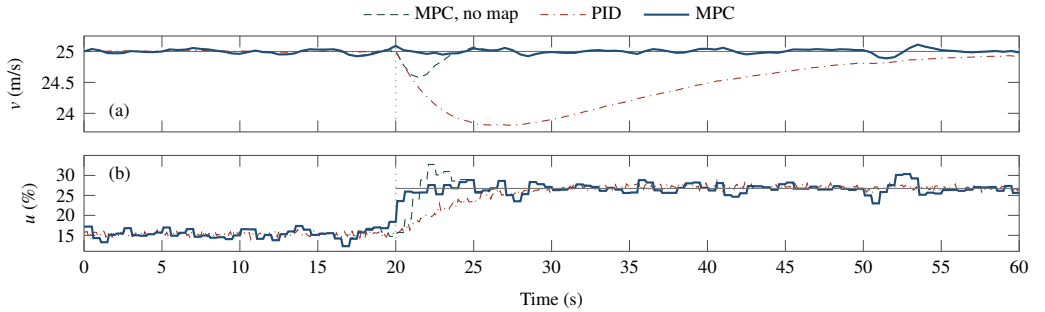


Figure 2.4 Case 9 under the predictive controller with the map (thick solid curve) and without it (dashed curve) and under the shipped PID (dash-dotted curve), with identical speedometer noise, from the 0.1 s log. A 5 % grade begins at 20 s (dotted vertical line). (a) True speed, with the setpoint as a thin gray line. (b) Pedal position, with the 26.74 % that holds the setpoint on the grade as a thin gray line.

map on only in case 9, which is also run without it. The PID has no feedforward, so in case 9 the MPC without the map is the like-for-like comparison.

In case 3, a setpoint step taken where the model was identified, the first plan puts the pedal on its 100 % stop for one decision period, where the proportional kick of the PID reaches 42.6 %. The true speed then covers 10 to 90 % of the step in 1.2 s against 7.0 s and overshoots by 0.172 against 0.145 m/s, and the pedal travels 21 % less.

Case 4 gives the MPC no map, so the grade reaches it only as lost speed. The estimate $\hat{\delta}$ books half of the 11.43 % pedal deficit within 1.2 s and 90 % within 2.2 s, and the true speed bottoms at 24.61 m/s 1.5 s after the grade, where the PID reaches 23.80 m/s after 7.1 s. The true-speed IAE of the PID, 22.67 m, is near the 22.85 m that Equation (2.10) fixes for this 342.8 N load whatever K_c is, while the estimate, integral action routed through the model, escapes that relation with 2.22 m and 41 % of the travel.

Case 8 is a tie. With the thrust at 10 N/% the setpoint lies above the 36.89 m/s that full pedal can hold, so both controllers keep the pedal on its stop from 5 s and reach 36.686 m/s at 60 s, the MPC with a scorecard IAE 0.06 % higher.

Case 9 cruises on the flat until a 5 % grade begins at 20 s, as shown in Figure 2.4. With the map, the pedal stays more than 0.5 % above its quiet level from 17.6 s and reaches 18.37 % at 19.9 s, so the

car meets the grade 0.09 m/s above its setpoint, and one second later it is at 25.95 % of the 26.74 % that the grade needs. The true speed then stays between 24.89 and 25.11 m/s. Without the map the MPC bottoms at 24.58 m/s and the PID at 23.81 m/s. Relying on its estimate alone, the MPC thus cuts the true-speed IAE from 22.28 to 2.45 m, and the map removes another 0.79 m, after which the true error averages 0.027 and 0.028 m/s before and after the grade. That remainder is the filter's response to the speedometer, each reading error of σ_n shifting $\hat{\delta}$ by $\ell_\delta \sigma_n = 0.28\,\%$, while the PID, whose white pedal noise the lag smooths, averages 0.0067 m/s before the grade. Of the scorecard IAE of 4.25 m, 3.92 m is the speedometer noise itself.

A static feedforward of $-p$ on the shipped PID, run outside the application, which offers none, cancels the grade exactly on the default vehicle, because pedal and grade enter Equation (2.1) alike, and reaches a true-speed IAE of 0.37 m. The MPC spreads the same pedal around the grade because κ prices each move. With 500 kg of cargo, when the map supplies only 70 % of the pedal the grade needs, the order reverses, 9.79 m for the feedforward PID against 1.83 m for the MPC with the map, whose estimate makes up the rest. The map then misses 30 % of the pedal, yet the lowest speed under the MPC with the map falls only from 24.89 to 24.84 m/s.

In the challenge the MPC reproduces its first 30 s of case 4 exactly, because it does not read K_c, and its largest measured error in the judged final third is 0.303 m/s, against 3.52 m/s for the shipped K_c, which fails, and 0.918 m/s for the suggested tuning. The MPC computed 660 plans and rejected none. A bound was active only in the first plan after the step of case 3 and in 110 plans of case 8, so cases 4, 9 and 10 were won by prediction and estimation alone. The mean solve time, a wall-clock value from one desktop computer, was 0.08 to 0.31 ms under Node.js and 311 and 325 µs in Chrome over two runs of case 9, never above 0.065 % of one core in real time. The MPC wins cases 3, 4, 9 and 10 and ties case 8.

2.8 Estimator Choice and Relay Tuning

The runs of this section last 180 s, the challenge 30 s, with every setting chosen before the run and the shipped noise sequence, so their integrals are longer than those of Table 2.7. With no bound active, Equation (2.23) is the least-squares problem that the filter solves recursively, and the two estimates coincide. The largest gap between them over 1799 executions of case 9 is $4.5 \times 10^{-10} \sigma_n$ in $\hat{\xi}$ and 7.7×10^{-10} % in $\hat{\delta}$. That residue comes from the Riccati iteration that designed the filter, which stops at a relative change of 10^{-11}. Iterated to its fixed point, the covariance brings the difference down to $1.4 \times 10^{-14} \sigma_n$. Each of the ten cases, run under the MPC once with each estimator, gives pedal records that agree within 2.0×10^{-9} %, and no window was rejected. The bound never comes close, since the lowest speed reached in any case is 12.68 m/s, in the manual case 5.

Judged against the true speed, which only the measuring script reads, both estimators place the speed of case 9 with a root-mean-square error of 0.030 m/s and a largest error of 0.114 m/s, inside the 0.08 m/s standard deviation of the speedometer itself. An estimate took a median of 248 µs and a 99th percentile of 501 µs under Node.js over 16,490 estimates. In Chrome, on the frame loop of the application during case 9, the estimates took 0.60 and the plans 0.09 percent of one processor core in real time and 6.02 and 0.47 percent at ten times real time, the larger of two runs taken while other work held the load average of the same computer at about two. The second figure is within a factor of two of the tenth of a core allowed for estimating and planning together.

The relay was pressed 6 s into case 3, run with its setpoint step removed so that the vehicle holds 25 m/s, and swung the pedal between 7.9 and 22.9 %. It converged after three cycles and 6.15 s on $a = 0.369\,\mathrm{m/s}$, $D = 0.46$, $P_u = 2.89\,\mathrm{s}$ and $K_{cu} = 33.9\,\%\,\mathrm{s/m}$, which Equation (2.24) turns into $K_c = 10.6\,\%\,\mathrm{s/m}$ and $\tau_I = 6.36\,\mathrm{s}$ against the shipped 6 and 12 s. Ten further noise seeds all converged, with K_{cu} from 34.1 to 60.5 and P_u from 1.93 to 3.29 s. None of these is the ultimate point of the vehicle. Equation (2.19) sampled with a zero-order hold at Δt_c reaches -180° only at the Nyquist frequency, where its ultimate gain is $(1+\phi)/\gamma = 469\,\%\,\mathrm{s/m}$ at a period of 0.2 s, because a first-order lag with no dead time carries no phase crossover of its own. The relay instead cycles where its

Table 2.8 Shipped and Relay-Tuned PI Against the Predictive Controller on Cases 3 and 4 Over 180 s, With the Same Speedometer Noise in Every Run

Case	Controller	IAE	IAE_v (m)	IAE_v^0	$e_{\max}$ (m/s)	Travel (%)
3, setpoint	PI, shipped	26.48	18.36	17.4	—	1010
	PI, relay	21.32	12.69	11.3	—	1799
	MPC	15.24	7.53	—	—	511
4, grade	PI, shipped	31.35	23.62	22.9	1.205	981
	PI, relay	17.22	8.51	6.86	0.708	1744
	MPC	12.96	5.17	—	0.389	364

Shipped PI $K_c = 6\,\%\,\text{s/m}$, $\tau_I = 12\,\text{s}$, relay PI $K_c = 10.6\,\%\,\text{s/m}$, $\tau_I = 6.36\,\text{s}$. IAE is the scorecard's on the speedometer reading, IAE_v the true-speed IAE on the 0.05 s step, IAE_v^0 the same with $\sigma_n = 0$, and $e_{\max}$ the largest true deviation after the grade. The MPC rows are identical with either estimator.

band puts it, as it does on the blending tank of Chapter 3. On the noise-free model it settles at $K_{cu} = 107\,\%\,\text{s/m}$ and $P_u = 3.2\,\text{s}$, and the application with the noise set to zero returns 103 and 3.20 s. The default swing is only 1.54 times the band, so $\psi = 40.6°$, and over the ten seeds ψ runs from 40.8 to 56.8°, every run past the flag of Section 2.5.6. The noise then lowers K_{cu} by a further factor of three.

The measured constants still make the faster loop on this vehicle, as listed in Table 2.8. They raise K_I from 0.5 to 1.67 %/m and ω_n from 0.146 to 0.267 rad/s, and lower ζ of Equation (2.9) from 1.058 to 0.948. For the 342.8 N grade of case 4, Equation (2.10) then predicts 6.86 m in place of 22.85 m, and the noise-free runs give 6.86 and 22.9 m. The pedal travels 1.78 times as far on both cases, close to the ratio of the proportional gains, 1.77, because the speedometer noise, amplified by the proportional gain, carries 95 to 99 % of the travel in these runs, the noise-free runs traveling 48.1 and 90.5 % on case 3 and 12.4 and 13.7 % on case 4. The MPC with either estimator remains ahead on both cases, with 5.17 m of true-speed IAE on case 4 at a fifth of the travel of the relay-tuned loop.

The relay result also moves with the state from which it is started. Started at 60 s of case 3, once the loop has settled at its new setpoint, the relay returns K_{cu} of 46.0 with the declared noise and 101 % s/m with the noise set to zero, beside the 469 % s/m of the sampled model. The second value calls for a K_c of 31.7, which the top of its range holds at 30. The first gives $K_c = 14.4\,\%\,\text{s/m}$ and $\tau_I = 6.46\,\text{s}$. Run on case 4 for 180 s, that pair lowers the true-speed IAE from 23.6 to 6.98 m at 2.4 times the pedal travel, against noise-free values of 22.85 and 5.13 m from Equation (2.10).

2.9 Technology in Practice

The loop of Section 2.3 reads a speedometer and moves a pedal. Figure 2.5 draws it with real devices in the place of the schematic's symbols.

What Measures It

In a cruise control system the measurement noise comes from the electronics that convert pulses measured on a rotating shaft into a velocity signal [6]. On a road vehicle the pulses commonly come from a variable-reluctance pickup facing a toothed wheel [20], described below. The white noise of 0.08 m/s in Equation (2.4) stands for that conversion, while the bias from tire wear and the quantization by the tooth count that Section 2.3.4 leaves out are what a real pickup adds.

> **TECHNOLOGY** **Sensor: The Wheel-Speed Pickup**
>
> A variable-reluctance pickup sits close to a toothed ferrous wheel. Each passing tooth changes the magnetic flux through its coil, so the output is an alternating voltage whose frequency and amplitude are both proportional to the speed of rotation. The pickup needs no external power and detects iron up to about 2.5 mm away. It cannot sense zero speed, and below about 5 km/h it is very susceptible to noise. Differential Hall-effect sensors resist interference better but need a minimum switching frequency [20].

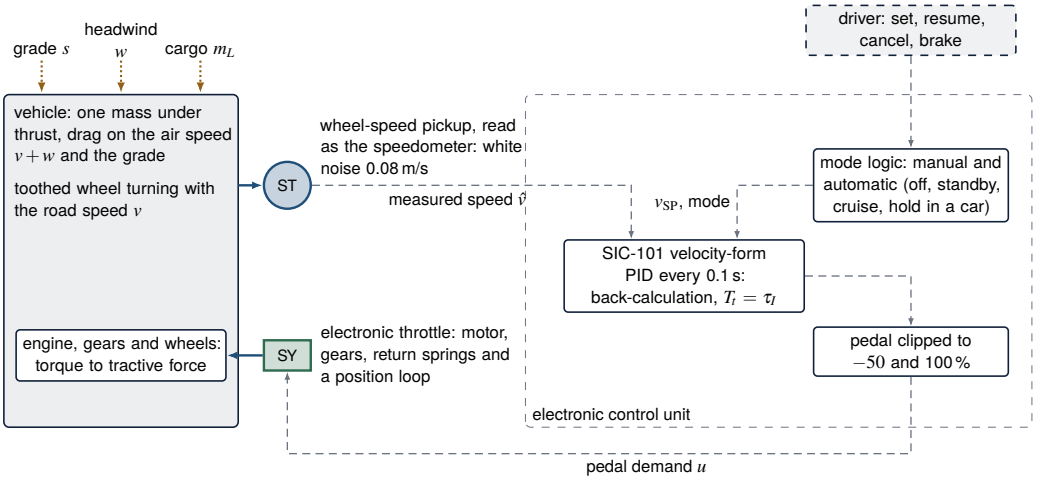


Figure 2.5 The loop of the chapter with real devices in place of the schematic's symbols. A wheel-speed pickup ST reads the road speed, the controller SIC-101 on the vehicle's electronic control unit computes a pedal demand under the driver's mode logic, and the electronic throttle SY turns it into engine torque. Notice that the three disturbances act on the vehicle and none of them is measured, so the controller learns of a hill, a wind or a load only from the speed it loses.

What Moves It

The final element of a cruise loop is the throttle, which sets the engine torque against disturbance forces from the slope of the road, the rolling resistance and the aerodynamic forces. A touch of the brake disconnects the cruise control altogether [6]. The signed pedal of Equation (2.1), clipped at -50 and 100 % with a force linear in its position, stands for throttle, engine and gears together, without their dead time or the throttle's own position loop.

> **TECHNOLOGY** **Final Element: The Electronic Throttle**
>
> The cruise controller sends its output to an actuator that sets the throttle, and the throttle sets the torque the engine delivers through the gears and wheels [6]. An electronic throttle is an electromechanical actuator that sets the angle of the throttle plate, and so the air flow into the engine, with a DC motor, reduction gears, return springs and a throttle position sensor. Its springs and friction are nonlinear and its parameters vary with operating conditions, so the plate angle is held by a position loop of its own. One laboratory throttle closed that loop every 0.01 s on a microcontroller [21].

Where the Controller Runs

Control in a car is executed by electronic control units, microprocessors with input and output interfaces, and a modern car has more than 100 of them. Computer control entered the car in the late 1970s to meet stringent emission requirements [6]. Loops are often nested, with the setpoint of one loop based on the variable of another, and they run with cycle times from milliseconds to minutes [22]. The cruise loop is such an outer loop over the throttle position, which one laboratory throttle closed every 0.01 s [21], and the PID of this chapter executes every 0.1 s. A digital PID reads the sensor, computes and sends the output, and only then updates its states, which keeps the delay short. It usually filters the derivative with a time constant of the derivative time divided by 5 to 20, and it resets its integrator at a tracking gain that is a multiple of the integral gain but not so large that noise resets it [6]. The velocity form of Section 2.5 acts on an unfiltered derivative and tracks with the integral time.

> **IN PRACTICE The Logic Around the PI Loop**
>
> A cruise controller is typically a PI loop inside mode logic. A finite state machine with four modes, off, standby, cruise and hold, answers the driver. Set takes the speed at the moment the button is released as the setpoint. Pressing the gas pedal speeds the car up, and it returns to the set speed when the pedal is released. A touch of the brake puts the loop in hold, which remembers the setpoint, and resume brings it back. Switching modes must not cause a transient, so in manual the controller's output tracks the manual input [6]. The application of this chapter keeps only the manual and automatic modes of that logic.

What Is Modernizing It

Sensing ahead is the first development. Adaptive cruise control, based on radar sensors, maintains a constant distance to the car in front [6], and cooperative adaptive cruise control adds wireless links over which vehicles share their speed, acceleration and position [23]. Preview is the second. An adaptive cruise controller built on model predictive control can read the road grade from a prior map of the road to predict the vehicle's motion and improve safety, comfort, tracking and energy use [24], as the predictive controller of Section 2.5.4 reads the hill of case 9. Software is the third. A partnership of vehicle makers, suppliers and software companies formed in 2003 has created standards for a common automotive software platform, so that functions become modular, transferable and reusable [6].

2.10 Limitations

The application is built for teaching rather than for vehicle design. A point mass with linear thrust cannot represent an engine map, a gearbox, a torque converter or a traction limit, so it must not be used to predict acceleration times, gradeability or fuel consumption. Rolling resistance is absent altogether, which makes every coast-down in the application longer than a real one and removes the resistance that dominates below about 10 m/s. Braking is a speed-independent negative thrust with no wheel dynamics, so nothing in the application can produce lock-up, anti-lock intervention or a stopping distance worth quoting, the subject of Volume 7, Chapter 4. The road has one grade and one wind at a time, both applied everywhere at once, so a real hill that arrives as a ramp is easier to reject than the step tested here. The speedometer carries white Gaussian noise and nothing else, so the tire-radius bias and encoder quantization that dominate a real wheel-speed measurement are not represented. Noise is drawn from a seeded generator that restarts at every case load and Reset, so every run of a case sees the same noise sequence and the spread of a noisy statistic across realizations cannot be seen from inside the application. There is no lead vehicle, no spacing loop and no lateral motion, so the adaptive cruise control and platooning questions that the literature of Section 2.2 is mostly about cannot be posed here at all. Four features were kept deliberately and are not candidates for removal, the quadratic drag law, the signed relative air speed, the single signed pedal with its two stops and the 0.1 s execution rate, because each of them produces something a tuning decision has to answer for. The operating region in which every remaining assumption breaks down is given in Section 2.3.4.

The model of the predictive controller, a first-order pair frozen at 25 m/s on the flat with 700 kg, omits the quadratic drag (its gain is 1.41 times the vehicle's at 35 m/s), the mass and the engine (three times too strong in case 8, which it only ties). Its map is exact, unlike any road database, and a feedforward of the same map does better on the default vehicle. In quiet running it lets the true speed wander four times as far as the PID, and every comparison here uses a single noise realization. The MHE bounds only the states of that same frozen model, and its one bound lies more than 12 m/s below any speed a case reaches, so on this vehicle it adds 0.60 % of a core in real time and nothing else. The relay measures the band and the noise more than the vehicle, because a lag with no dead time has no phase crossover for it to find.

OPEN CHALLENGE **Cruise Control in a String of Cars**

Cooperative adaptive cruise control lets vehicles share their speed, acceleration and position over wireless links, so that they can travel in platoons at shorter gaps. The links bring communication delay, a limited range, packet loss and sampling intervals, and real traffic keeps changing which vehicles talk to which. Many control methods have been proven in theory and in simulation, but few have been tested on real roads, and a system of several vehicles is hard to test over enough conditions and distance before it reaches the market [23]. Progress would be a controller whose spacing stays safe, with errors that do not grow down the string, when messages arrive late or not at all.

2.11 Discussion

The energy a vehicle spends against aerodynamic drag over a distance is the drag force times that distance, so it scales with the square of the speed held and is independent of how long the trip takes. The loop of this chapter therefore has an economic and environmental dimension that most regulatory loops do not. Steadier speed and lower speed both reduce fuel use, and the driving behavior changes grouped under eco-driving have been assessed as one of the cheapest available reductions in transport emissions [25]. A cruise controller that rejects grade and wind with a smaller pedal excursion is doing exactly the thing that assessment recommends, and the pedal travel row of Table 2.6 is the quantity that measures it. The trade is visible in the derivative results of Section 2.6, where raising τ_D from 0 to 0.2 s increases pedal travel from 5.5 to 24.7 % per second and makes the speed regulation slightly worse.

The safety dimension pulls in the same direction and is quantified separately. Systems that hold a vehicle at or below the posted limit, rather than at a driver-chosen setpoint, have been evaluated for their accident savings and their cost, and the estimated benefit is large enough that the intervention is worth mandating on cost grounds alone [26]. The mechanism the application makes visible is the one such a system has to live with, because the achievable set of speeds is bounded by the pedal limit and by the road load, and Section 2.6 shows a vehicle that cannot hold 40 m/s up an 8 % grade into a 15 m/s headwind whatever the controller demands. Adaptive cruise control resolves the conflict between a speed target and a spacing target by switching between two loops, and the speed loop it switches back to is the one of this chapter [8]. Whether the gains of that inner loop should be scheduled on the operating point is a question this application answers with a measurement rather than an assertion [3], and the answer for a plant whose gain and time constant fall together is that the schedule buys less than the gain variation alone suggests. The neutralizer of Chapter 6 is the opposite case. Its gain changes by four orders of magnitude across one valve stroke, and no single set of constants serves it.

OPEN CHALLENGE **Keeping Loops Well Tuned**

A large industrial plant may run hundreds of PID regulators [19]. More than 1730 tuning rules have been collected, yet industrial studies agree that a significant fraction of PID controllers are poorly tuned [6]. A loop tuned once drifts out of tune as its plant moves to another operating point, as the damping of this chapter's loop moves with speed. Progress would be tools that notice when a loop has drifted from its tuning and correct it from normal operating data, without a test that upsets production.

2.12 Conclusion

This chapter studied an automobile cruise control loop in which the load is quadratic in speed. The simulator integrates one state under a longitudinal force balance with thrust linear in pedal position, drag acting on the signed relative air speed, a gravity component set by the road grade, a pedal clipped at -50 and 100 %, and a speedometer that adds 0.08 m/s of Gaussian noise. The linearized description is first order with a gain and a time constant that both shrink in inverse proportion to the cruise speed while their ratio holds at the thrust-to-mass ratio of 0.0429 m/(s^2 %), and pedal steps run in the application follow that scaling to within the width of a finite step, with the identified gain down

by a factor of 3.26 and the identified time constant down by 3.09 between 9.04 and 34.99 m/s against the 3.87 the ratio of starting speeds predicts, the shortfall being the secant of a step that is 41 % of the starting speed at the slow end and 3 % of it at the fast end.

Adding 600 kg of cargo instead of none multiplies the identified time constant by 2.201 against a mass ratio of 2.200 and leaves the identified gain unchanged at 0.799 m/(s %). The three load disturbances separate cleanly, because cargo has no steady-state effect, a headwind removes its own magnitude from the road speed to four decimal places, and a 4 % grade costs a proportional loop, modeled with the integral frozen, 1.417 m/s at 10 m/s and 1.190 m/s at 35 m/s with identical gains. Under integral action the same grade is rejected with an integrated error of 18.29 m at every cruise speed, because that quantity is fixed by the disturbance force and the integral gain alone, while the peak deviation falls from 1.029 to 0.914 m/s as the damping ratio rises from 0.950 to 1.129 at a natural frequency that does not move from 0.146 rad/s.

The default controller is the velocity form of PID with the derivative on the measurement, bumpless transfer in both directions and back-calculation anti-windup with a tracking time equal to the integral time. That choice relaxes the integral to the stop during a saturation, so after 290 s on the stop the pedal stays there until the speed crosses the setpoint and the speed overshoots by 7.2 %, against 32 % when the correction is removed. A linear predictive controller keeps the true speed within 0.11 m/s of setpoint on the grade of case 9 with a map and 0.42 m/s without, against 1.19 m/s for the PID. Its moving horizon estimator reproduces that controller's runs to 2×10^{-9} % of pedal, and a relay test returns $K_{cu} = 33.9\,\%\,\mathrm{s/m}$ where the sampled vehicle has 469, yet its PI constants lower the true-speed IAE of the grade from 23.6 to 8.51 m at 1.8 times the pedal travel.

A fuller model would add rolling resistance, a gear-dependent tractive force, a lead vehicle for spacing control, and a scheduled gain so that the measured damping variation can be removed rather than merely observed. The result to carry into the chapters that follow is that the plant a step test finds depends on how fast the vehicle was going when the test was run.

KEY POINTS

- Equal steps of the pedal buy less and less speed as the car goes faster, because air drag grows with the square of the speed.
- The linearized gain and time constant both shrink in inverse proportion to the cruise speed, at the fixed ratio of thrust to mass.
- Under one PI tuning the natural frequency does not move with the cruise speed, and only the damping ratio does, from 0.950 to 1.129.
- Integral action rejects a 4 % grade with an integrated error of 18.29 m at every cruise speed.
- Back-calculation bounds windup: after 290 s on the stop the speed overshoots by 7.2 %, against 32 % without it.

Where else this applies. A gain and a time constant that move together with the operating point, at a fixed ratio, also shape the gravity-drained tank of Volume 1, Chapter 2.

References

[1] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Boston, MA: Springer US, 2012.

[2] W. Hucho and G. Sovran, "Aerodynamics of road vehicles," *Annual Review of Fluid Mechanics*, vol. 25, pp. 485–537, 1993.

[3] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[4] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, III, *Process Dynamics and Control*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.

[5] B. W. Bequette, *Process Control: Modeling, Design, and Simulation*. Upper Saddle River, NJ: Prentice Hall, 2003.

[6] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed. Princeton, NJ: Princeton University Press, 2021.

[7] P. A. Ioannou and C. C. Chien, “Autonomous intelligent cruise control,” *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 657–672, 1993.

[8] L. Xiao and F. Gao, “A comprehensive review of the development of adaptive cruise control systems,” *Vehicle System Dynamics*, vol. 48, no. 10, pp. 1167–1192, 2010.

[9] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, “Automated vehicle control developments in the PATH program,” *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, 1991.

[10] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[11] J. C. Maxwell, “On governors,” *Proceedings of the Royal Society of London*, vol. 16, pp. 270–283, 1868.

[12] K. J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2006.

[13] M. V. Kothare, P. J. Campo, M. Morari, and C. N. Nett, “A unified framework for the study of anti-windup designs,” *Automatica*, vol. 30, no. 12, pp. 1869–1883, 1994.

[14] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[15] G. Pannocchia and J. B. Rawlings, “Disturbance models for offset-free model-predictive control,” *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[16] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, “Look-ahead control for heavy trucks to minimize trip time and fuel consumption,” *Control Engineering Practice*, vol. 17, no. 2, pp. 245–254, 2009.

[17] D. Goldfarb and A. Idnani, “A numerically stable dual method for solving strictly convex quadratic programs,” *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[18] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, “Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations,” *IEEE Transactions on Automatic Control*, vol. 48, no. 2, pp. 246–258, 2003.

[19] K. J. Åström and T. Hägglund, “Automatic tuning of simple regulators with specifications on phase and amplitude margins,” *Automatica*, vol. 20, no. 5, pp. 645–651, 1984.

[20] W. Hernandez, “Improving the response of a wheel speed sensor by using a RLS lattice algorithm,” *Sensors*, vol. 6, no. 2, pp. 64–79, 2006.

[21] V. D. Phan, V. C. Le, V. N. Dinh, D. T. Duong, and S. P. Ho, “A novel adaptive terminal sliding mode tracking control of engine electronic throttle system using an enhanced observer under composite disturbances,” *Sensors*, vol. 26, no. 19, p. 6070, 2026.

[22] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, “Guide to operational technology (OT) security,” National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-82r3, 2023.

[23] Z. Wang, G. Wu, and M. J. Barth, “A review on cooperative adaptive cruise control (CACC) systems: Architectures, controls, and applications,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2884–2891.

[24] R. Firoozi, S. Nazari, J. Guanetti, R. O’Gorman, and F. Borrelli, “Safe adaptive cruise control with road grade preview and V2V communication,” arXiv preprint arXiv:1810.09000, 2018.

[25] J. N. Barkenbus, “Eco-driving: An overlooked climate change initiative,” *Energy Policy*, vol. 38, no. 2, pp. 762–769, 2010.

[26] O. M. J. Carsten and F. N. Tate, “Intelligent speed adaptation: accident savings and cost-benefit analysis,” *Accident Analysis & Prevention*, vol. 37, no. 3, pp. 407–416, 2005.

2.13 Exercises

2.1 ○ Concept check · 2 min

Pedal step tests of the same small size are run on the default vehicle from a steady cruise at 10 m/s and at 35 m/s. How do the process gain K_u and the time constant τ fitted at the higher speed compare with those fitted at the lower speed?

(a) The gain is lower and the time constant is the same, because the mass has not changed
(b) The gain is the same and the time constant is lower, because drag acts on the dynamics alone
(c) Both are higher, because the slope of the drag curve grows with the cruise speed
(d) Both are lower by nearly the same factor, so their ratio stays at the thrust-to-mass ratio

2.2 Concept check · 1 min

The vehicle cruises on a flat road in Manual with the pedal fixed, and 400 kg of cargo is added. What does the model predict?

(a) The resting speed is unchanged, and a later pedal step takes longer to settle
(b) The resting speed falls, because a heavier vehicle needs more thrust to hold its speed
(c) The resting speed falls, and a later pedal step settles in the same time as before
(d) The resting speed is unchanged, and a later pedal step settles in the same time as before

2.3 Concept check · 2 min

A 2 % pedal step from a steady 9.04 m/s returns a fitted gain of 1.871 m/(s %). The linearization at the starting speed gives 2.259 m/(s %). Which explanation is right?

(a) The fit is biased low by the speedometer noise, which is largest at low speed
(b) The step raises the speed by 41 %, so the secant gain falls below the tangent gain
(c) The control period adds a dead time that the first-order fit reads as lost gain
(d) The Euler integration step loses a share of the response when the vehicle is slow

2.4 Theory and calculation · 5 min

The default vehicle has a total mass $M = 700\,\mathrm{kg}$, a thrust parameter $F_p = 30\,\mathrm{N/\%}$ and a drag constant $k = 0.735\,\mathrm{kg/m}$. It cruises at 30 m/s on a flat road with no wind.

(a) Find the pedal that holds this speed. *Checkpoint:* 22.05 %.

(b) Find the process gain and the time constant of the linearized model at this speed. *Checkpoint:* 0.680 m/(s %) and 15.87 s.

(c) Find the ratio of the gain to the time constant and compare it with the value at 25 m/s. *Checkpoint:* 0.0429 m/(s^2 %) at both speeds.

2.5 Theory and calculation · 8 min

The default vehicle ($M = 700\,\mathrm{kg}$, $F_p = 30\,\mathrm{N/\%}$, $k = 0.735\,\mathrm{kg/m}$) holds 25 m/s under the default PI tuning, $K_c = 6\,\%\,\mathrm{s/m}$ and $\tau_I = 12\,\mathrm{s}$. The road grade steps from 0 to 5 %.

(a) Find the load force along the road and the pedal increment that rejects it in steady state. *Checkpoint:* 342.8 N and 11.43 %.

(b) Find the integral of the error over the whole rejection, and say whether it changes at a cruise speed of 35 m/s. *Checkpoint:* 22.85 m at either speed.

(c) Integral action is switched off before the grade arrives. Estimate the steady offset. *Checkpoint:* 1.58 m/s.

2.6 Theory and calculation · 5 min

A setpoint is worth tuning for only if the pedal can hold it. Use the steady force balance with $k = 0.735\,\mathrm{kg/m}$ and the pedal at its 100 % stop.

(a) Scenario 8 lowers the thrust parameter to 10 N/% on a flat road with no wind. Find the highest speed the vehicle can hold, and say whether a setpoint of 40 m/s is reachable. *Checkpoint:* 36.89 m/s, so it is not.

(b) With the default 30 N/%, the vehicle carries 600 kg of cargo ($M = 1100$ kg) up an 8 % grade into a 15 m/s headwind. Find the highest road speed it can hold. *Checkpoint:* 38.96 m/s.

2.7 App challenge · 9 min

Load scenario 2, Open-loop actuator step, and change no setting. The pedal steps from 15.31 to 35.31 % at 5 s. Run to 35 s with the default seed and export the CSV file.

(a) Read the true speed, column `true_pv`, at 35 s. *Checkpoint:* 36.56 m/s, within 0.05 m/s.

(b) Compute the speed at which this pedal settles and the gain of the whole step, the settled speed change divided by the pedal change. *Checkpoint:* 37.96 m/s and 0.648 m/(s %).

(c) Find in the file the time after the step at which the true speed has covered 63.2 % of the settled change, and compare it with the time constant of the linearized model at 25 m/s. *Checkpoint:* 14.1 s, within 0.3 s, against 19.05 s.

2.8 App challenge · 8 min

Load scenario 4, Disturbance rejection. The road grade steps from 0 to 5 % at 5 s. Use the default seed and run each case for 60 s.

(a) Run the scenario as shipped. Read the IAE tile of the scorecard, and find in the exported file the lowest true speed and its time. *Checkpoint:* IAE 23.6 m, within 0.2 m. Lowest speed 23.80 m/s at 12.1 s.

(b) Press the reload button, set the Integral time to its top, 120 s, which switches integral action off, and run again. Read the true speed at 60 s. *Checkpoint:* 23.41 m/s, within 0.03 m/s.

Compare both results with the values that Equations (2.10) and (2.11) predict for this grade.

2.9 App challenge · 8 min

Load scenario 9, MPC with the hill on the map. A 5 % grade begins at 20 s. Use the default seed and run each case for 60 s. For each run read the IAE tile of the scorecard and find in the exported file the lowest true speed after 20 s.

(a) Run the scenario as shipped, with the Controller at MPC and Road ahead on. *Checkpoint:* 4.25 m and 24.89 m/s.

(b) Set Road ahead off, press Reset and run again. *Checkpoint:* 4.87 m and 24.58 m/s.

(c) Set the Controller to PID, press Reset and run again. *Checkpoint:* 23.59 m and 23.81 m/s.

Accept an IAE within 0.1 m and a speed within 0.03 m/s. State what the disturbance estimate and the map each contribute.

2.10 Application area · 3 min

The model omits rolling resistance. Take a rolling resistance coefficient of 0.01 on the default vehicle of 700 kg, with $k = 0.735\,\mathrm{kg/m}$ and $F_p = 30\,\mathrm{N/\%}$.

(a) Find the rolling resistance force and the pedal it would add at any speed. *Checkpoint:* 68.6 N and 2.3 %.

(b) Find the speed below which it exceeds the aerodynamic drag, and name one prediction for which the app must not be used as a result. *Checkpoint:* 9.7 m/s.

2.11 Application area · 4 min

A fleet operator compares cruise setpoints of 25 and 30 m/s for a 100 km run on a flat road with no wind, with the default vehicle ($k = 0.735\,\mathrm{kg/m}$).

(a) Find the drag force and the propulsive power at each speed. *Checkpoint:* 459.4 N and 11.5 kW, 661.5 N and 19.8 kW.

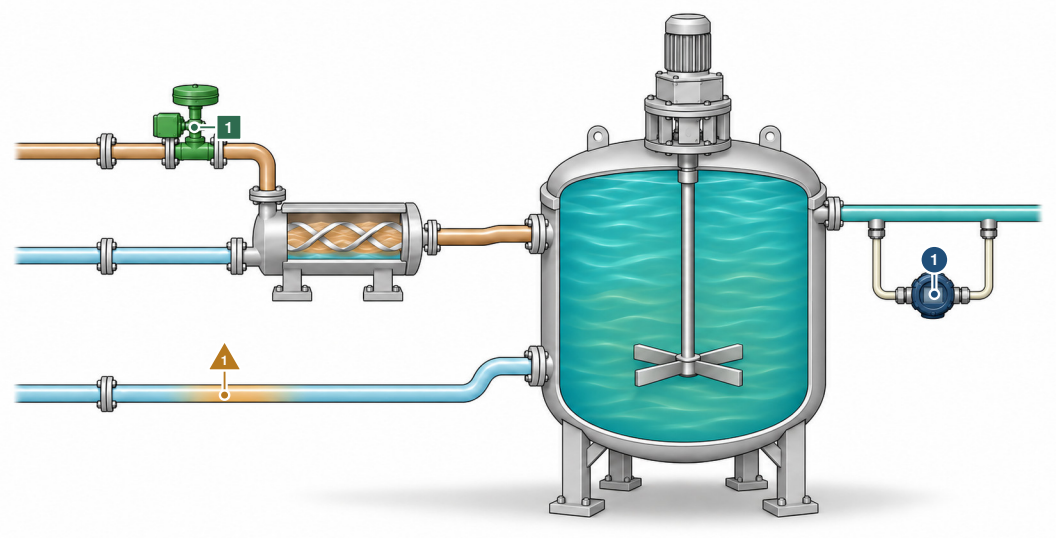
(b) Find the energy spent against drag over the run at each speed and the time saved at the higher speed. *Checkpoint:* 45.9 and 66.2 MJ, and 11.1 min.

2.12 Review the agent · 5 min

An AI assistant is asked to improve the cruise controller and replies as follows. “The process gain is $K_u = F_p/(2k\bar{v})$, so it is 2.5 times higher at 10 m/s than at 25 m/s and 1.4 times lower at 35 m/s. A fixed K_c lets the loop gain change by a factor of 3.5 over the speed range. Schedule the gain as $K_c = 6\,\bar{v}/25$ with $\tau_I = 12$ s, which gives 2.4 at 10 m/s and 8.4 at 35 m/s, and the loop will respond in the same way at every speed.” Find the flaw in the proposal and say how it could be checked.

CHAPTER 3

Stirred Blending Tank Simulator for Composition Control Against an Actuator Limit



The controller moves 1 the valve that sets how strong the feed is

The controller reads 1 an analyzer that reads the strength of the product

What disturbs it 1 plain diluent that waters the product down

Notice that the diluent enters on its own line, so no setting of the valve can make the feed stronger than the concentrate it draws on. Illustration, not to scale.

IN PLAIN TERMS **Holding a Mixture's Strength With a Limited Supply**

The main idea. A blending tank mixes a strong feed with a plain stream that carries none of the ingredient. A controller reads the product and makes the feed stronger or weaker to hold a chosen strength as more of the plain stream comes in.

The challenge. The feed can never be stronger than the supply it comes from. When too much of the plain stream arrives, even the strongest feed falls short, and no setting of the controller can close the gap.

Why it matters. Before blaming the tuning for a lasting error, check whether the supply could reach the target at all.

THREE WAYS TO READ THIS CHAPTER

Explore	12 min	the first page, the history of reset (Section 3.1), the history of model-based tuning (Section 3.4), the try-it segment of case 8 (Section 3.6), the try-it segment of case 4 and the two facts (Section 3.7) and the key points
Study	70 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	25 min	the first page, the results (Section 3.7), Technology in Practice (Section 3.8), Limitations (Section 3.9) and the open challenge

The control question A blending loop manipulates a feed concentration that is bounded by what the upstream supply contains. Which standing errors can a controller remove, and at what dilution flow does a composition setpoint pass out of reach of any tuning?

Learning objectives

- Derive the gain, the residence time and the steady-state inverse of the tank from the species balance.
- Explain why a dilution change moves the integral gain of an internal model control tuning and leaves the proportional gain alone.
- Decide from the steady-state balance whether a setpoint is reachable, and compute the offset when it is not.
- Distinguish the offset of a proportional loop from the offset that an actuator bound imposes.
- Describe how back-calculation holds the stored output at the bound, and what a model predictive controller adds at the limit.

Prerequisites Sections 1.4.2 and 1.6.2 (velocity form and offset), 1.4.4 (step test), 1.4.5 (tuning rule) and 1.4.9 (windup). Sections 1.4.7 and 1.6.3 are needed only for the relay test and the predictive controller. Chapter 2 uses the same algorithm.

The app https://apmonitor.com/apps/blend/

See also Volume 1, Chapter 2 and Volume 1, Chapter 3 for step tests on tanks whose model moves with the operating point. Volume 4, Chapter 9 for cases that separate a tuning problem from missing capacity. Volume 13, Chapter 5 for feeds held in ratio with analyzer trims and Volume 13, Chapter 10 for a second handle that keeps the first off its limit.

3.1 Introduction

Composition is the product property that a specification or a permit constrains, and blending is the operation that delivers it. Streams are combined so that the mixture meets a number written into a product specification or a permit, and the combination is enforced continuously on gasoline in a refinery header [1–3], on coagulant in a water treatment works [4], and on active ingredient in a continuous tablet line [5, 6]. The stirred vessel that performs the mixing is among the earliest systems studied in chemical engineering dynamics, and its residence time behavior was formalized as a distribution rather than a single number in 1953 [7]. A composition loop built on such a vessel is the standard first example of process dynamics in the textbooks, because a species balance on a well-mixed inventory reduces conservation of mass to a single linear ordinary differential equation [8, 9]. There is one state, no dead time when the mixing is ideal, a gain that is close to unity, and a first-order response that any proportional-integral controller can be made to follow.

What separates a blending loop from the level, temperature and flow loops taught beside it is the nature of its manipulated variable. A temperature loop moves a valve, and the travel of that valve is a sizing decision that can be revisited. A blending loop moves a composition or a ratio, and its ceiling is set by what the upstream tank contains. Raising the concentrate strength is a procurement decision rather than a tuning decision. The consequence is that a large enough load disturbance does not degrade the loop. It removes the setpoint from the reachable set, and the resulting steady-state error is a property of the mass balance rather than of the controller.

The usual treatment presents offset as a deficiency with a known remedy. A proportional controller leaves an error, integral action removes it, and the integrator that removes it is then protected against saturation by an anti-windup mechanism so that the loop recovers promptly when the limit clears [8, 10]. Sections 1.4.2 and 1.4.9 present the two steps in that order. Each of those statements is correct, and taken together they suggest that any standing error is a tuning failure. However, an actuator bound makes some standing errors a property of the steady-state balance, and the literature that treats saturation as a design problem is concerned with recovery after the limit clears rather than with whether the setpoint was reachable at all [11–14]. A reader who has only encountered the removable offset may respond to the irremovable one by raising the gain.

> ⧖ HISTORY **Early Process Controllers and Reset, 1930s**
>
> By the mid 1930s many companies supplied sensors, actuators and controllers to the process industry. A proportional controller holds its setpoint without error only if one more number is right: the steady output that the process needs, called the reset value. In early controllers it was adjusted by hand. Automatic reset, one of the early inventions of integral control, produced it by passing the controller output through a low-pass filter and feeding it back with positive gain, which was easier to build than an integrator [10]. Integral action is in this sense a self-resetting bias, which is why it is also called reset. An older name is floating [15]. The stored output of this chapter's PID is that reset value.

This chapter works with a stirred blending tank simulation in which the manipulated feed concentration is bounded at 0 to 2 mol/L. It quantifies the dilution flow at which a composition setpoint passes from a tuning limitation to a capacity limitation. The experiments built on the simulation cover process gain and time constant identification from a step test, proportional offset, integral action, actuator saturation, anti-reset windup and steady-state reachability, all of which are standard topics in a first course on process dynamics and control [8–10].

The chapter is organized as follows. Section 3.2 places the composition loop in blending practice. Section 3.3 specifies the simulator, covering the balance, the bound, the instrument, the constants and the integration scheme in enough detail for a second implementation to reproduce the same trajectories. Section 3.4 shows that the dilution stream moves the gain and the time constant together. As a result, internal model control leaves the proportional gain of the loop untouched by the dilution while moving its integral gain, and both results are checked against step tests run in the application. Section 3.5 describes the velocity-form PID with its back-calculation, a linear model predictive controller offered beside it, a moving horizon estimator that may stand in for the filter of that controller, and a relay test that sets the PID constants from the running loop. Section 3.6 derives the reachability inequality for this plant in closed form. Section 3.7 reports the measured runs. They are built around two standing errors of similar size, 0.149 mol/L left by a proportional-only loop against a 33 m^3/min dilution and 0.167 mol/L imposed by the bound at 50 m^3/min, only the first of which is removable by feedback. The same section lists the ten shipped cases with their completion requirements, compares the predictive controller with the PID to separate what a model of the load contributes from what an explicit treatment of the bound contributes, and assesses the estimator and the relay on the same cases. Section 3.8 describes the devices and the platform of a real blend loop, Sections 3.9 and 3.10 state the limitations of the model and its economic context, Section 3.11 summarizes the results, and the exercises follow in Section 3.12.

3.2 Background

Blending entered the engineering literature as an optimization problem before it entered as a control problem. The aviation gasoline blend written as a linear program in 1952 is one of the first industrial applications of linear programming of any kind, and it was posed exactly as a blending question, which combination of available stocks meets every octane, volatility and vapor pressure specification at the lowest cost [1]. That economic framing remains current. The economic incentive in a modern blend is giveaway, the quality delivered above the specification that the customer pays for, and moving the average blend closer to its limit has value in direct proportion to how tightly the composition can be held [2]. Refinery blend planning is still posed as a simultaneous optimization over recipes and schedules, with the nonlinearity of the quality models and the discrete structure of the tank assignments as the two difficulties [3]. Volume 11, Chapter 15 treats an in-line gasoline blender with a recipe optimizer, whose product is certified from the average in its tank.

Feedback control of blending became practical when the in-line analyzer arrived. An offline laboratory assay supports recipe adjustment between batches, while an in-line analyzer supports feedback within a blend, and the second changes what the optimization is allowed to assume. Model-based real-time optimization of a gasoline blend header reduces giveaway by driving the blend toward

Table 3.1 Classification of Variables. Nominal Values Are Those of the 1.5 mol/L Operating Point With the Diluent Stream Closed.

Symbol	Role	Nominal	Range
C_A	state, true concentration	1.50 mol/L	≥ 0
$\hat{C}_A$ (AT)	CV, analyzer reading	1.50 mol/L	≥ 0
C_A^{SP}	setpoint	1.50 mol/L	0 to 2, step 0.01
C_{Af}	MV, feed concentration	1.50 mol/L	0 to 2, step 0.01
q_d	DV, diluent flow	0 m^3/min	0 to 100, step 1
q	parameter, feed flow	100 m^3/min	20 to 200, step 5
T_f	DV, feed temperature	300 K	270 to 370, step 1
$q+q_d$	output, outlet flow	100 m^3/min	≥ 0
τ	output, residence time	1.00 min	$V/(q+q_d)$
limit flag	output, feed at a bound	false	—

the specification and relying on the regulatory layer to hold it there, so the achievable margin is set by the variance of the composition loop [2]. Water treatment carries the same structure with a different specification. Coagulant dose is manipulated against a raw water quality that changes with weather and season, and the reason online monitoring of the coagulation process has attracted sustained attention is that dosing too little fails the treatment objective while dosing too much wastes chemical and loads the sludge [4]. Continuous pharmaceutical manufacturing has adopted the same architecture most recently, with blend uniformity held by feedback around a continuous mixer inside a plantwide control structure that mixes model-based and proportional-integral-derivative loops [5].

The modeling side of the subject is older and is dominated by one question, how far a real vessel departs from perfect mixing. The residence time distribution was introduced to answer it, with the perfectly mixed tank and the plug flow tube as the two limiting distributions and every real vessel somewhere between [7]. The same tool is now the standard characterization of continuous solids equipment, where the width of the distribution sets how much an upstream feeder fluctuation is smoothed before it reaches the tablet press [6]. A perfectly mixed vessel is the limiting case in which the outlet composition equals the inventory composition and the transfer function is a single first-order lag with no dead time, which is the case the application implements.

3.3 Process Model

The simulated vessel is a closed, liquid-full, agitated tank fed by two streams. The first is a concentrated stream of fixed flow whose concentration is set by the controller, and the second is a pure diluent stream of variable flow whose concentration is zero. Because the vessel is liquid full and the liquid is incompressible, the outlet flow is the sum of the two inlet flows at all times and the volume is not a state. The controlled variable is the outlet concentration reported by analyzer AT, the manipulated variable is the feed concentration, and the diluent flow is left to the operator whichever mode the loop is in. Figure 3.1 places every symbol used below on the equipment.

Table 3.1 sorts the signals into their roles. The feed flow q is a model parameter that the operator can change during a run, and it appears in every model coefficient. The feed temperature is measured, recorded in the exported file, and deliberately absent from every equation. Concentrations are carried in mol/L, flows in m^3/min, volume in m^3 and time in min, and no conversion between m^3 and L appears in the source because the factor divides out of every term of the balance.

3.3.1 Species Balance

A solute balance over the liquid-full vessel, with the inlet solute carried entirely by the concentrated stream and the outlet composition equal to the inventory composition, is shown in Equation (3.1).

$$V\frac{dC_A}{dt} = qC_{Af} - (q+q_d)C_A, \qquad 0 \leq C_{Af} \leq C_{Af}^{\max} \tag{3.1}$$

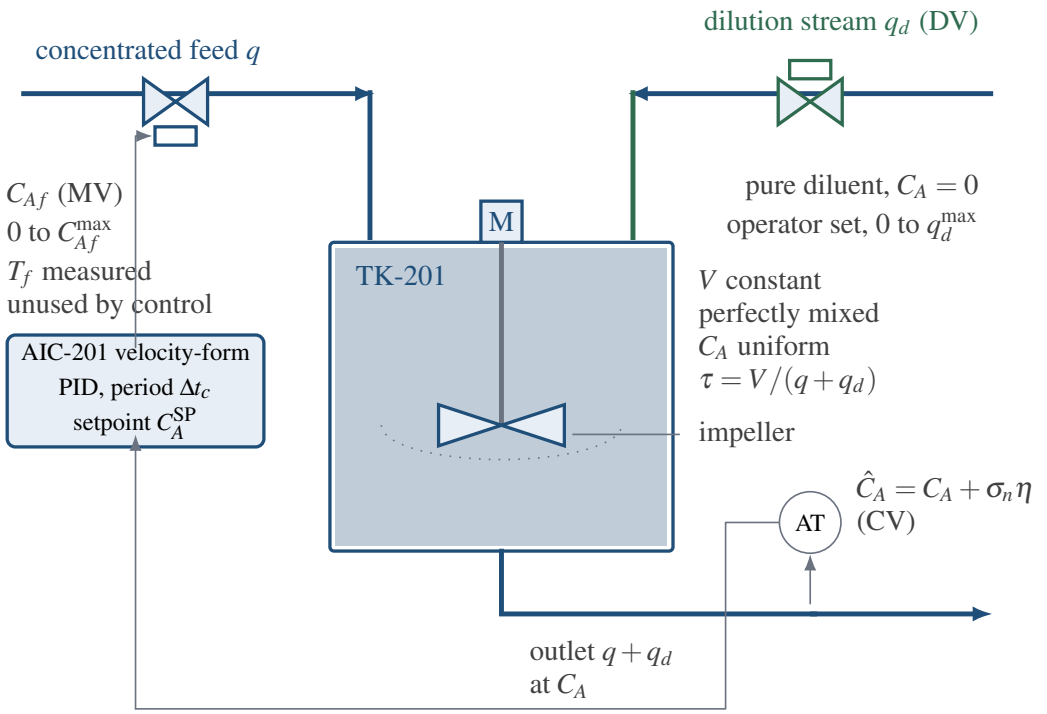


Figure 3.1 Blending tank TK-201 with its concentrated feed, its diluent stream and the outlet composition analyzer. The feed concentration C_{Af} is the manipulated variable and is bounded at $C_{Af}^{\max}$, the diluent flow q_d is the load disturbance and carries no solute, and the vessel is liquid full so the outlet flow is $q+q_d$. Controller AIC-201 reads the noisy analyzer value $\hat{C}_A$ and sets the feed concentration.

where V is the liquid volume in m^3, C_A is the outlet and inventory concentration in mol/L, C_{Af} is the concentration of the feed stream in mol/L, q is the feed flow and q_d is the diluent flow, both in m^3/min. Every term carries units of m^3/min times mol/L. The diluent contributes no term on the right because its concentration is zero, and it acts only through the outlet term, which is the mechanism by which it disturbs the loop.

Two structural facts follow from the shape of Equation (3.1). The volume V is a constant and not a state, because the vessel is liquid full and the outlet flow is defined as the sum of the inlets rather than by a valve or a pump. There is no total mass balance to integrate and no level loop. The equation is also linear in C_A at fixed flows, so the nonlinearity of the plant is entirely in the coefficients. Both the coefficient of C_{Af} and the coefficient of C_A move when a stream moves, which is what makes a diluent flow change a retuning of the plant rather than an additive load.

Holding the feed concentration and the two flows constant gives the first-order response of Equation (3.2).

$$C_A(t) = \bar{C}_A + \left[C_A(0) - \bar{C}_A\right] \mathrm{e}^{-t/\tau}, \qquad \bar{C}_A = \frac{q}{q+q_d} C_{Af} \tag{3.2}$$

where the residence time is $\tau = V/(q+q_d)$ in min. The steady-state gain from feed concentration to outlet concentration is $K_p = q/(q+q_d)$, which is dimensionless and is exactly unity when the diluent valve is shut. There is no dead time anywhere in the plant, because a perfectly mixed vessel begins to respond in the first instant and the outlet is drawn from the same well-mixed inventory that the feed enters.

Table 3.2 Model and Controller Parameters

Symbol	Meaning	Default	Range	Unit
Configuration parameters				
V	liquid volume	100	20–300	m^3
q	feed flow	100	20–200	m^3/min
σ_n	analyzer noise	0.01	0–0.05	mol/L
Operating parameters				
C_{Af}	manual feed concentration	1.50	0–2	mol/L
C_A^{SP}	concentration setpoint	1.50	0–2	mol/L
q_d	diluent flow	0	0–100	m^3/min
T_f	feed temperature	300	270–370	K
K_c	proportional gain	2.0	0–10	—
τ_I	integral time (20 = off)	1.0	0.01–20	min
τ_D	derivative time	0.0	0–2	min
—	controller	PID	PID, MPC	—
T_p	MPC prediction horizon	4	0.4–8	min
λ	MPC move suppression	30	0–100	—
—	MPC estimator	Kalman	or MHE	—
—	speed multiplier	1	0.5–10	—
Set when a run starts				
$C_A(0)$	initial concentration, steady	1.50	—	mol/L
u_b	controller bias, starting feed	1.50	—	mol/L
Fixed in the source				
C_{Af}^{max}	feed concentration limit	2.0	—	mol/L
Δt	integration step (Euler)	0.01	—	min
Δt_c	controller sample period	0.02	—	min
Δt_h	logging interval	0.1	—	min
T_s	MPC decision period	0.04	—	min
N_e	MHE window	10	—	executions
ε_r	relay band	0.03	—	mol/L

3.3.2 Analyzer

What the control law sees is not the state of Equation (3.1) but the analyzer output of Equation (3.3).

$$\hat{C}_A = \max(0,\, C_A + \sigma_n\, \eta) \tag{3.3}$$

where σ_n is the analyzer noise standard deviation in mol/L and η is a fresh standard normal deviate at each controller execution, obtained through a Box-Muller transformation from the seeded generator of the engine (Section 1.6.1), which is reseeded with the same seed whenever a run starts. Nothing else stands between the state and the reading. There is no analyzer lag, no sample transport delay, no calibration bias and no drift, and the noise is white at the control rate. The true concentration and the measured value are both displayed and both exported, which puts the instrument error beside the state it corrupts.

Table 3.2 holds every number a second implementation would need. Two numbers of the default configuration are deliberately unphysical. A feed of 100 m^3/min carrying 2 mol/L of solute is a throughput no blend drum of 100 m^3 would ever see, but it makes the residence time exactly 1 min and fits a complete step response into a window that can be watched at ordinary speed. A diluent stream reaching half the feed flow would not occur in a physical system, but it is scaled that way so that the boundary of Equation (3.19) can be crossed with one input change.

Configuration values are guarded before use, with V raised to at least 1 m^3 and both q and σ_n raised to at least zero, and the feed concentration is clipped to $[0, C_{Af}^{max}]$ as it enters the model whichever mode produced it. The proportional gain is dimensionless because the manipulated variable and the controlled variable are both concentrations, which is uncommon among process loops. The integral

and derivative gains of the parallel form (Section 1.4.2) follow from the two times as $K_I = K_c/\tau_I$ in 1/min and $K_D = K_c\tau_D$ in min, so the default tuning carries $K_I = 2.0/\text{min}$, and the top of the integral-time range, 20 min, switches integral action off. Every case starts at steady operation. In automatic the tank is placed at the steady state of the setpoint in force, and in manual at the steady state of the manual feed, with the flows and parameters in force, so the defaults of the second group in Table 3.2 are those of the reference case.

3.3.3 Numerics and Clocks

A fixed 0.01 min step and explicit Euler advance Equation (3.1), with the state floored at zero afterwards. The app supplies this step in place of the Runge–Kutta default of the framework (Section 1.6.1). The most demanding configuration the input ranges allow is $q = 200\,\text{m}^3/\text{min}$ with $q_d = 100\,\text{m}^3/\text{min}$ and $V = 20\,\text{m}^3$, giving $\tau = 0.067\,\text{min}$ and a ratio $\Delta t/\tau$ of 0.15, far inside the stability bound of the method across the whole parameter range. Control acts on every second step, so the sample period is 0.02 min of process time, and one historian row is written every 0.1 min, at every fifth execution. Neither clock is driven by wall time. The engine keeps them apart from the drawing of the screen, as Section 1.6.1 describes, which leaves the trajectory and the exported time base identical at every speed. One second of wall time corresponds to 1 min of process time at $1\times$.

Within one step the order is that of Figure 1.10, and it is visible in the exported file. Scheduled events are applied first, then, on every second step, the feed concentration is chosen by the control law or copied from the manual input and applied, and the balance is advanced across the step. The analyzer reading for an execution is drawn when the step ending at that instant completes, from the state at that instant, and a row is written at the end of the step. Rows fall on every fifth execution, so each row holds the noise-free concentration and the analyzer reading the controller acts on at its time stamp, beside the feed applied over the 0.02 min period that ends there. A time constant read graphically from the reading column is therefore not displaced, while the feed column trails its time stamp by one sample period.

3.3.4 Assumptions

A mechanism appears in the model when it changes a commissioning decision and not otherwise. Every item listed here is given with the conditions that make it acceptable and the conditions that break it.

1. *Perfect mixing.* The inventory is uniform, so the outlet concentration equals the tank concentration at every instant and the residence time distribution collapses to the single exponential of Equation (3.2). Real agitated vessels deviate from that limit, and the deviation is normally reported as the spread of a measured residence time distribution [6, 7]. The consequence here is that the simulated plant has no dead time at all, while a real blend header carries transport delay from the mixing point to the analyzer and a finite analyzer cycle. A loop tuned on this simulator is therefore tuned for the most forgiving composition plant that exists.

2. *Constant volume.* The vessel is liquid full and incompressible, so the outlet flow equals the sum of the inlets instantaneously and the volume never appears as a state. A real blend drum with a level controller would pass an inlet flow change to the outlet only after the level loop had acted, which adds a second dynamic element between the diluent valve and the outlet composition. None of that is represented.

3. *Ideal streams and constant density.* The two streams mix without volume change, the solute is dilute enough that the concentrated feed carries the same volumetric flow whatever its concentration, and no density is used anywhere in the source. A concentrate strong enough to matter would change the volumetric balance as well as the solute balance.

4. *Feed concentration as a direct actuator.* The manipulated variable is the concentration of the feed itself, applied with no valve, no blend dynamics upstream and no lag. In practice that

concentration is produced by a ratio of two streams or by a dosing pump, and the loop would be a cascade whose inner element has its own time constant, rangeability and stiction. Volume 13, Chapter 5 treats such a ratio station, with feeds held in ratio to the cell current and trimmed by analyzers. The simulator collapses that inner element to an ideal instantaneous setting, so the only actuator imperfection retained is the bound itself.

5. *Ideal analyzer apart from noise.* AT has no lag, no dead time, no drift, no calibration error and no resolution limit, and the noise it adds is Gaussian and white. Real composition measurements are dominated by sampling delay and by low-frequency calibration error, neither of which is white, and the cycle time of a real analyzer is frequently long enough to be the dominant dynamic element in the loop. Volume 5, Chapter 3 measures what an analyzer lag removes from a composition loop.

6. *Isothermal operation.* No energy balance is integrated. The feed temperature is measured and recorded because the model signature that the application inherited carries it, and it enters no equation. Any real blend in which viscosity, solubility or reaction rate depends on temperature would need that balance.

7. *Omitted entirely.* Upstream flow disturbances on the concentrated stream, a second solute, reaction in the vessel, agitator failure, level control, heat effects and any second vessel are absent. None of them alters the one behavior this application is built to show, which is what a bounded feed composition does to a setpoint, so all of them are left out.

3.4 Dynamic Analysis

3.4.1 Dependence of Gain and Time Constant on Throughput

Both constants of the first-order model in Equation (3.2) carry the same denominator, as collected in Equation (3.4).

$$K_p = \frac{q}{q+q_d}, \qquad \tau = \frac{V}{q+q_d}, \qquad \theta_p = 0 \tag{3.4}$$

where K_p is dimensionless, τ is in min and θ_p is the dead time. Opening the diluent valve shrinks both. At the nominal $q = 100\,\mathrm{m^3/min}$ and $V = 100\,\mathrm{m^3}$ the plant is a unity-gain lag of 1.00 min, and at $q_d = 50\,\mathrm{m^3/min}$ the same vessel is a plant of gain 0.667 and time constant 0.667 min. The response becomes faster and weaker at the same time, while the more common pattern in process loops pairs a slower response with a stronger one. The gravity-drained tank of Volume 1, Chapter 2 and the vehicle of Chapter 2 show the same pairing, a gain and a time constant that move together at a fixed ratio.

The feed flow acts differently, because it appears in the numerator of the gain as well as in the shared denominator. With the diluent valve shut, K_p is unity whatever q is, so a feed flow change moves only the time constant. Doubling q from 100 to $200\,\mathrm{m^3/min}$ halves τ from 1.00 to 0.50 min and leaves the resting concentration unchanged. The residence time is set by the vessel volume and the throughput rather than by the controller, so a faster composition response requires a higher throughput or a smaller inventory.

3.4.2 Internal Model Control Tuning

For a first-order plant carrying no dead time, internal model control tuning gives $K_c = \tau_p/(K_p\tau_c)$ and $\tau_I = \tau_p$, with τ_c the specified closed-loop time constant [16, 17]. This is the rule of Equation (1.11) with $\theta_p = 0$ and τ_c large enough that the integral time stays at τ_p (Section 1.4.5). Substituting Equation (3.4) cancels the throughput term from the proportional gain but not from the integral gain, as shown in Equation (3.5).

$$K_c = \frac{V}{q\,\tau_c}, \qquad K_I = \frac{K_c}{\tau_I} = \frac{q+q_d}{q\,\tau_c} \tag{3.5}$$

Table 3.3 Step Tests From $C_{Af} = 1.0$ to 1.5 mol/L With $\sigma_n = 0$ and $V = 100\,\mathrm{m}^3$. Predictions Are Equation (3.4).

q	q_d	$\hat{K}_p$	K_p	$\hat{\tau}$	τ	t_{63}	R^2
(m^3/min)					(min)		
100	0	1.0000	1.0000	0.9950	1.0000	0.9952	1.00000
100	25	0.8000	0.8000	0.7950	0.8000	0.7953	1.00000
100	50	0.6667	0.6667	0.6617	0.6667	0.6634	1.00000
200	0	1.0000	1.0000	0.4950	0.5000	0.4954	1.00000

where K_c is dimensionless and K_I is in 1/min. The required proportional gain depends on the vessel volume and the feed flow and not at all on the diluent stream, because gain and time constant shrink in step and their quotient is a constant of the vessel. The integral gain depends only on the dilution ratio $(q+q_d)/q$ and on τ_c. With the default $V = 100\,\mathrm{m}^3$, $q = 100\,\mathrm{m}^3$/min and a moderate $\tau_c = 0.5$ min, Equation (3.5) gives $K_c = 2.0$ and $K_I = 2.0$/min, which are the default $K_c = 2.0$ and $\tau_I = 1.0$ min, and opening the diluent valve to 50 m^3/min raises the correct K_I to 3.0/min ($\tau_I = 0.667$ min) while leaving K_c at 2.0. A dilution change therefore requires retuning of the integral gain only.

Pause. Try Exercise 3.1 before reading on.

> HISTORY **Tuning From a Model, 1986 to 2003**
>
> In 1986 a paper on internal model control derived PI and PID settings analytically from a process model. These IMC-PID rules achieved widespread industrial acceptance. They proved robust and gave very good responses to setpoint changes, but a poor response to load disturbances on integrating processes. A paper of 2003 kept the model-based start and changed the rule for the integral term, so that a single rule serves a first-order or second-order model with time delay, for setpoints and for load disturbances. Its integral time is the smaller of the process time constant and four times the sum of the closed-loop time constant and the time delay [17].

3.4.3 Identification From the Running Application

Four manual step tests are collected in Table 3.3, and Section 1.4.4 gives the step test and its fit in general. Each starts from the reference case with the analyzer noise switched off and the flows and a 1.0 mol/L feed set before the run, so the tank starts at the steady state of that feed, and raises the feed concentration by 0.5 mol/L 1 min into the run. Gain is the secant between the starting state and the state 10 min after the step. The time constant comes from a least-squares exponential fitted to the logged noise-free concentration with the initial value, the final value and τ all free, and t_{63} is read from the logged analyzer column by linear interpolation.

The secant gains reproduce $q/(q+q_d)$ to four decimals in all four tests, because with the noise switched off the starting state is the exact steady state of Equation (3.2). The largest shortfall, 0.00004 at $\tau = 1$ min, is the part of the step that a lag of that length has not completed 10 min after it. The fitted time constants sit below the analytic values by 0.50 to 1.00 %, and the explicit Euler integrator accounts for all of it, because a deviation decays by a factor $1 - \Delta t/\tau$ per step against the $\mathrm{e}^{-\Delta t/\tau}$ of the exact solution and the discrete response therefore runs slightly fast. The equivalent discrete time constant, $-\Delta t/\ln(1-\Delta t/\tau)$, is 0.9950, 0.7950, 0.6617 and 0.4950 min over these four flows, equal to the fitted values to four decimals. The graphically read t_{63} exceeds the fitted time constant by 0.0002 to 0.0017 min, which is the error of interpolating linearly between logged rows 0.1 min apart, because each logged reading is taken at its own time stamp. Both discrepancies are smaller than the difference between any two rows of the table, so neither obscures the scaling result.

Three distinct models appear in this chapter, and two of them consist of one or two numbers. Equation (3.1) with its bound is the simulation, exact inside the assumptions of Section 3.3.4. The pair $(\hat{K}_p, \hat{\tau})$ in Table 3.3 is the identified model, and it describes the plant at one pair of flows and nowhere else. Whatever model the PID holds is carried by the sum of its bias and its integral state, a

single number that starts at the feed of the starting steady state and settles at the steady-state inverse of the plant derived in Section 3.6. The PID gains carry no model of the tank. The predictive controller of Section 3.5.4 carries the second model itself, the pair of the first row of Table 3.3, and keeps it at every dilution.

3.5 Controller Design

3.5.1 Velocity Form With Derivative on the Measurement

The PID is the velocity-form algorithm of the framework, which Section 1.6.2 gives in general, and this subsection writes it in the symbols of the composition loop. Each cycle the algorithm returns an adjustment to the feed concentration rather than the concentration itself, and the operating point is held in an integral state rather than in a bias. Error is defined in Equation (3.6) as the setpoint less the analyzer reading, and Equations (3.7) to (3.9) give the three contributions to the increment.

$$e_k = C_A^{\mathrm{SP}} - \hat{C}_{A,k} \tag{3.6}$$

$$\Delta u_k^P = K_c\,(e_k - e_{k-1}) \tag{3.7}$$

$$\Delta u_k^I = \frac{K_c}{\tau_I}\,e_k\,\Delta t_c \tag{3.8}$$

$$\Delta u_k^D = -K_c\,\tau_D\,\frac{\hat{C}_{A,k} - 2\hat{C}_{A,k-1} + \hat{C}_{A,k-2}}{\Delta t_c} \tag{3.9}$$

where K_c is dimensionless, τ_I and τ_D are in min and $\Delta t_c = 0.02\,\mathrm{min}$. The proportional and derivative increments are accumulated, the demand is clipped to the bound, and the integral is updated afterwards, as shown in Equation (3.10).

$$\begin{aligned} s_k &= s_{k-1} + \Delta u_k^P + \Delta u_k^D \\ c_k &= u_b + s_k + I_{k-1} \\ C_{Af,k} &= \min\!\left(C_{Af}^{\max},\ \max(0,\ c_k)\right) \\ I_k &= I_{k-1} + \Delta u_k^I + \frac{\Delta t_c}{\tau_I}\left(C_{Af,k} - c_k\right) \end{aligned} \tag{3.10}$$

Here s_k accumulates the proportional and derivative increments without limit, c_k is the feed concentration the algorithm demands, the bias u_b is the feed concentration that holds the starting steady state, the integral starts at zero, and the integral used at execution k is the one accumulated through execution $k-1$. In the notation of Equations (1.23) to (1.26), s_k is the running sum S_k and c_k is the output before limiting v_k. The last term is a back-calculation correction with a tracking time equal to τ_I. Away from the bound $C_{Af,k} = c_k$, the correction vanishes, and the recursion is algebraically identical to the positional law $C_{Af,k} = u_b + K_c e_k + (K_c/\tau_I)\sum_{j<k} e_j \Delta t_c$ plus the derivative term. At the bound the two differ, because only the integral is corrected. At the 20 min top of its range the integral time switches integral action off, and then neither increment nor correction is applied in automatic, so the law is proportional and derivative action about $u_b + I$. The derivative acts on the measurement rather than on the error [18], as Section 1.4.8 explains, so a setpoint change reaches the feed only through the proportional term. The controller is reverse acting in the ISA sense (Section 1.4.3), because more solute in the feed means more solute in the outlet. With $e_k = C_A^{\mathrm{SP}} - \hat{C}_{A,k}$ a positive K_c delivers that action, and no action switch is offered.

3.5.2 Initialization and Transfer

The bias and the integral state carry the operating point that a positional law carries in its bias alone. A case that opens in automatic places the tank at the steady state of its setpoint and sets u_b to the feed concentration that holds it, the value of Equation (3.18), with the integral at zero, so its first execution issues Equation (3.11).

$$C_{Af,0} = \min\!\left(C_{Af}^{\max},\ \max(0,\ u_b + K_c\,e_0)\right) \tag{3.11}$$

where u_b is 1.50 mol/L at the default setpoint with the diluent shut, and the error e_0 is the analyzer noise alone. A case that opens in manual places the tank at the steady state of the manual feed u_m and sets $u_b = u_m$. A setting changed before a run starts, whether a setpoint, a flow, a tuning value or the mode, is part of these starting conditions, so the run that follows is the one the same setting gives when carried through Reset. Transfer between manual and automatic is bumpless in both directions by the tracking of Sections 1.4.10 and 1.6.2. In manual the controller sets $I_k = u_m - u_b - s_k$ at each execution so that its demand equals the manual feed, whatever error stands at that moment. The sum $u_b + I$ at the moment of transfer therefore plays the role of the operator-entered bias of a positional law, and of the bias in the offset analysis of Section 3.7. The transfer back to manual holds the last controller output to its full precision. Edits made in automatic act through the increments alone, because the increments are differences of the complete proportional and derivative terms, so a setpoint change moves the feed by $K_c \Delta C_A^{\mathrm{SP}}$ and a gain change on a settled loop moves nothing.

3.5.3 Back-Calculation as the Anti-Windup Mechanism

Anti-windup is the last term of Equation (3.10) rather than a separate stage, and it is one of the remedies that Section 1.4.9 compares on a reachable setpoint. While the feed is clipped, the correction drives the integral toward the value that reproduces the applied feed with a tracking time of τ_I, and the running sum s_k is never truncated [11, 12]. With $\tau_D = 0$ the sum s_k telescopes to $K_c e_k$, so on the upper bound $c_k - C_{Af}^{\max} = u_b + K_c e_k + I_{k-1} - C_{Af}^{\max}$, and the error term of the correction cancels Δu_k^I exactly. What remains is Equation (3.12), the sampled counterpart of Equation (1.18).

$$u_b + I_k - C_{Af}^{\max} = \left(1 - \frac{\Delta t_c}{\tau_I}\right)\left(u_b + I_{k-1} - C_{Af}^{\max}\right) \tag{3.12}$$

The stored output $u_b + I$ therefore relaxes to the bound itself with time constant τ_I whatever the error, and the demand exceeds the bound only by the proportional term $K_c e_k$. Once the stored output has relaxed, the feed leaves the bound when the error changes sign, which is when the outlet crosses the setpoint, and no stored integral has to be worked off. The application offers no free integrator, so it cannot be used to demonstrate the failure mode itself, only its absence, and the size of the failure it avoids has to be computed from the measured error history rather than observed.

That computation quantifies the failure the anti-windup literature addresses [18]. During the constrained upset of Section 3.7 the feed sat on its 2.00 mol/L bound for 9.34 min and accumulated an error integral of 1.559 mol min/L. A positional algorithm with a free integrator and the same $K_I = 2.0/\mathrm{min}$ would have carried 3.12 mol/L of integral demand above the bound at the instant the disturbance was removed, and would have had to accumulate the same 1.559 mol min/L of negative error area before the feed could begin to come down. With the outlet at 2.0 mol/L against a 1.5 mol/L setpoint that unwinding would take 3.1 min of full overshoot. The measured bias and integral sum to 2.0000 mol/L (1.5000 and 0.5000) when the dilution closes, which is the bound itself, and the feed leaves the bound 0.30 min later, when the outlet crosses the setpoint. The comparison is a calculation and not a measurement, because the algorithm it describes is not implemented.

3.5.4 Model Predictive Control

A linear model predictive controller (MPC) can hold the feed concentration in automatic mode in place of the PID, which remains the default of every case except case 9. It follows the general arrangement of Section 1.6.3, and this subsection gives the model, the estimator and the settings it carries for the tank. Every $T_s = 0.04$ min, two executions of the control law, it solves for a sequence of feed concentrations over a horizon, applies the first of them and keeps it until the next decision [19, 20]. The prediction model is Equation (3.13), identified on the simulator at the design point $\bar{C}_A = \bar{C}_{Af} = 1.50\,\mathrm{mol/L}$ with the diluent shut and the analyzer noise off, from the averaged responses to feed steps of $\pm 0.1\,\mathrm{mol/L}$.

$$\tau_m \frac{\mathrm{d}x}{\mathrm{d}t} = -x + K_m\left(C_{Af} - \bar{C}_{Af} + d\right), \qquad \tilde{C}_A = \bar{C}_A + x \tag{3.13}$$

Here x is the model state in mol/L, $\tilde{C}_A$ is the outlet concentration the controller predicts, d is an unmeasured disturbance expressed as an equivalent feed concentration in mol/L, $K_m = 1.000$ and $\tau_m = 0.995$ min. The fit finds no dead time and leaves a root-mean-square residual below 1×10^{-6} mol/L.

The fitted pair repeats the first row of Table 3.3, and the model sampled at Δt_c shrinks a deviation by $0.99^2 = 0.9801$ per execution, exactly as the Euler step of the simulator does with the diluent shut. The model contains no flow. Against 20 m³/min of diluent the tank has $K_p = 0.833$ and $\tau = 0.833$ min, but their ratio $K_p/\tau = q/V$ does not depend on q_d, which is also why the dilution leaves the proportional gain of Equation (3.5) alone. The model is therefore right about the start of a feed move at any dilution and wrong only about where the move settles. A feed step of 1 mol/L raises the predicted outlet by 0.0394 mol/L after one decision period and by 0.982 mol/L at the end of the horizon, against 0.0393 and 0.827 mol/L for the diluted tank.

Offset-free control comes from the estimator. A steady-state Kalman filter on Equation (3.13), sampled with a zero-order hold at Δt_c and extended with d as a random walk, which is the input-disturbance model of offset-free predictive control [21], runs at every execution whichever controller holds the feed. Its update is Equation (3.14).

$$\begin{aligned} \bar{x}_k &= a\,\hat{x}_{k-1} + b\left(C_{Af,k-1} - \bar{C}_{Af} + \hat{d}_{k-1}\right) \\ \varepsilon_k &= \hat{C}_{A,k} - \bar{C}_A - \bar{x}_k \\ \hat{x}_k &= \bar{x}_k + \ell_x\,\varepsilon_k, \qquad \hat{d}_k = \hat{d}_{k-1} + \ell_d\,\varepsilon_k \end{aligned} \tag{3.14}$$

In Equation (3.14) $a = 0.9801$ and $b = K_m(1-a) = 0.0199$, $C_{Af,k-1}$ is the feed applied over the period just ended, $\hat{C}_{A,k}$ is the reading of Equation (3.3) that the PID also receives, and $\ell_x = 0.166$ and $\ell_d = 0.913$ are the converged gains for a measurement variance $r = \sigma_n^2$, a state noise variance of $10^{-6}r$ and a disturbance noise variance $Q_d = r/K_m^2$. A load is booked in $\hat{d}$ as the feed deficit that would cause it, −0.30 mol/L when 1.80 mol/L of feed holds 1.50 mol/L against 20 m³/min of diluent, and the plan holds $\hat{d}$ over its horizon, so a constant load leaves no offset.

Each decision solves a quadratic program on Equation (3.13) sampled at T_s, over a horizon $T_p = 4$ min of $N = 100$ steps, about four model time constants. Twelve feed concentrations v_j, held over blocks of $1,1,1,1,2,2,3,4,6,9,15$ and 55 steps, minimize Equation (3.15) subject to $0 \le v_j \le C_{Af}^{\max}$.

$$J = \frac{T_s}{\tau_n}\sum_{k=1}^{N}\left(\frac{\tilde{C}_{A,k} - C_A^{\mathrm{SP}}}{y_s}\right)^2 + \lambda\sum_{j=1}^{12}\left(\frac{v_j - v_{j-1}}{u_s}\right)^2 \tag{3.15}$$

Here $\tilde{C}_{A,k}$ is the outlet predicted k steps ahead with $\hat{d}$ held, v_0 is the feed applied now, $\tau_n = 1$ min turns the first sum into a time integral, $y_s = 0.05$ mol/L is the settling band of the scorecard, $u_s = 2$ mol/L is the span of the feed and $\lambda = 30$ is a dimensionless move suppression, so the two weights are 16 and 7.5. Only the feed bound constrains the plan, as a hard limit. With no output band the program carries no slack, so an unreachable setpoint shows only as an active bound, and there is no rate limit, terminal weight or preview. Neither controller reads the diluent flow. The decisions are departures from $\bar{C}_{Af}$, a ridge of 10^{-9} relative to the weights keeps the program strictly convex, and a dual active-set quadratic programming code [22] returns its optimum. A rule common to the seven single-loop studio apps fixed λ and Q_d before any comparison, keeping, among the settings whose actuator travel on cases 3 and 4 stays within that of the shipped PID, the largest λ within 10 % of the lowest true-value IAE.

A plan is rejected when the solver reports no optimum or when its first move is not finite or falls outside the bound. The PID is the fallback of Section 1.6.3. It follows the applied feed at every execution as it follows the manual feed, so it takes the feed without a bump and keeps it, with the history marked, until the solver again returns an acceptable plan. Forced at 1.2 min of case 9 with the noise off, a rejection changes the feed by 0.006 mol/L, less than each of the next three PID moves of 0.007 to 0.008 mol/L, and the returning plan moves it by 0.069 mol/L. Because the first plan is

made while a case loads, the MPC is in charge from the first logged sample. It uses the analyzer reading, the feed it applied, the setpoint and the feed limits and nothing else, which keeps the plant of Equation (3.1), the instrument of Equation (3.3) and the model of Equation (3.13) apart, and the exported history gains its estimates and the outlet each plan predicts at the end of its horizon.

3.5.5 Moving Horizon Estimation

An estimator option decides the state from which each plan of the MPC starts. The default is the filter of Equation (3.14). The alternative is a moving horizon estimator (MHE), which at every execution refits the same augmented model to a sliding record of analyzer readings under box bounds on the model state, with the arrival cost of [23]. The filter keeps running and keeps being logged under either choice. At execution k the MHE chooses the augmented state at the start of its window and the noise sequence $w_{k-N_e},\dots,w_{k-1}$ that solve Equation (3.16).

$$\begin{aligned} \min\ & \|z_{k-N_e}-\bar{z}_{k-N_e}\|^2_{\bar{P}^{-1}}+\sum_{j=k-N_e}^{k-1} w_j^{\mathsf{T}} Q^{-1} w_j \\ & +\frac{1}{r}\sum_{j=k-N_e}^{k}\left(\hat{C}_{A,j}-\bar{C}_A-x_j\right)^2 \\ \text{s.t. } & z_{j+1}=\begin{bmatrix} a & b \\ 0 & 1 \end{bmatrix} z_j+\begin{bmatrix} b \\ 0 \end{bmatrix}\left(C_{Af,j}-\bar{C}_{Af}\right)+w_j \\ & 0\le \bar{C}_A+x_j\le C_{Af}^{\max} \end{aligned} \tag{3.16}$$

In Equation (3.16) $z_j=[x_j\ \ d_j]^{\mathsf{T}}$ stacks the state and the disturbance of Equation (3.13), a and b are the coefficients of Equation (3.14), $C_{Af,j}$ is the feed applied over the period that ends at execution $j+1$, w_j is the process noise with covariance $Q=\operatorname{diag}(10^{-6}r,\,Q_d)$, and $r=\sigma_n^2$, all in the units of Equation (3.14). The arrival term ties the start of the window to $\bar{z}_{k-N_e}$, the estimate the MHE made one execution earlier carried one step through the model, through the steady a-priori covariance $\bar{P}$ of the filter, which is held fixed. With $N_e=10$ the window spans 0.2 min and eleven readings, so each solve has 22 variables and 22 bound rows and goes to the dual active-set code that solves the plan.

The upper bound is the feed limit $C_{Af}^{\max}$, a constant of the source, because an outlet fed at most that strength and diluted only by a stream free of solute cannot exceed it, and a nonnegative feed keeps the outlet of Equation (3.1) at or above zero. No declared value limits d, so it is left free. Unless a bound is active, the window reproduces the recursive estimate of the filter, and the option can change a plan only where the filter would put the outlet outside $[0,C_{Af}^{\max}]$. A window that the solver does not bring to an optimum, or whose estimate is not finite or leaves the bounds, is replaced by the Kalman correction of the one-step prediction of the MHE, clipped to the bounds, and the exported history marks the replacement. The estimator reads the analyzer, the applied feed and declared constants and nothing else, so a parameter edited during a run reaches it only through the reading.

3.5.6 Relay Autotuning

The PID constants can also be taken from an experiment on the running loop, the relay feedback test of Åström and Hägglund [24], which Section 1.4.7 introduces. A button starts it from the present state whenever the PID holds the feed in automatic, and no case starts with it engaged. The relay sets the feed to $u_0-\delta_r$ when the reading exceeds the setpoint by more than ε_r and to $u_0+\delta_r$ when it falls short by more than ε_r. Here u_0 is the mean feed applied over the last 0.5 min, a tenth of the trend window, δ_r is 0.1 mol/L, five percent of the feed range, unless 0.9 of the distance from u_0 to the nearer bound is smaller, and $\varepsilon_r=3\sigma_{n,0}=0.03\,\text{mol/L}$ is three times the default analyzer noise $\sigma_{n,0}$, whatever noise the run carries. A cycle counts only if each of its halves lasts four executions or more, and the test converges once two successive counted cycles have periods, and swings $\pm\alpha$, within 5 % of each other. The describing function of the hysteretic relay then supplies the ultimate gain,

which the Tyreus–Luyben rule of Table 1.2 turns, together with the period P_u, into the PI settings of Equation (3.17), in which D is the fraction of each cycle spent at $u_0+\delta_r$, one half for a symmetric cycle. The first expression is Equation (1.16) with the factor $\sin(\pi D)$ added for a cycle that is not symmetric.

$$K_{cu}=\frac{4\delta_r\sin(\pi D)}{\pi\sqrt{\alpha^2-\varepsilon_r^2}},\qquad K_c=\frac{K_{cu}}{3.2},\qquad \tau_I=2.2\,P_u \tag{3.17}$$

Both gains are dimensionless, and P_u and τ_I are in min. The settings are rounded to three significant figures with $\tau_D=0$, each is held inside its input range, τ_I below the 20 min that switches integral action off, and they remain editable. A test that does not converge ends after 30 cycles or 10 min, and a second press, a change to manual or a change to the MPC cancels it. The PID follows the applied feed during the test as it does in manual, so it resumes without a bump. The result note gives $\psi=\arcsin(\varepsilon_r/\alpha)$, the phase lag that the band adds to the relay, so that the cycle settles where the loop phase is $-180^\circ+\psi$, and from $\psi\geq 10^\circ$ on it warns that the band rather than the tank has set the cycle.

3.6 Setpoint Reachability

3.6.1 The Steady-State Inverse and Its Range

Setting the derivative in Equation (3.1) to zero and solving for the feed concentration gives the exact steady-state inverse of the plant, as shown in Equation (3.18).

$$C_{Af}^{\star}=\frac{q+q_d}{q}\,C_A^{\mathrm{SP}} \tag{3.18}$$

Equation (3.18) is the value at which the stored output u_b+I of Equation (3.10) comes to rest, and it is the only model of the tank the PID ever holds. Holding a composition against dilution requires feed strength in direct proportion to the dilution ratio, and the expression contains no dynamic term. The application does not display the required value, but once a reachable case has settled the applied feed concentration equals it. The feed concentration is flagged whenever it sits on a bound, which turns the test of Equation (3.19) into a visible condition rather than an inference from a trend.

The manipulated variable is bounded, so Equation (3.18) defines a reachable set rather than a solution. A setpoint is attainable at steady state only when the required feed concentration falls inside the actuator range, which gives the three equivalent statements of Equation (3.19).

$$C_A^{\mathrm{SP}}\leq\frac{qC_{Af}^{\max}}{q+q_d},\quad q_d\leq q\left(\frac{C_{Af}^{\max}}{C_A^{\mathrm{SP}}}-1\right),\quad \frac{q_d}{q}\leq\frac{C_{Af}^{\max}}{C_A^{\mathrm{SP}}}-1 \tag{3.19}$$

With the default $q=100\,\mathrm{m^3/min}$, $C_{Af}^{\max}=2.0\,\mathrm{mol/L}$ and $C_A^{\mathrm{SP}}=1.5\,\mathrm{mol/L}$, the loop rejects any diluent flow up to 33.3 $\mathrm{m^3/min}$ and no more. Beyond that the highest outlet concentration the plant can produce is $qC_{Af}^{\max}/(q+q_d)$, and the standing error is the difference of Equation (3.20).

$$e_\infty=C_A^{\mathrm{SP}}-\frac{qC_{Af}^{\max}}{q+q_d} \tag{3.20}$$

At $q_d=50\,\mathrm{m^3/min}$ the ceiling is 1.333 mol/L and the standing error is 0.167 mol/L, which is 11.1 % of the setpoint. Neither expression contains a controller gain, a sample period or a tracking time.

🕮 **Pause.** Try Exercise 3.2 before reading on.

3.6.2 Open-Loop Behavior During Saturation

A pinned actuator opens the loop. With C_{Af} held at its bound, Equation (3.1) is the same first-order equation it is in manual mode, so the approach to the constrained steady state has the open-loop

residence time $\tau = V/(q+q_d)$ regardless of how the controller is tuned. The trajectory during saturation is a plant property. Fitting the pinned portion of the two constrained runs of Section 3.7 returns $\hat{\tau} = 0.6617\,\text{min}$ at $K_c = 2.0$ and $\hat{\tau} = 0.6617\,\text{min}$ at $K_c = 8.0$, against the 0.6667 min of $V/(q+q_d)$ and against the $\tau_p/(K_pK_c)$ of 0.50 and 0.125 min that the same two tunings would have imposed had the feed been free to move. The two fits agree to five decimals at 0.66165 min, which is the discrete time constant of the Euler step at $\tau = 0.6667\,\text{min}$, so the controller leaves no trace on the pinned trajectory.

> **TRY IT IN THE APP** **8 · Actuator Limit**
>
> Load the case and run it to 5 min. Against a standing dilution of 20 m^3/min the setpoint steps to 1.9 mol/L at 1 min. Watch the feed trend reach its 2.0 mol/L limit and stay there for 80 % of the run, while the outlet stops at 1.6653 mol/L. The predictive controller ends at the same outlet concentration.

3.7 Results

Each experiment starts from the reference case, configured by hand from the operating inputs of Table 3.2 before the run, so the run starts at the steady state those settings define, and every step or upset is applied 1 min into the run, the instant at which the shipped cases apply their first. The exported file (Section 1.6.4) records, every 0.1 min, the time, mode, case, setpoint, analyzer reading, noise-free outlet concentration, applied feed, limit flag and both disturbances. Times, peaks and integrated errors quoted below are evaluated at the 0.01 min step of the same runs, which the exported rows reproduce to within their spacing and 0.007 mol min/L. Table 3.5 lists the sequence.

The application also ships ten cases, listed with their completion requirements in Table 3.4, and each hand-configured experiment of Table 3.5 starts from the first of them with its settings changed before the run. Every case starts at steady operation, and each case with a scripted step or upset applies the first one 1 min into the run. Completion is judged as Section 1.6.4 describes. The challenge is judged on the case as given, so changing a model parameter or a disturbance during the attempt voids it. A setting changed before the run gives the same run as the same setting carried through Reset, while reloading the case restores its shipped settings.

3.7.1 Startup From an Empty Tank

Every case starts at steady operation, so a startup is built from the reference case by setting the manual feed to zero before the run, which starts the tank empty at the steady state of zero feed, with the default $K_c = 2.0$, $\tau_I = 1.0\,\text{min}$, $\tau_D = 0$ and the noise switched off. How the loop is then closed decides the approach. Switching to automatic 1 min into the run with the 1.5 mol/L setpoint already in force is a bumpless transfer onto a standing error, so the feed starts from zero and is raised by integral action alone. It never reaches the bound, the outlet comes within 5, 2 and 1 % of setpoint 3.63, 4.54 and 5.23 min after the switch with no overshoot, and the feed comes to rest at exactly 1.5000 mol/L. Setting that feed by hand instead reaches the same bands in 2.98, 3.89 and 4.58 min, so this transfer is slower than no feedback at all. Holding the setpoint at zero until the switch and then raising it to 1.5 mol/L applies the proportional kick of Equation (3.7) instead. The demand of $2.0 \times 1.50 = 3.00\,\text{mol/L}$ puts the feed on its upper bound at the first execution, where it stays for 0.70 min while the stored output relaxes toward the bound as Equation (3.12) describes and reaches 1.03 mol/L, and the outlet comes within 5, 2 and 1 % in 1.61, 2.05 and 2.38 min, passing the setpoint by less than 0.0001 mol/L. Even that time exceeds the four closed-loop time constants that $\tau_c = 0.5\,\text{min}$ would suggest, because the first part of the approach is made at the bound at the open-loop rate. A nominal startup can touch the constraint, and the way automatic is engaged rather than the tuning decides how fast it goes.

Table 3.4 Scenarios and Completion Requirements

	Scenario	Kind	Required to complete
1	Reference operating point	Experiment	Hold the tank in manual at steady operation, outlet and feed at 1.50 mol/L, for 1.3 min and note which quantities stay stationary.
2	Open-loop actuator step	Experiment	In manual from steady operation, run through the feed step from 1.50 to 1.90 mol/L at 1 min and on to 3.5 min, reading the direction, size and speed of the outlet response.
3	Closed-loop setpoint step	Experiment	In automatic from steady operation, run through the setpoint step from 1.5 to 1.7 mol/L at 1 min until the outlet stays within ±0.05 mol/L of the new setpoint for 0.25 min.
4	Disturbance rejection	Experiment	In automatic, run through the dilution step from 0 to 20 m^3/min at 1 min and on to 3.5 min.
5	Recover by hand	Exploration	After the same dilution step in manual, move the feed by hand to a setting that holds the outlet, once settled, within half of its largest error since the step (or within ±0.05 mol/L), and keep the outlet there for 0.25 min in manual.
6	Conservative tuning	Experiment	With $K_c = 0.8$, run through the setpoint step from 1.5 to 1.66 mol/L at 1 min and on to 3.5 min.
7	Aggressive tuning	Experiment	With $K_c = 4.0$, run through the same setpoint step and on to 3.5 min.
8	Actuator limit	Demonstration	Against a standing dilution of 20 m^3/min, with the feed at 1.80 mol/L, run through the setpoint step from 1.5 to 1.9 mol/L at 1 min until the feed reaches its 2.0 mol/L limit and on to 1.7 min, the outlet then able to reach only 1.67 mol/L.
9	MPC at the edge of the feed	Experiment	Under the MPC from $t = 0$, run 5 min through the dilution step from 0 to 20 m^3/min at 1 min, the setpoint step to 1.70 mol/L at 2 min and its return to 1.60 mol/L at 3.5 min, then run the case again from $t = 0$ under the PID and compare the two runs.
10	Challenge	Challenge	Starting at steady operation from $K_c = 0.4$ and $\tau_I = 1.0$ min, with the dilution stepping from 0 to 20 m^3/min at 1 min, hold the analyzer reading within ±0.12 mol/L of 1.5 mol/L from 2.64 to 4.0 min, with the model parameters and disturbances as given.[a]

[a] With the shipped $K_c = 0.4$ the worst reading error over the scored window is 0.159 mol/L, 1.33 times the band, and the outlet is still 0.114 mol/L low at 4.0 min. With the suggested $K_c = 2.0$ and $\tau_I = 1.0$ min, revealed before the run or carried through Reset, which give one run, it is 0.053 mol/L, inside the band with 0.067 mol/L to spare. Most of it is the end of the recovery rather than analyzer noise, because the true concentration is still up to 0.048 mol/L below setpoint over the same window.

Table 3.5 Experiment Sequence and the Mechanism Each One Exposes

Experiment	Setup	What it exposes
Step test	manual, C_{Af} 1.0 to 1.5 at four flow settings	$K_p = q/(q+q_d)$ and $\tau = V/(q+q_d)$
Startup	empty tank, automatic engaged on the error or by a setpoint step	how engaging the loop sets the approach
Reachable upset	q_d 0 to 25 m^3/min	integral action removes the offset, feed rests at 1.875
Boundary	q_d = 33 then 34 m^3/min	Equation (3.19) closes between two adjacent settings
Unreachable upset	q_d 0 to 50 m^3/min at K_c = 2 and 8	the same offset at two tunings
Recovery	q_d back to 0	off the bound as the outlet crosses setpoint
Proportional only	integral off, q_d = 33 and 50 m^3/min	an offset a controller does remove
Analyzer noise	$\sigma_n = 0.01$ mol/L, K_c = 2 and 8	feed activity proportional to K_c
Edits in automatic	C_A^{SP} 1.50 to 1.45, K_c 2.0 to 2.5	increments act, nothing is reseeded

3.7.2 Reachable and Unreachable Dilution

Table 3.6 reports four dilution upsets applied to a settled loop at 1.5 mol/L, with the analyzer noise switched off so that the offsets reported are the deterministic ones. Each column is the 10 min

Table 3.6 Dilution Upsets Applied to a Settled Loop at $C_A^{SP} = 1.5\,\text{mol/L}$ With $\tau_D = 0$ and $\sigma_n = 0$, Measured Over the 10 min Following the Step

q_d (m³/min)	25	33	34	50
K_c, τ_I (min)	2, 1	2, 1	2, 1	2, 1
$C_{Af}^{\star}$ required (mol/L)	1.875	1.995	2.010	2.250
Reachable by Eq. (3.19)	yes	yes	no	no
Final C_{Af} (mol/L)	1.8750	1.9949	2.0000	2.0000
Final C_A (mol/L)	1.4999	1.4999	1.4925	1.3333
Offset (mol/L)	0.0001	0.0001	0.0075	0.1667
Predicted offset, Eq. (3.20)	—	—	0.0075	0.1667
Maximum $\lvert e \rvert$ (mol/L)	0.0898	0.1165	0.1198	0.1703
IAE (mol min/L)	0.187	0.247	0.288	1.643
Time on the bound (min)	0.00	0.00	6.48	9.34

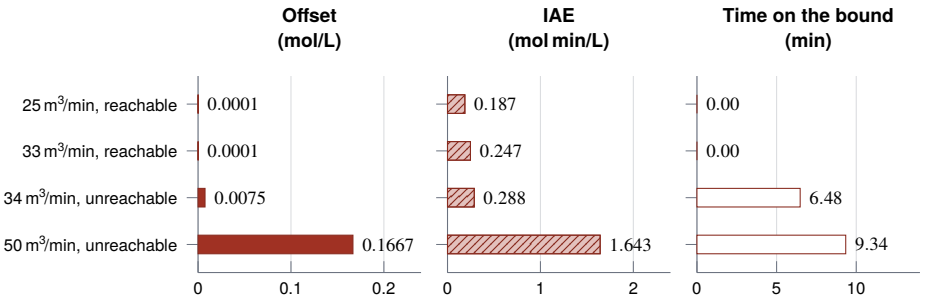


Figure 3.2 The dilution upsets of Table 3.6 under one tuning: the offset left after ten minutes (solid bars), the integrated absolute error (hatched bars) and the time the feed spends on its bound (open bars). Notice that one more m³/min of diluent, from 33 to 34, puts the feed on its bound for most of the window and leaves an offset, and that the offset at 50 is the bound's, not the controller's.

following a step in the diluent flow. The disturbance rejection case ships a reachable upset of 20 m³/min, shown in Figure 3.3.

> **TRY IT IN THE APP** **4 · Disturbance Rejection**
>
> Load the case and press Run. Watch the outlet trend when the diluent opens at 1 min. The outlet falls to 1.428 mol/L at 1.66 min and stays within ±0.05 mol/L of the setpoint after 2.55 min, while the feed trend climbs toward the 1.80 mol/L that the new dilution requires without reaching its bound.

The reachable set closes between two adjacent values of a diluent flow whose resolution is 1 m³/min. At $q_d = 33\,\text{m}^3/\text{min}$ the required feed concentration is 1.995 mol/L, the feed comes to rest at 1.9949 mol/L and the outlet returns to 1.4999 mol/L with an integrated absolute error of 0.247 mol min/L. At 34 m³/min the requirement is 2.010 mol/L, the feed spends 6.48 min of the ten on its bound and the outlet settles 0.0075 mol/L low, matching Equation (3.20) to four decimals. The loop has not changed and its tuning has not changed. The inequality of Equation (3.19) has simply stopped being satisfied. Figure 3.2 draws the table as bars.

Table 3.7 repeats the largest upset at two tunings that differ by a factor of four in both gains, with τ_I held at 1.0 min so that K_I rises with K_c.

Quadrupling both gains changes the final outlet concentration by nothing at all. Both tunings settle at 1.3333 mol/L against a 1.5 mol/L setpoint, both leave an offset of exactly 0.1667 mol/L, and the integrated absolute error over the ten minutes improves only from 1.643 to 1.578 mol min/L, which is

Table 3.7 The Same $q_d = 50\,\mathrm{m}^3/\mathrm{min}$ Upset and Its Removal at Two Tunings, $\tau_I = 1.0\,\mathrm{min}$, $\sigma_n = 0$. The Upset Window Is 10 min and the Recovery Window Is 8 min.

	$K_c = 2.0$	$K_c = 8.0$
Diluent opened to 50 m³/min		
Final C_A (mol/L)	1.3333	1.3333
Offset (mol/L)	0.1667	0.1667
Maximum $\lvert e \rvert$ (mol/L)	0.1703	0.1667
IAE (mol min/L)	1.643	1.578
Time on the bound (min)	9.34	9.86
Fitted $\hat{\tau}$ while pinned (min)	0.6617	0.6617
Diluent closed again		
Delay off the bound (min)	0.30	0.30
Peak C_A (mol/L)	1.6266	1.5476
Overshoot (%)	8.44	3.17
Return within 1 % (min)	3.72	1.83
IAE (mol min/L)	0.272	0.084

3.9 % for four times the gain. The benefit of the additional gain is concentrated in the initial transient and in the recovery. The aggressive tuning reaches the bound sooner and spends 9.86 min of the ten there against 9.34 min, and after the diluent valve is closed it returns to within 1 % of setpoint in 1.83 min against 3.72 min, with 3.17 % overshoot against the 8.44 % of the default tuning. Both tunings hold the feed on the bound for 0.30 min after the dilution closes, until the outlet crosses the setpoint, because Equation (3.12) has left both stored outputs at the bound. The recovery overshoot is therefore not windup. The reachable upset at 33 m³/min, which never touches the bound but leaves the stored output at 1.995 mol/L, gives a recovery peak of 1.6253 mol/L against the 1.6266 mol/L that follows the unreachable one.

🕮 **Pause.** Try Exercise 3.3 before reading on.

3.7.3 Two Offsets of Similar Size

A second standing error, close in size to the first, arises by a mechanism a controller does remove. A proportional loop whose stored output $u_b + I$ is frozen at b has the steady state of Equation (3.21) while its demand stays inside the range. Equation (1.3) is the same balance for a setpoint step taken from a bias that held the old setpoint.

$$C_A^{\infty} = \frac{K_p\,(b + K_c\,C_A^{\mathrm{SP}})}{1 + K_p\,K_c}, \qquad e_{\infty} = \frac{C_A^{\mathrm{SP}} - K_p\,b}{1 + K_p\,K_c} \tag{3.21}$$

Setting the integral time to its 20 min top before the run switches integral action off, and a run started in automatic at the 1.5 mol/L setpoint then has $b = u_b = 1.50\,\mathrm{mol/L}$, the value of Equation (3.18) with the diluent shut, so Equation (3.21) predicts no offset at all until a load arrives, which is the purpose of the bias in a positional law. With $K_c = 2.0$ and the noise switched off, opening the diluent to 33 m³/min gives $K_p = 0.752$, and Equation (3.21) predicts 1.3514 mol/L with the feed at 1.797 mol/L. The run settles there to four decimals, an offset of 0.149 mol/L that the same upset with $\tau_I = 1.0\,\mathrm{min}$ removes entirely (Table 3.6). At 50 m³/min the proportional loop stands at 1.2857 mol/L, an offset of 0.214 mol/L, with the feed at 1.929 mol/L and still inside its range. Restoring $\tau_I = 1.0\,\mathrm{min}$ 5 min after the step puts the feed on its bound 0.20 min later, and the outlet rises only to the 1.3333 mol/L of Equation (3.20). Quadrupling the gain instead lowers the offset that Equation (3.21) predicts to 0.079 mol/L, but its 2.13 mol/L demand exceeds the bound, and the outlet again stops at 1.3333 mol/L.

The two experiments therefore report offsets of similar size against the same 2.0 mol/L feed ceiling, and they are not the same phenomenon. The offset of Equation (3.21) is the finite loop gain



Figure 3.3 A reachable dilution upset rejected by feedback, in the disturbance rejection case as shipped, with the default $K_c = 2.0$, $\tau_I = 1.0\,\text{min}$, $\tau_D = 0$ and the default 0.01 mol/L analyzer noise, at $t = 5.00\,\text{min}$. The loop starts in automatic at steady operation, outlet and feed at 1.50 mol/L, and the diluent stream opens to 20 m^3/min at $t = 1.0\,\text{min}$. Equation (3.18) then requires 1.80 mol/L of feed, inside the range, so the feed never reaches its bound (its peak is 1.87 mol/L). The outlet falls to 1.428 mol/L at 1.66 min and stays within ±0.05 mol/L of setpoint after 2.55 min. At the instant shown the outlet is 1.494 mol/L, the analyzer reads 1.504 mol/L, the feed is 1.80 mol/L and the scorecard reads an IAE of 0.150 mol min/L, 1.17 mol/L of feed travel and no saturation. The residence time is 0.83 min because the throughput has risen to 120 m^3/min. The screenshot is cropped to the plant view, with the scorecard tiles moved into a frame at its right, above the trends. Callouts: (1) the outlet readout, (2) the analyzer, which the app labels CT, (3) the scorecard, (4) the low point of the outlet trend after the dilution step at the dotted marker, and (5) the feed trend climbing to carry the new dilution.

of a proportional controller and is removed by integral action or reduced by more gain, each of which the application demonstrates within minutes, until the feed reaches its bound. The offset of Equation (3.20) contains no controller parameter and is removed only by more feed strength, less dilution or a lower setpoint. At 50 m^3/min the two meet, and integral action removes the 0.048 mol/L by which the proportional offset exceeds the reachability limit and nothing more.

Table 3.8 Feed Activity of a Settled Loop With $\sigma_n = 0.01\,\text{mol/L}$, $q_d = 0$ and $\tau_D = 0$

K_c, τ_I (min)	2.0, 1.0	8.0, 1.0
σ of $\hat{C}_A$ (mol/L)	0.01068	0.01103
σ of C_A (mol/L)	0.00121	0.00297
σ of C_{Af} measured (mol/L)	0.02119	0.08739
σ of C_{Af} from $K_c\sigma_n$ (mol/L)	0.02000	0.08000
Feed travel (mol/L per min)	1.22	5.14

DID YOU KNOW? More Gain Buys No Feed Strength

From the simulator: against 50 m^3/min of diluent, a gain of 8.0 leaves the outlet at 1.3333 mol/L, exactly where a gain of 2.0 leaves it, and trims the integrated absolute error only from 1.643 to 1.578 mol min/L (Table 3.7).

3.7.4 Analyzer Noise, With and Without the Bound

Summing Equation (3.7) over a run telescopes the proportional increments, so with $\tau_D = 0$ the instantaneous noise contribution to the feed concentration is $-K_c n_k$ with n_k the noise sample at step k, and the standing feed noise is $\sigma_u = K_c\sigma_n$ rather than a random walk. Table 3.8 reports a loop started in automatic at 1.5 mol/L with the default 0.01 mol/L analyzer noise over the first 6 min of the run with no disturbance applied, sampled at every controller execution, because the exported rows see one execution in five and record only about a quarter of the feed travel.

The telescoped estimate falls 5.6 % below the measured feed noise at $K_c = 2.0$ and 8.5 % below it at $K_c = 8.0$. Part of that deficit belongs to the noise itself, because the 300 draws in the window have a standard deviation of 0.01054 mol/L, and with that value the estimate falls only 0.5 and 3.5 % below. The remainder underpredicts because the loop has started to move the tank in response to its own noise. The evidence is in the second row, where the true concentration standard deviation rises from 0.00121 to 0.00297 mol/L while the noise draws are identical. Quadrupling the gain therefore multiplies the feed travel by 4.22 and the concentration variability the loop is meant to suppress by 2.46 (2.45 to 2.70 over three 6 min windows of the same run), which is the ordinary argument against aggressive tuning on a measurement-limited loop.

Noise does not move the saturated offset. Repeating the $q_d = 50\,\text{m}^3/\text{min}$ upset with the analyzer noise left at its default puts the feed exactly on 2.000 mol/L at every controller execution from 0.70 min after the step (93.2 % of the 10 min window, the rest being the approach), and the mean outlet concentration over the last 6 min is 1.33333 mol/L, the noise-free value. The stored output sits at the bound by Equation (3.12), and over those 6 min the proportional term holds the demand at least 0.27 mol/L above it, so no noise sample of the default size brings the demand back inside the range and nothing is clipped on one side only.

3.7.5 Edits in Automatic

Editing the setpoint or a gain while the loop is in automatic acts through the increments of Equations (3.7) to (3.9) alone, as described in Section 3.5, and nothing is reseeded. Lowering the setpoint from 1.50 to 1.45 mol/L on a settled loop with the noise switched off lowers the feed concentration from 1.5000 to 1.4000 mol/L at the next execution, the $K_c\,\Delta C_A^{\text{SP}}$ that Equation (3.7) gives, and the outlet then moves to the new setpoint without passing it, coming within 0.005 mol/L of it in 1.12 min. Raising K_c from 2.0 to 2.5 with the setpoint untouched leaves the feed at 1.5000 mol/L and the outlet where it was, because the error is zero when the gain changes. Two tunings can therefore be compared within one run without an excursion caused by the edit itself.

Table 3.9 Five Cases Under the PID and Under the Predictive Controller, Each Pair Run With Identical Analyzer Noise

Case	Controller	IAE (mol min/L)	IAE_C	e_{max} (mol/L)	Travel (mol/L)	Limit (%)	Bound
3, setpoint	PID	0.111	0.089	0.162	1.44	0	—
	MPC	0.082	0.053	0.163	1.30	4	8
4, dilution	PID	0.150	0.143	0.071	1.17	0	—
	MPC	0.061	0.032	0.048	0.91	0	0
8, limit	PID	1.066	1.062	0.379	0.33	80	—
	MPC	1.068	1.063	0.381	0.25	80	100
9, edge	PID	0.281	0.272	0.240	1.20	30	—
	MPC	0.207	0.197	0.184	0.99	30	38
10, challenge	PID, $K_c = 0.4$	0.378	0.372	0.149	0.24	0	—
	PID, $K_c = 2^a$	0.138	0.133	0.071	0.94	0	—
	MPC	0.053	0.030	0.048	0.79	0	0

IAE uses the analyzer reading, as the scorecard does, and IAE_C the true concentration. e_{max} is the largest true deviation from setpoint after the first event, Travel is the summed absolute change of the feed, Limit is the fraction of the run spent with the feed at a bound, and Bound counts the plans, of 125 (100 in the challenge), with the feed bound active. [a]Suggested, $\tau_I = 1.0$ min.

3.7.6 Predictive Control on the Shipped Cases

Table 3.9 runs cases 3, 4, 8 and 9 for 5 min and the challenge for its 4 min under each controller. The controller is selected before each run starts, so both runs of a case meet the same events and the same analyzer noise draws, and the noise columns of the two runs of case 9 agree to the last digit. The PID keeps the shipped $K_c = 2.0$, $\tau_I = 1.0$ min and $\tau_D = 0$ in cases 3, 4, 8 and 9, the challenge ships $K_c = 0.4$ and suggests $K_c = 2.0$ with $\tau_I = 1.0$ min, while the MPC runs every case with the settings of Section 3.5.4. Unlike the preceding subsections, the numbers here are read on the 0.1 min log, the samples from which the scorecard of the application computes them.

At the setpoint step of case 3 the model reproduces the plant apart from the noise, and the MPC uses the bound. Its first eight plans after the step have the upper bound active, the feed rests on 2.00 mol/L for 0.20 min, and the outlet stays within ±0.01 mol/L of 1.70 mol/L from 0.60 min after the step, where the PID needs 1.60 min. The MPC passes the new setpoint by 0.005 mol/L and the PID by 0.001 mol/L, and the true-value IAE is 0.053 against 0.089 mol min/L with 9 % less feed travel.

The dilution of case 4 activates the bound in none of the 125 plans, so the whole difference there belongs to the model and its estimate. The estimate $\hat{d}$ reaches 90 % of the −0.30 mol/L deficit within 0.5 min of the upset and ends the case at −0.297 mol/L. The true outlet falls to 1.452 mol/L under the MPC and to 1.429 mol/L under the PID, and it returns for good within ±0.02 mol/L of setpoint 0.70 min after the upset against 2.70 min. The scorecard IAE falls from 0.150, the value of Figure 3.3, to 0.061 mol min/L, and the feed travel from 1.17 to 0.91 mol/L. A PID at $K_c = 8.0$, the aggressive tuning of Table 3.7 and not a shipped one, reaches a similar 0.058 only with 4.53 mol/L of travel, five times that of the MPC.

Case 8 is a tie. Both controllers reach the bound at the execution at 1.00 min and hold it for 80 % of the run, both leave the outlet at 1.6653 mol/L at 5 min, and their scorecard IAEs, 1.066 and 1.068 mol min/L, differ by 0.1 %. All 100 plans after the step have the bound active, $\hat{d}$ ends at −0.331 mol/L against the −0.333 mol/L that the feed at its bound implies at the 1.667 mol/L ceiling of Equation (3.19), and the last plan predicts 1.671 mol/L at the end of its horizon. The MPC therefore plans with the setpoint out of reach and gains nothing from doing so. Its lower travel accrues in the quiet minute before the step, 0.08 against 0.21 mol/L.

Case 9 joins the dilution of case 4 to a setpoint of 1.70 mol/L at 2 min, which Equation (3.18) prices at 2.04 mol/L of feed, and to a reachable 1.60 mol/L at 3.5 min, which needs 1.92 mol/L, as Figure 3.4 shows. Before the setpoint step the MPC rejects the dilution as in case 4, with a true-value IAE over the first 2 min of 0.024 against 0.061 mol min/L, and meets the step with the outlet at 1.497 mol/L, where the PID has 1.434 mol/L. Both feeds reach the bound at the execution at 2.00 min

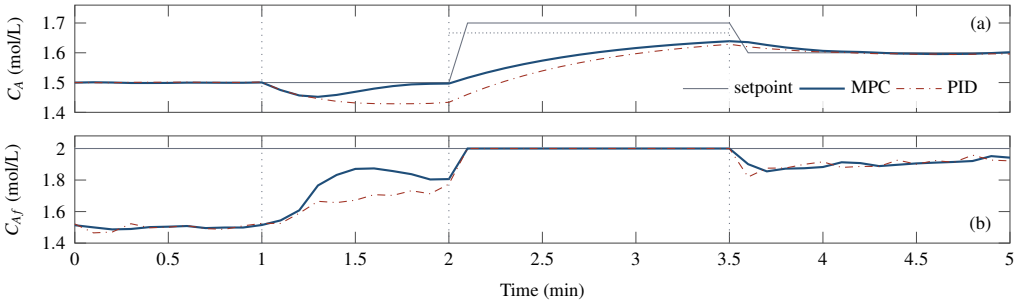


Figure 3.4 The two controllers on case 9, with identical analyzer noise draws, from the 0.1 min log. The MPC is the thick solid line and the PID the dash-dotted line. The diluent opens to 20 m^3/min at 1 min, the setpoint steps to the unreachable 1.70 mol/L at 2 min and returns to 1.60 mol/L at 3.5 min (vertical dotted lines). (a) True outlet concentration and setpoint (thin gray line), with the 1.667 mol/L ceiling of Equation (3.19) dotted between 2 and 3.5 min. (b) Feed concentration and its 2.0 mol/L bound (thin gray line).

and stay on it until the setpoint returns, so for 1.5 min the tank runs open loop under either controller, and the 0.063 mol/L gap present at 2 min shrinks by the Euler factor $1 - \Delta t\,(q + q_d)/V = 0.988$ per step. That decay, summed over the logged samples, is 0.0411 mol min/L, which is all of the difference between the true-value IAEs of the two runs over the interval, 0.161 and 0.202 mol min/L. Every one of the 38 plans in the interval has the bound active, and at 3.0 min the plan already predicts 1.658 mol/L at the end of its horizon, close to the ceiling, yet the MPC gains nothing while the feed is pinned.

When the setpoint returns, the PID leaves the bound at the execution at 3.50 min and the MPC at its next decision, 0.02 min later. The stored output $u_b + I$ of the PID is then 1.916 mol/L, below the bound, because Equation (3.12) had been drawing it up from 1.624 mol/L after the execution at 2.00 min and predicts the measured value to the fifth decimal. No integral demand above the bound remains to be worked off, so the windup that a constraint inside the plan is meant to prevent is already absent from the PID. The PID drops the feed to 1.886 mol/L at that execution, while the move suppression $\lambda = 30$ makes the MPC lower it by 0.069, 0.030 and 0.026 mol/L over its first three decisions. The PID is within ±0.02 mol/L of 1.60 mol/L for good 0.20 min after the return and the MPC after 0.30 min, and over the last 1.5 min the MPC loses, 0.012 against 0.009 mol min/L of true-value IAE. Over the whole run it wins, 0.207 against 0.281 mol min/L on the scorecard, and the whole margin traces to the rejection of the dilution before the bound was reached.

The MPC reads none of the PID settings, so its challenge run repeats the first 4 min of its case 4 run. Its worst reading error over the judged window from 2.64 to 4.0 min is 0.018 mol/L, against 0.159 mol/L for the shipped $K_c = 0.4$, which fails the ±0.12 mol/L band, and 0.053 mol/L for the suggested tuning, which passes.

All 600 plans were optimal, so no fallback occurred, and plans with the bound active needed up to 12 active-set iterations. The mean logged solve time was 0.05 to 0.30 ms per case in four Node.js runs and 237 and 253 µs in two Chrome runs of case 9 (largest 2.1 ms), 0.6 % of one processor core at one process minute per second and at most 6.3 % at the 10× top speed, wall-clock values from one desktop computer that vary between runs. Across the five cases the MPC wins the setpoint step, the dilution and the challenge, ties case 8, and in case 9 loses only the 1.5 min after the setpoint returns, so at the feed bound its advantage over this PID is its model of the load rather than its handling of the bound.

Table 3.10 Cases 3 and 4 Over 15 min Under the Shipped PI, Two Counterfactual PI Settings, the Relay-Tuned PI and the MPC, With One Noise Seed

Case	Controller	IAE (mol min/L)	IAE_C	$e_{\max}$ (mol/L)	Travel (mol/L)	Travel^0
3, setpoint	PI, shipped	0.194	0.108	—	3.86	0.54
	PI^a	0.205	0.123	—	3.33	0.45
	PI^b	0.243	0.183	—	3.86	0.54
	PI, relay	0.267	0.209	—	3.33	0.46
	MPC	0.167	0.088	—	2.60	0.82
4, dilution	PI, shipped	0.233	0.159	0.072	3.59	0.30
	PI^a	0.253	0.181	0.078	3.10	0.30
	PI^b	0.325	0.272	0.079	3.59	0.30
	PI, relay	0.363	0.313	0.086	3.10	0.30
	MPC	0.145	0.058	0.048	2.19	0.50

PI, shipped: $K_c = 2.0$, $\tau_I = 1.0$ min. PI^a: the written $K_c = 1.73$ with the shipped τ_I. PI^b: the shipped K_c with the written $\tau_I = 1.83$ min. PI^a and PI^b are counterfactuals that the application neither ships nor writes. PI, relay: $K_c = 1.73$, $\tau_I = 1.83$ min. Columns as in Table 3.9, with IAE_C on the 0.01 min step and $e_{\max}$ after the dilution step. Travel^0 repeats Travel with $\sigma_n = 0$. Either estimator gives the MPC rows.

3.7.7 Estimator Choice and Relay Autotuning

The runs of this subsection last 15 min, three trend windows, with the challenge at its 4 min, the shipped noise seed and every setting made before the run, so their integrals are larger than those of Table 3.9. In a run of case 9 under the MPC, the outlet estimates of the MHE and of the filter differ by no more than $2.6 \times 10^{-10}\sigma_n$ over 749 executions, and the disturbance estimates by 4.1×10^{-11} mol/L. This residue does not come from rounding. The Riccati iteration that produced ℓ_x and ℓ_d stopped after 111 steps at a relative distance of 2.6×10^{-10} from its fixed point, the window repeats the covariance update that the fixed gain of the filter leaves out, and with the fixed-point covariance the two estimates agree to $1.4 \times 10^{-14}\sigma_n$. Run under the MPC with one estimator and then the other, the ten cases give feed records within 2.1×10^{-11} mol/L of each other, with no bound active, no window replaced and no plan rejected. The true outlet stays between 1.25 and 1.90 mol/L in every case and the filter never places it outside the bounds, so no case separates the two estimators and no trend of them is drawn.

Read against the true concentration by the measuring script, and by nothing the controller sees, the two estimates have the same error to the digits shown, a root-mean-square value of 0.0038 to 0.0047 mol/L across the ten cases, under half of σ_n, and a largest value of 0.013 to 0.019 mol/L. Under Node.js an estimate took a median of 37 to 43 µs and a 99th percentile of 160 to 222 µs in two runs. On the animation loop of the application in Chrome, during case 9, the estimator and the planner together used at most 1.3 % of one core at normal speed and 6.0 % at the tenfold speed, the higher of two runs at a load average of 2.1 to 2.8, against a budget of one tenth of a core. A window of 15 executions, the first choice, needs a median of 70 µs per estimate against 38 µs for 10 in the same Node.js session, a ratio of 1.84, and the same ratio applied to the share of the estimates in Chrome, 3.7 of the 6.0 percent, places it at about 9 % of a core at the tenfold speed. That margin was judged too narrow under load, so the window is 10, and the feed records of the two windows differ by 5.2×10^{-12} mol/L on case 9.

For the relay, case 3 was started without its setpoint step, so the loop held 1.50 mol/L with the feed at 1.50 mol/L, and the button was pressed at 0.5 min. With $\delta_r = 0.1$ mol/L the test converged after two cycles and 1.49 min, with $\alpha = 0.0378$ mol/L, $D = 0.49$, $P_u = 0.830$ min and $K_{cu} = 5.55$, and it wrote $K_c = 1.73$ and $\tau_I = 1.83$ min over the shipped 2.0 and 1.0 min. The ratio ε_r/α is 0.794, so ψ is 53° and the note flags the result. Under ten other noise seeds, 1001 to 1010, every test converged, with K_{cu} between 3.3 and 11.5, P_u between 0.558 and 1.09 min and ψ between 38° and 70°.

The tank has no ultimate point anywhere near these values. Its model, Equation (3.13), is a lag without dead time, and with a zero-order hold at Δt_c its phase first reaches −180° at the Nyquist frequency, where $K_{cu} = (1 + a)/b = 99.5$ and $P_u = 2\Delta t_c = 0.04$ min. The same relay run on that sampled model without noise cycles with $K_{cu} = 37.0$ and $P_u = 1.24$ min at $\varepsilon_r/\alpha = 0.993$, a cycle

placed by the band and the sampling. The analyzer noise widens the swing, and Equation (3.17) amplifies that error, since $\mathrm{d}\ln K_{cu}/\mathrm{d}\ln\alpha = -1/[1-(\varepsilon_r/\alpha)^2]$ is -2.71 at $\varepsilon_r/\alpha = 0.794$. The measured K_{cu} is therefore 0.15 of the noise-free relay value and 0.056 of the sampled ultimate gain.

The relay settings lose to the shipped ones in both cases of Table 3.10. The true-value IAE rises from 0.108 to 0.209 mol min/L on the setpoint step and from 0.159 to 0.313 mol min/L on the dilution step, where the largest deviation grows from 0.072 to 0.086 mol/L. The integral time carries 74 percent of each loss, because the shipped K_c with the written τ_I already reaches 0.183 and 0.272 mol min/L, while the written K_c with the shipped τ_I costs only 0.123 and 0.181 mol min/L. Equation (3.5) explains the direction. For a lag without dead time, internal model control sets τ_I to the residence time, 1.00 min with the diluent shut, whereas $2.2P_u$ gives 1.83 min and lowers K_I from 2.0 to 0.945/min, although the 20 m^3/min of case 4 calls for 2.4/min. The written settings move the feed 14 percent less, 3.33 against 3.86 and 3.10 against 3.59 mol/L, which is the ratio $1.73/2.0$ of the gains, because most of the travel is analyzer noise passed through K_c, as Table 3.8 showed for a settled loop. With $\sigma_n = 0$ the travel falls to 0.54 and 0.46 mol/L on case 3 and to 0.30 mol/L under both settings on case 4. The MPC, with either estimator, reaches 0.088 and 0.058 mol min/L on the same runs.

> **DID YOU KNOW? The Autotuner Loses to the Model Rule**
>
> From the simulator: the relay test writes an integral time of 1.83 min against the 1.0 min residence time of the undiluted tank, and over 15 min of the dilution step of case 4 the true-value integrated error rises from 0.159 to 0.313 mol min/L (Table 3.10). The tank has no ultimate point near the cycle the relay finds.

A relay started away from the reference point shows how strongly the swing sets the result. Once the setpoint step of case 3 has settled at 1.70 mol/L, a relay started at 5 min converges after four cycles and 2.41 min on $K_{cu} = 4.87$ and $P_u = 0.960$ min with $\alpha = 0.0398$ mol/L and $\psi = 49°$, and it writes $K_c = 1.52$ and $\tau_I = 2.11$ min. Run from the shipped settings with $\sigma_n = 0$, the same test needs two cycles, returns $K_{cu} = 37.6$ at $P_u = 1.24$ min with $\alpha = 0.0302$ mol/L and $\psi = 84°$, and writes $\tau_I = 2.73$ min with K_c held at the top of its range, 10, in place of the 11.7 the rule gives. Swings that differ by a factor of 1.32 yield ultimate gains that differ by a factor of 7.7, because both swings lie close to the band ε_r under the square root of Equation (3.17), and both gains stay far below the 99.5 of the sampled model.

3.8 Technology in Practice

The loop of Section 3.3 reads an ideal analyzer and sets a feed concentration directly. Figure 3.5 draws it with the devices a real blend would use in the place of the schematic's symbols.

What Measures It

Analyzer AT of Equation (3.3) adds white noise of 0.01 mol/L and nothing else. A real composition measurement brings what Section 3.3.4 leaves out: a sample line, an analysis cycle and calibration error. Continuous drug lines, for example, have relied heavily on near-infrared spectroscopy to monitor blend composition [25]. The range, accuracy and speed of such instruments depend on the principle and the sample system, and no single figure applies.

> **TECHNOLOGY Sensor: The Process Analyzer and Its Sample System**
>
> A composition analyzer either measures inside the process stream, in situ, or receives a sample carried to it through tubes, filters, pumps and regulators. Problems with process analyzers often stem from sample systems that are poorly built or poorly maintained. If debris plugs the sample line, the analyzer goes on measuring old, stale sample with no way of knowing it, and reports values that no longer follow the process. Some analyzers are slow by principle. An on-line chromatograph separates species in time, so cycle times of minutes are common, and enough of that dead time makes a feedback loop oscillate [15].

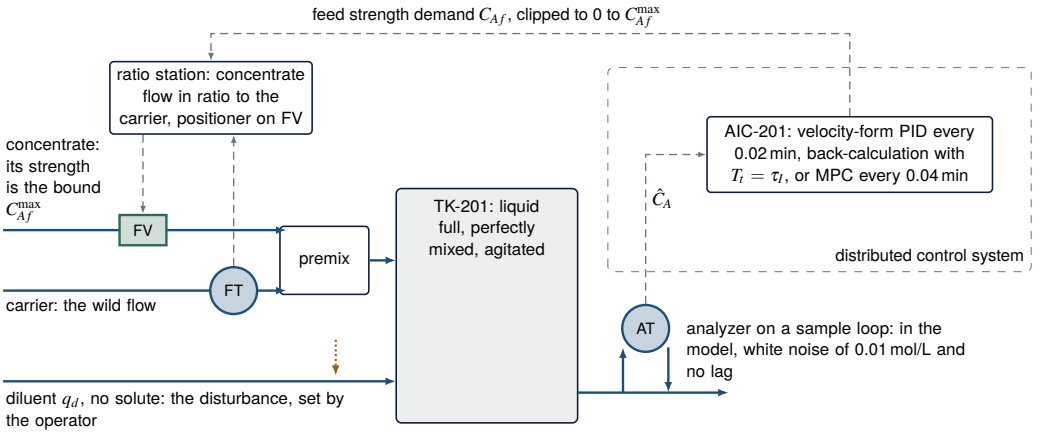


Figure 3.5 The loop of the chapter with real devices in place of the schematic's symbols. An analyzer AT on a sample loop reads the outlet, the controller AIC-201 on a distributed control system computes a feed strength, and a ratio station sets it by holding the concentrate flow through valve FV in ratio to the carrier flow measured by FT. The model collapses the ratio station into an ideal, instantaneous setting. Notice that the diluent is not measured, and that no valve opening can make the feed stronger than the concentrate.

What Moves It

The manipulated feed concentration of Equation (3.1) is applied with no valve and no lag. In a plant it is made by a ratio of two streams or by a dosing pump, so the composition loop is the outer loop of a cascade whose inner element has its own time constant, rangeability and stiction, as the fourth assumption of Section 3.3.4 states. The bound $C_{Af}^{\max}$ stands for the strength of the supply itself, which no valve opening can raise.

> **TECHNOLOGY** **Final Element: The Concentrate Valve of a Ratio Station**
>
> A feed of set strength is usually made by holding two flows in ratio. The transmitter on the uncontrolled, or wild, flow supplies the setpoint of a flow loop on the other line, whose flow is held captive to it [15]. That flow is set by a control valve in a cascade, with an inner loop, the valve positioner, that reduces the effects of friction and nonlinearity [10]. Friction the positioner cannot overcome makes the stem stick and then jump, and integral action then hunts above and below the setpoint without settling [15].

Where the Controller Runs

A distributed control system runs production at one site in refineries, water treatment, chemical manufacturing and pharmaceutical processing, with a supervisory level over the local controllers. Its loops are often nested, with the setpoint of one taken from another, and they run with cycle times from milliseconds to minutes [26]. The distributed control system replaced the cabinets of analog controllers and is now the standard tool of the process industry. Valve positioners close the innermost loops, while loops on quality variables sit at the higher levels [10]. AIC-201 is such a quality loop. Its PID executes every 0.02 min, and the predictive controller decides every 0.04 min. Model predictive control originated in the process industries, where the processes are often slow enough to allow the computation [10].

> **IN PRACTICE** **Windup Where the Setpoint Is Out of Reach**
>
> Reset windup happens when outside conditions make the setpoint impossible to reach. The integral term accumulates for as long as the error lasts, and the measurement must later pass the setpoint long enough to cancel what was stored. Controllers may limit how far the integral term can grow. The surest fix is the operator, who places the loop in manual. Because these conditions usually arise at shutdown, when operators run the process by hand anyway, windup often never shows [15]. A weak supply can bring it on in normal running, and there the back-calculation of Section 3.5 holds the stored output at the bound with the loop left in automatic.

What Is Modernizing It

Three developments touch this loop. The first is the smart positioner. An electronic positioner records actuator pressure against stem position, and this valve signature identifies excessive packing friction, trim interference and poor seating [15], the faults that make an inner flow loop stick. The second is the model of the mixing itself. Residence time models fed with real-time upstream data predict the downstream concentration and allow material that does not conform to be removed from the product [25]. The third is estimation and prediction in the controller, which Sections 3.5.4 and 3.5.5 already carry: a disturbance estimate that books the dilution as a feed deficit and a planner that sees the bound.

3.9 Limitations

The simulator is not intended for design work. A perfectly mixed vessel with no transport delay is the most favorable composition plant that can exist, and the dead time, the analyzer cycle and the sample line that dominate a real blend loop are all absent, so a tuning obtained here is optimistic by the amount of delay a real header carries. The manipulated variable is a feed concentration applied directly, with no inner ratio loop, no dosing pump dynamics, no rangeability limit, no stiction and no stroking time, so the only actuator imperfection represented is the bound itself. The vessel is liquid full with the outlet flow defined as the sum of the inlets, so there is no level loop and no inventory dynamic between the diluent valve and the outlet composition. A single solute is tracked, no reaction occurs, no energy balance is integrated, and the feed temperature is carried through the model and the exported file without entering any equation. Every case starts at steady operation, so a startup from an empty tank has to be configured by hand through the manual feed. The noise generator is reseeded whenever a run starts, so a case repeats sample for sample at any speed multiplier, noise included, while the seed itself is not an input.

Four things were kept deliberately, the bound on the feed, a diluent stream that shifts the gain and the time constant at once, white analyzer noise and a short sample period, because deleting any of them deletes something the loop must be commissioned around. Where each assumption breaks is stated in Section 3.3.4.

The predictive controller plans on one first-order model fixed at the design point. It omits the dilution, whose effect on the gain and time constant (0.833 and 0.833 min at 20 m^3/min) the estimate $\hat{d}$ must absorb, and it omits q and V, so a change in either is a mismatch the estimate also has to absorb. The measured diluent flow is not used as feedforward, the Kalman gain assumes the default analyzer noise, no output band reports an unreachable setpoint through a slack, $\lambda = 30$ comes from a rule shared with six other plants, and the comparison rests on one noise realization.

The moving horizon estimator bounds the state of that same fixed lag, so wherever the lag is wrong its estimate is wrong inside the bounds, and in the shipped cases the outlet stays 0.1 mol/L or more inside both. The relay records the switching band, the analyzer noise and the sampling more than the vessel, because a lag without dead time reaches $-180°$ only at the Nyquist frequency, and the Tyreus–Luyben conversion is one published rule among several. Chapter 2 reaches the same finding on a second lag without dead time.

3.10 Discussion

The reachable set of Equation (3.19) has a direct economic reading. Product giveaway is the quality delivered above the specification the customer pays for, and it exists because a blend has to be aimed away from its limit by enough to cover the variability of the composition loop. Real-time optimization of a gasoline blend header is profitable precisely to the extent that the regulatory layer beneath it can hold a tighter band, so every reduction in the standard deviation of the composition converts into a shift of the target toward the specification [2]. The planning layer above it assigns components to blends on the assumption that the recipes it computes can actually be delivered, and a recipe outside the reachable set of the blend header is an infeasible plan rather than a difficult one [1, 3].

Water treatment inverts the direction of the same argument. Coagulant dosing has an asymmetric cost, because underdosing fails the treatment objective while overdosing wastes chemical and increases sludge production, and raw water quality changes on a timescale no dosing pump can be tuned out of [4]. Continuous pharmaceutical manufacturing adds a specification that is legally rather than commercially binding, since blend uniformity is a release criterion and the plantwide control structure around a continuous mixer exists to keep the product inside it [5]. In each of those settings the useful question at a commissioning meeting is not how the loop is tuned but whether the requested composition lies inside what the streams can produce, and that question is answered by a steady-state balance rather than by a step test. A reader who has watched a well-tuned loop sit 0.167 mol/L short of setpoint with its feed on a bound has encountered that distinction in its simplest form.

The model of this chapter also supports control methods beyond those implemented, the PID loop, its relay autotune, the predictive controller and its two estimators. A model predictive controller built on Equation (3.1) rather than on the fixed lag of Section 3.5.4 would carry the dilution in its model instead of in a disturbance estimate, and a measured diluent flow would enter its prediction as feedforward information rather than as an unannounced load [27]. The two estimators of Section 3.5.5 already recover the true concentration from the noisy analyzer of Equation (3.3) for the planner, and reported as a soft sensor they would turn the offset the bound imposes into a displayed quantity rather than one read from a noisy trend. Economic operation follows directly, because giveaway minimization against the reachable set is an objective the one-equation balance can carry. The APMonitor framework implements these estimation and predictive control methods for differential algebraic models of this class [27], and the Gekko Optimization Suite exposes them in Python against the same exported data [28].

> **OPEN CHALLENGE** **Measuring a Blend Fast Enough to Control It**
>
> Continuous drug manufacture must hold its active ingredient tightly, and tracking that ingredient through the line is a major hurdle. In-line near-infrared spectroscopy watches blend composition, and residence time models can predict the downstream concentration from upstream data, but such models have served mainly for monitoring rather than as part of the control strategy [25]. Offline checks of blend uniformity still stop the process to take a sample, and an in-line probe must look at several locations for an accurate estimate [29]. Progress would be a composition measurement fast and representative enough to close a loop on.

3.11 Conclusion

This chapter studied a stirred blending tank whose manipulated variable is a composition and whose composition has a ceiling. The simulator integrates one species balance on a liquid-full, perfectly mixed vessel with a concentrated feed whose concentration is limited to 0 to 2 mol/L, a pure diluent stream as the load disturbance, and an analyzer carrying 0.01 mol/L of Gaussian noise.

The steady-state gain $q/(q+q_d)$ and the residence time $V/(q+q_d)$ share a denominator, so step tests in the running application return 1.0000 and 0.9950 min at zero dilution, against analytic values of 1.0000 and 1.0000 min, and 0.6667 and 0.6617 min at 50 m^3/min, against analytic values of 0.6667 and 0.6667 min. The internal model control proportional gain $V/(q\tau_c)$ carries no dilution term at all while the integral gain $(q+q_d)/(q\tau_c)$ carries it in full.

The reachability condition $q_d \le q(C_{Af}^{\max}/C_A^{\mathrm{SP}} - 1)$ closes at 33.3 m³/min for the default setpoint, and the measured loop crosses it between 33 m³/min, where the offset is 0.0001 mol/L, and 34 m³/min, where it is the 0.0075 mol/L the balance predicts. At 50 m³/min the offset is 0.1667 mol/L at both $K_c = 2.0$ and $K_c = 8.0$, and the trajectory while the feed is pinned fits a time constant of 0.6617 min at both gains, which is the open-loop value of the discrete plant.

Feedback is applied by an incremental PID whose back-calculation holds the stored output at the bound while the feed is pinned, so the feed leaves the bound 0.30 min after the disturbance clears, as the outlet crosses the setpoint, rather than after working off the 3.12 mol/L of integral demand a free integrator would have stored. A predictive controller planning on the zero-dilution model with a disturbance estimate rejects a 20 m³/min dilution with an integrated error of 0.061 against 0.150 mol min/L and no active bound, ties the PID while the feed is pinned and leaves the bound 0.02 min later, so its advantage lies in its model of the load rather than in its treatment of the constraint. Its moving horizon estimator, bounded by the feed range, never reaches a bound and changes the feed by at most 2.1×10^{-11} mol/L, and a relay autotune writes an integral time of 1.83 min against the residence time of 1.0 min and raises the true-value IAE of the dilution step over 15 min from 0.159 to 0.313 mol min/L. With integral action switched off, a 33 m³/min dilution leaves the outlet 0.149 mol/L below setpoint from a completely different cause, which integral action removes entirely, and at 50 m³/min it removes only the 0.048 mol/L by which that offset exceeds the reachability limit.

A fuller model would add an analyzer dead time, a cascade to an inner ratio loop, a level loop under the vessel and a second solute so that the reachable set becomes a region rather than an interval. The experiment to carry away from the blending tank is the one that separates the offset a controller removes from the offset an actuator bound imposes.

KEY POINTS

- The feed can never be stronger than its supply, so too much dilution leaves an error that no tuning removes.
- The gain and the residence time share a denominator, so internal model control keeps the proportional gain and moves only the integral gain.
- The reachable set closes between 33 and 34 m³/min of diluent at the default setpoint.
- A proportional loop leaves a standing error of similar size that integral action removes entirely.
- Back-calculation holds the stored output at the bound, so the feed leaves the bound as the outlet crosses the setpoint.

Where else this applies. A velocity-form PID with back-calculation against a hard stop also holds the car of Chapter 2.

References

[1] A. Charnes, W. W. Cooper, and B. Mellon, "Blending aviation gasolines—a study in programming interdependent activities in an integrated oil company," *Econometrica*, vol. 20, no. 2, p. 135, 1952.

[2] A. Singh, J. F. Forbes, P. J. Vermeer, and S. S. Woo, "Model-based real-time optimization of automotive gasoline blending operations," *Journal of Process Control*, vol. 10, no. 1, pp. 43–58, 2000.

[3] C. A. Méndez, I. E. Grossmann, I. Harjunkoski, and P. Kaboré, "A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations," *Computers & Chemical Engineering*, vol. 30, no. 4, pp. 614–634, 2006.

[4] H. Ratnaweera and J. Fettig, "State of the art of online monitoring and control of the coagulation process," *Water*, vol. 7, no. 11, pp. 6574–6597, 2015.

[5] R. Singh, M. Ierapetritou, and R. Ramachandran, "System-wide hybrid MPC–PID control of a continuous pharmaceutical tablet manufacturing process via direct compaction," *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 85, no. 3, pp. 1164–1182, 2013.

[6] Y. Gao, F. J. Muzzio, and M. G. Ierapetritou, "A review of the residence time distribution (RTD) applications in solid unit operations," *Powder Technology*, vol. 228, pp. 416–423, 2012.

[7] P. V. Danckwerts, “Continuous flow systems,” *Chemical Engineering Science*, vol. 2, no. 1, pp. 1–13, 1953.

[8] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, III, *Process Dynamics and Control*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.

[9] B. W. Bequette, *Process Control: Modeling, Design, and Simulation*. Upper Saddle River, NJ: Prentice Hall, 2003.

[10] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed. Princeton, NJ: Princeton University Press, 2021.

[11] K. J. Åström and L. Rundqwist, “Integrator windup and how to avoid it,” in *1989 American Control Conference*. IEEE, 1989, pp. 1693–1698.

[12] R. Hanus, M. Kinnaert, and J.-L. Henrotte, “Conditioning technique, a general anti-windup and bumpless transfer method,” *Automatica*, vol. 23, no. 6, pp. 729–739, 1987.

[13] M. V. Kothare, P. J. Campo, M. Morari, and C. N. Nett, “A unified framework for the study of anti-windup designs,” *Automatica*, vol. 30, no. 12, pp. 1869–1883, 1994.

[14] S. Tarbouriech and M. Turner, “Anti-windup design: an overview of some recent advances and open problems,” *IET Control Theory & Applications*, vol. 3, no. 1, pp. 1–19, 2009.

[15] T. R. Kuphaldt, *Lessons in Industrial Instrumentation*, version 2.33 ed. Open textbook, Creative Commons Attribution 4.0 license, ibiblio.org/kuphaldt/socratic/sinst, 2024.

[16] D. E. Rivera, M. Morari, and S. Skogestad, “Internal model control: PID controller design,” *Industrial & Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252–265, 1986.

[17] S. Skogestad, “Simple analytic rules for model reduction and PID controller tuning,” *Journal of Process Control*, vol. 13, no. 4, pp. 291–309, 2003.

[18] K. J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2006.

[19] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[20] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[21] G. Pannocchia and J. B. Rawlings, “Disturbance models for offset-free model-predictive control,” *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[22] D. Goldfarb and A. Idnani, “A numerically stable dual method for solving strictly convex quadratic programs,” *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[23] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, “Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations,” *IEEE Transactions on Automatic Control*, vol. 48, no. 2, pp. 246–258, 2003.

[24] K. J. Åström and T. Hägglund, “Automatic tuning of simple regulators with specifications on phase and amplitude margins,” *Automatica*, vol. 20, no. 5, pp. 645–651, 1984.

[25] S. Hurley, A. Tantuccio, M. S. Escotet-Espinoza, M. Flamm, and M. Metzger, “Development and use of a residence time distribution (RTD) model control strategy for a continuous manufacturing drug product pharmaceutical process,” *Pharmaceutics*, vol. 14, no. 2, p. 355, 2022.

[26] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, “Guide to operational technology (OT) security,” National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-82r3, 2023.

[27] J. D. Hedengren, R. Asgharzadeh Shishavan, K. M. Powell, and T. F. Edgar, “Nonlinear modeling, estimation and predictive control in APMonitor,” *Computers & Chemical Engineering*, vol. 70, pp. 133–148, 2014.

[28] L. D. R. Beal, D. C. Hill, R. A. Martin, and J. D. Hedengren, “GEKKO optimization suite,” *Processes*, vol. 6, no. 8, p. 106, 2018.

[29] E. J. Kim, J. H. Kim, M.-S. Kim, S. H. Jeong, and D. H. Choi, “Process analytical technology tools for monitoring pharmaceutical unit operations: A control strategy for continuous process verification,” *Pharmaceutics*, vol. 13, no. 6, p. 919, 2021.

3.12 Exercises

3.1 Concept check · 2 min

The loop runs with the default tuning, $K_c = 2.0$ and $\tau_I = 1.0\,\text{min}$, which is the internal model control tuning for $\tau_c = 0.5\,\text{min}$ with the diluent shut. The diluent flow then rises to $50\,\text{m}^3/\text{min}$ and stays there, with $q = 100\,\text{m}^3/\text{min}$ and $V = 100\,\text{m}^3$. Which retuning keeps the same τ_c?

(a) Raise the gain to 3.0 and keep the integral time, because the process gain has fallen to 0.667

(b) Raise the gain to 3.0 and shorten the integral time to 0.667 min, because both constants moved

(c) Keep the gain at 2.0 and shorten the integral time to 0.667 min, because both constants fell together

(d) Keep the gain and the integral time, because the gain and the time constant fell by the same factor

3.2 Concept check · 1 min

With the default tuning and a setpoint of 1.5 mol/L, the diluent flow steps to $50\,\text{m}^3/\text{min}$. The outlet settles at 1.333 mol/L with the feed on its 2.00 mol/L limit. Which change brings the outlet back to the setpoint?

(a) A proportional gain of 8.0 in place of 2.0, which quadruples the loop gain

(b) An integral time of 0.667 min, which matches the new residence time

(c) A switch to the predictive controller, which carries the bound in its plan

(d) A diluent flow of $33\,\text{m}^3/\text{min}$ or less, which puts the required feed in range

3.3 Concept check · 2 min

The feed has been on its limit for most of 10 min against a diluent flow of $50\,\text{m}^3/\text{min}$. The diluent valve is then closed. Under the default tuning the outlet peaks at 1.6266 mol/L, an overshoot of 8.44 %. Is this overshoot integrator windup?

(a) No. The stored output sat at the bound itself, and a reachable upset that never saturates gives nearly the same peak

(b) No. The overshoot comes from analyzer noise, which the proportional term passes to the feed while it is on the bound

(c) Yes. The integral grew for 9.34 min on the bound and had to be worked off before the feed could move

(d) Yes. The feed stayed on the bound for 0.30 min after the valve closed, the time the integral took to unwind

3.4 Theory and calculation · 5 min

The tank holds $V = 100\,\text{m}^3$. The feed flow is raised to $q = 200\,\text{m}^3/\text{min}$ and the diluent runs at $q_d = 50\,\text{m}^3/\text{min}$.

(a) Find the process gain and the residence time. *Checkpoint:* 0.80 and 0.40 min.

(b) The feed concentration is held at 1.5 mol/L. Find the steady outlet concentration. *Checkpoint:* 1.20 mol/L.

(c) Find the internal model control settings for $\tau_c = 0.5\,\text{min}$, and say which of them changes when the diluent valve is shut. *Checkpoint:* $K_c = 1.0$ and $K_I = 2.5/\text{min}$. Only the integral gain changes, to 2.0/min.

3.5 Theory and calculation · 6 min

A customer asks for 1.5 mol/L from the default tank, with $q = 100\,\text{m}^3/\text{min}$ and a supply that ends at $C_{Af}^{\max} = 2.0\,\text{mol/L}$.

(a) Find the largest diluent flow at which this setpoint is reachable. *Checkpoint:* 33.3 m³/min.

(b) The diluent runs at 40 m³/min. Find the highest outlet concentration the tank can hold and the standing error. *Checkpoint:* 1.429 mol/L and 0.071 mol/L.

(c) Operations proposes to restore the setpoint against 50 m³/min of diluent by raising the feed flow. Find the feed flow needed and the residence time at that flow. *Checkpoint:* At least 150 m³/min, with 0.50 min.

3.6 Theory and calculation · 7 min

Integral action is switched off, the gain is $K_c = 2.0$ and the stored output is frozen at $b = 1.50\,\text{mol/L}$. The setpoint is 1.5 mol/L and $q = 100\,\text{m}^3/\text{min}$.

(a) The diluent opens to 20 m³/min. Find the steady outlet concentration, the offset and the feed concentration. *Checkpoint:* 1.406 mol/L, 0.094 mol/L and 1.69 mol/L.

(b) Find the gain that would lower this offset to 0.05 mol/L, and check that the feed stays inside its range. *Checkpoint:* $K_c = 4.8$, with the feed at 1.74 mol/L.

(c) The diluent opens to 50 m³/min instead and the gain is 8.0. Show that Equation (3.21) does not apply, and give the offset that results. *Checkpoint:* The demand of 2.13 mol/L is past the bound. The offset is 0.167 mol/L.

3.7 App challenge · 9 min

Load scenario 2, Open-loop actuator step. The loop is in Manual and the feed concentration steps from 1.50 to 1.90 mol/L at 1 min. Use the default seed, run each case to 6 min and export the CSV file. Read the true outlet concentration, column `true_pv`.

(a) Run the scenario as shipped. Read the value at 6 min and compute the gain of the step. *Checkpoint:* 1.897 mol/L, within 0.003 mol/L, and a gain of 0.99.

(b) Find the time after the step at which the true outlet has covered 63.2 % of the change to 1.90 mol/L. *Checkpoint:* 1.0 min, within 0.05 min.

(c) Press the reload button, set the Dilution flow to 50 m³/min before Run and run again. Read the values at the start and at 6 min, and repeat part (b) for the settled change. *Checkpoint:* 1.000 and 1.267 mol/L, a gain of 0.67, and 0.66 min, within 0.05 min.

3.8 App challenge · 9 min

This exercise sets an offset that a controller removes beside one that it cannot remove. Use the default seed and run each case to 5 min. At 5 min read the IAE tile of the scorecard, and in the exported file the true outlet concentration (`true_pv`) and the feed concentration (`mv`).

(a) Load scenario 4, Disturbance rejection, and run it as shipped. The diluent opens to 20 m³/min at 1 min. *Checkpoint:* IAE 0.150 mol min/L, outlet 1.494 mol/L, feed 1.80 mol/L.

(b) Press the reload button, set the Integral time to its top, 20 min, which switches integral action off, and run again. *Checkpoint:* Outlet 1.406 mol/L, feed 1.69 mol/L.

(c) Load scenario 8, Actuator limit, and run it as shipped. The diluent stands at 20 m³/min and the setpoint steps from 1.5 to 1.9 mol/L at 1 min. *Checkpoint:* IAE 1.066 mol min/L, outlet 1.665 mol/L, feed 2.00 mol/L.

(d) Press the reload button, set the Proportional gain to 8 and run again. *Checkpoint:* IAE 1.065 mol min/L, outlet 1.665 mol/L, feed 2.00 mol/L.

Accept an IAE within 0.005 mol min/L, an outlet within 0.005 mol/L and a feed within 0.05 mol/L. Name the equation that predicts each standing error.

3.9 App challenge · 7 min

Load scenario 9, MPC at the edge of the feed. The diluent opens to 20 m^3/min at 1 min, the setpoint steps to 1.70 mol/L at 2 min and returns to 1.60 mol/L at 3.5 min. Use the default seed and run each case to 5 min. For each run read the IAE tile of the scorecard and, in the exported file, the true outlet concentration at 2.0 and at 3.5 min.

(a) Run the scenario as shipped, with the Controller at MPC. *Checkpoint:* IAE 0.207 mol min/L, 1.497 and 1.639 mol/L.

(b) Set the Controller to PID, press Reset and run again. *Checkpoint:* IAE 0.281 mol min/L, 1.434 and 1.629 mol/L.

Accept an IAE within 0.005 mol min/L and an outlet within 0.005 mol/L. State the highest outlet concentration either controller could hold between 2 and 3.5 min, and say where the MPC earns its lower IAE.

3.10 Application area · 4 min

A blend is sold against a minimum concentration, and the setpoint is aimed above that minimum by three standard deviations of the true outlet concentration. Table 3.8 gives this standard deviation as 0.00121 mol/L at $K_c = 2.0$ and 0.00297 mol/L at $K_c = 8.0$, with the same analyzer noise. The outlet flow is 100 m^3/min.

(a) Find the giveaway for each tuning, in mol/L and in solute per day. *Checkpoint:* 0.0036 mol/L and 523 kmol per day, 0.0089 mol/L and 1283 kmol per day.

(b) State from Table 3.7 what the higher gain buys in return, and whether it buys anything while the feed is on its bound. *Checkpoint:* A return within 1 % in 1.83 min against 3.72 min. Nothing on the bound.

3.11 Application area · 3 min

Section 3.9 states that a real blend header carries a transport delay and an analyzer cycle that the model omits. Suppose the analyzer adds a dead time of 0.5 min to the default tank, with $K_p = 1$ and $\tau = 1.0$ min.

(a) Find the PI settings of Equation (1.11) with $\tau_c = \theta_p$. *Checkpoint:* $K_c = 1.0$ and $\tau_I = 1.0$ min.

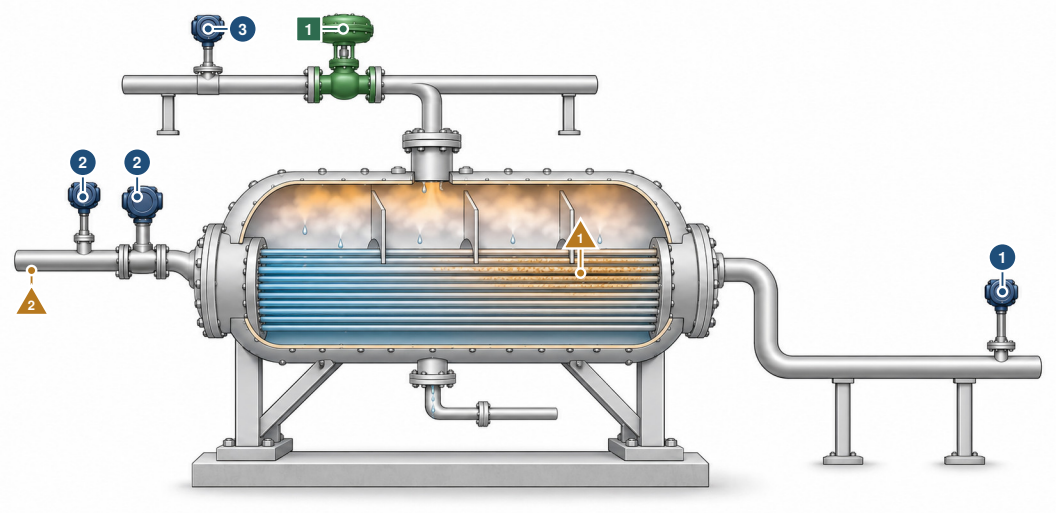
(b) Find the dead time at which the same rule returns the default gain of 2.0, and state what this means for a tuning carried from the app to a plant. *Checkpoint:* 0.25 min.

3.12 Review the agent · 5 min

An AI assistant is given the result of the first relay test of Section 3.7.7 and replies as follows. “The test measured an ultimate gain of 5.55 and an ultimate period of 0.830 min. The Tyreus–Luyben rule gives $K_c = 1.73$ and $\tau_I = 1.83$ min, which should replace the shipped 2.0 and 1.0 min. The loop becomes unstable at a gain of 5.55, so the proportional gain must stay below about half of that.” Find the flaw in the reply and say how it could be checked.

CHAPTER 4

Steam-Heated Exchanger Laboratory with Hidden Parameters for Step Testing and Tuning



The controller moves 1 the steam valve

The controller reads 1 the outlet temperature, read far down the pipe 2 the feed's temperature and flow 3 the steam header pressure

What disturbs it 1 deposits building up on the tubes 2 changes in feed rate and feed temperature

The controller moves only the steam valve, and it reads the product temperature well past the bundle, so each move shows up late. Illustration, not to scale.

IN PLAIN TERMS **Tuning a Steam Heater From One Test**

The main idea. Steam flows around a bundle of tubes and warms the liquid inside them. A controller opens or closes the steam valve to hold the liquid at a set temperature.

The challenge. Its settings come from one small test that moves the valve and times the reply, but the reply changes with how much liquid flows. The thermometer sits far down the pipe, and a wait read too short makes the controller too bold, so the valve swings back and forth.

Why it matters. Steam is the most common way a plant warms a stream, and a misread test wears the valve for no gain.

THREE WAYS TO READ THIS CHAPTER

Explore	11 min	the first page, the fact on a real steam heater (Section 4.2), the try-it segment of case 8 and the history of predictive control (Section 4.5), the try-it segment of case 14 (Section 4.6), the fact in the results (Section 4.7) and the key points
Study	65 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	25 min	the first page, the results (Section 4.7), Technology in Practice (Section 4.8), Limitations (Section 4.9) and the two open challenges

The control question The constants of a steam-heated exchanger are hidden, and its outlet temperature loop has a dead time and a gain that both move with the feed rate. What does one valve step test identify on such a plant, and how far can a tuning rule be trusted with the three numbers it returns?

Learning objectives

- Write the energy balances of the exchanger and compute its duty, steam flow and transport delay at an operating point.
- Identify a first-order-plus-dead-time model from a valve step and explain why its dead time exceeds the transport delay.
- Apply the SIMC and reaction-curve rules to the fit and predict what a misread dead time does to each.
- Distinguish a duty limit, fouling and valve stiction from a tuning problem by the controller output.
- Compare a single loop, a cascade, a heat-balance feedforward and a predictive controller built from the same identified model.

Prerequisites Sections 1.4.4 (step test and fit), 1.4.5 (tuning rules), 1.4.9 (windup) and 1.6.2 (the algorithm of the apps). Section 1.6.3 is needed only for the predictive controller. Chapter 2 introduces the step test on a simpler plant.

The app https://apmonitor.com/apps/heat-exchanger/

See also Volume 1, Chapter 2 and Volume 1, Chapter 4 for step tests on a tank and on a heated device. Volume 3, Chapter 2, Volume 3, Chapter 3 and Volume 3, Chapter 6 for the choice of structure, the speed of an inner loop and feedforward with a feedback trim. Volume 13, Chapter 7 for a transport delay that stretches with flow, and Volume 13, Chapter 4 for exchanger bypasses that set reactor inlet temperatures.

4.1 Introduction

Steam condensing on one side of a tube bundle is the most common way a process plant adds heat to a stream. The service appears as feed preheat ahead of a reactor, as reboiler duty on a distillation column, as product conditioning before storage, and as jacket service on a fermenter, and the temperature loop that regulates it is usually the first controller a chemical engineering student designs [1, 2]. That loop is also the one on which a formula-sheet approach fails first, for four reasons that are physical rather than incidental. A transport delay separates the bundle from the transmitter. The steady-state gain is not a constant because the driving force is a log-mean temperature difference that collapses as the outlet approaches the tube wall. The tube-side film coefficient rises with the 0.8 power of flow [3, 4]. The actuator is a valve whose capacity depends on the square root of a pressure drop that the valve itself destroys as it opens.

Those four mechanisms are consequences of three fundamentals that a first course states in general form. Conservation of energy applied to the fluid, to the tube metal and to the shell gives one state equation per inventory, and the number of inventories in series sets the order of the response and therefore the phase lag a controller must survive [1]. Heat crosses each surface in proportion to a driving force, so the input-output gain of the loop is the derivative of a driving force rather than a constant, and it changes with the operating point. Time-scale separation orders the resulting lags, with the tube metal and the process inventory slow, the shell and the valve fast, and the outlet line a pure transport delay, which is what allows a fifth-order plant to be identified as a first-order-plus-dead-time (FOPDT) model at all [5]. The experiments of this chapter therefore cover process order, dead time, empirical identification from a reaction curve, PID tuning and anti-windup, sampled measurements, cascade structure, feedforward from a measured disturbance, gain scheduling, and the distinction between a saturated loop and a detuned one.

A simulator that supplies a FOPDT transfer function and asks for a controller exercises none of that behavior. The application of this chapter inverts the arrangement. Exchanger E-101 is modeled from first principles, and the constants of that model, including the two heat-transfer coefficients, the metal mass, the shell capacitance, the valve capacity and the transport volume, are withheld from the parameter panel, which lists only data-sheet and instrument values. What is available is what a real

unit is instrumented for, an outlet temperature transmitter, an inlet temperature transmitter, a feed flow meter, a steam flow meter, a header pressure gauge and a valve position. A model is obtained the way a commissioning engineer obtains one, by perturbing the plant and reading the answer off a trend, the step test of Section 1.4.4.

Dynamic exchanger models of this kind are well established. Distributed and lumped formulations appear in the simulation literature [6], fouling has been treated as a series resistance since the Kern and Seaton deposition model [7], compound cascade and feedforward structures are the industrial default [8], and nonlinear and predictive controllers have been applied to the same unit [9]. However, the published models are presented with their parameters supplied, which removes the identification step that determines whether the resulting controller works at all. This chapter works with an interactive steam-heated exchanger in which the plant parameters are withheld. The identified model, the simulator equations and the controller's internal model are kept separate, so a misread trend produces a badly tuned loop rather than a hidden correction.

The chapter is organized as follows. Section 4.2 gives the history of the loop and of its models. Section 4.3 gives the five-state model, its instrument chain and every constant needed to rebuild the simulator. Section 4.4 identifies a FOPDT model from a valve step and shows how its three numbers move with the feed rate. Section 4.5 describes the PID algorithm, the single-loop, cascade and feedforward structures, the two tuning rules, and a model predictive controller that plans on the identified model, with an alarm band that can be tightened by the misfit of that model. Section 4.6 derives the three properties that govern the tuning, a dead time and a gain that move with load and a duty limit that looks like a tuning problem. Section 4.7 reports the measured comparisons, which start in manual operation and end in fault diagnosis. Section 4.8 names the devices and the platform of such a loop in a plant. Sections 4.9 and 4.10 state what the model cannot do and what the loop is worth on a plant, and Section 4.12 holds the exercises.

4.2 Background

Automatic regulation of heat-transfer duty preceded the theory that describes it. Nineteenth-century steam users throttled supply valves by hand against a thermometer, and the earliest self-acting devices were direct thermal regulators in which a filled bulb drove the valve with no external power. The development that matters for this chapter is the pneumatic three-term controller of the 1930s, which added reset and pre-act to a proportional response and made it possible to hold a temperature at setpoint rather than near it [10]. Ziegler and Nichols published their reaction-curve and continuous-cycling settings in 1942 for this class of loop [11], and the reaction curve remains the archetype of the step test the application asks the reader to perform. The model-based rules that displaced them derive the same three settings from an assumed model and an explicit closed-loop speed, which makes the quality of the identification the dominant term in the result [12, 13].

The modeling literature followed later. Steady-state exchanger design had long rested on film correlations of the Dittus-Boelter and Sieder-Tate form [3, 4], but the dynamic behavior a controller sees, metal capacitance in series with fluid capacitance plus a transport lag, was analyzed explicitly only as digital simulation became routine [6]. Fouling followed a similar path. The Kern and Seaton deposition-minus-removal picture, reviewed fifty years on by Müller-Steinhagen [7], expresses the deposit as a series resistance and makes the fouling factor a quantity that a controller output can be used to estimate.

The exchanger is a standard control testbed because a single PID loop is not the best available structure for it. Compound feedforward and cascade structures remain the industrial default on outlet temperature services [8], and the nonlinear input-output gain has motivated a long line of predictive and nonlinear controllers [9]. Surveys of control education report a persistent gap between that literature and the textbook treatment, namely the absence of an accessible plant on which the single loop can be measured before the structures that improve it are introduced [14, 15]. Volume 3,

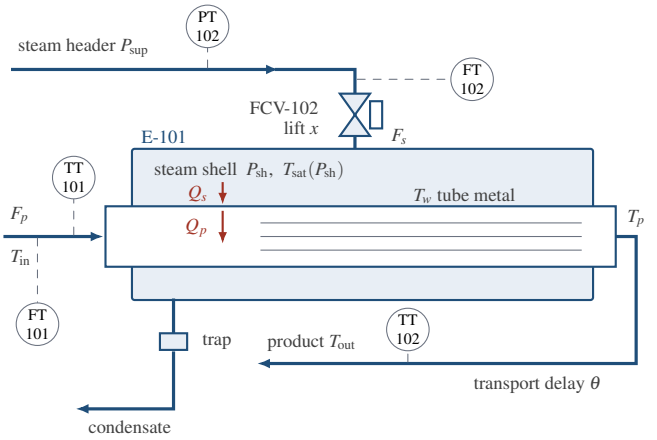


Figure 4.1 Exchanger E-101 with the steam header, control valve FCV-102, the condensing shell, the tube bundle and the outlet transport line. The process stream enters the tubes at T_{in} with flow F_p, receives duty Q_p from the tube metal at T_w, and leaves the bundle at T_p. Steam admitted at lift x against header pressure P_{sup} raises the shell to P_{sh} and delivers duty Q_s to the metal. TT-102 reads the product θ minutes after it leaves the bundle.

Chapter 2 takes up the choice among those structures for a given disturbance, and this chapter measures the single loop first.

> **DID YOU KNOW? A Steam Heater in Acid Service**
>
> In one phosphoric acid plant, studied in 2023, steam at 116 to 125 °C heats the acid from about 70 to about 80 °C at volume flows of up to 3407 m^3/h, and the fouling resistance of the exchanger was computed from plant readings over runs of up to 122 h [16].

4.3 Process Model

Exchanger E-101 heats a process liquid from a nominal 40 °C to a setpoint of 85 °C using saturated steam condensing in the shell. Steam arrives from a header at 4 barg through control valve FCV-102. Condensate leaves through a trap, which is represented only through its effect on the condensing area. The arrangement, the instruments and every symbol that appears in the equations below are shown in Figure 4.1. The controlled variable is the product outlet temperature reported by transmitter TT-102 downstream of the bundle, and the manipulated variable is the position of FCV-102 in every control structure the application offers.

The variables a controller interacts with are classified in Table 4.1. Four alarm limits bound normal operation, TAHH-102 at 95 °C, TAH-102 at 90 °C, TAL-102 at 79 °C, and FAL-102 on steam flow at 250 kg/h. The alarms do not trip the simulation. They latch and annotate the record, which is enough to show that an aggressive test has left the product off specification.

The plant is written in operator units, temperature in °C, pressure in bar absolute, process flow in kg/h, steam flow in kg/min, valve position in percent, with time in minutes. Five states are integrated. Each follows from a conservation law applied to one inventory, the process fluid in the bundle, the tube metal, the steam in the shell, the valve stem and the fouling deposit, which is the standard route from physical fundamentals to a state-space process model [2].

Table 4.1 Classification of Variables. Nominal Values Are Those of the Design Operating Point at Which Most Scenarios Start.

Symbol	Role	Nominal	Range
T_{out} (TT-102)	CV, measured	85 °C	60–95 °C SP
u (FCV-102)	MV	44.9 %	0–100 %
F_s (FT-102)	inner CV, measured	1727 kg/h	0–4000 kg/h span
F_p (FT-101)	DV, measured	20 000 kg/h	12 000–30 000 kg/h
T_{in} (TT-101)	DV, measured	40 °C	25–55 °C
P_{sup} (PT-102)	DV, measured	4 barg	1.5–5.5 barg
ϕ fouling	DV, unmeasured	0 %	0–150 %
T_p	state	85.0 °C	—
T_w	state	111.6 °C	—
P_{sh}	state	2.60 bar abs	0.3–P_{sup}
x	state	44.9 %	0–100 %
R_f	state	0	≥ 0

4.3.1 Energy Balances

The process fluid in the bundle is treated as a single well-mixed inventory M_p that receives heat from the tube metal and exchanges sensible heat with the throughput, as shown in Equation (4.1).

$$M_p C_p \frac{\mathrm{d}T_p}{\mathrm{d}t} = \dot{m}_p C_p \left(T_{\text{in}} - T_p\right) + Q_p \tag{4.1}$$

where M_p is the process-side thermal inventory in kg, C_p is the process heat capacity in kJ/kg/K, $\dot{m}_p = F_p/60$ is the process mass flow in kg/min, T_{in} is the inlet temperature and Q_p is the duty crossing from the metal in kJ/min. The tube metal is a second capacitance between the condensing steam and the process fluid, as shown in Equation (4.2).

$$M_w C_w \frac{\mathrm{d}T_w}{\mathrm{d}t} = Q_s - Q_p \tag{4.2}$$

where $M_w C_w$ is the tube-metal heat capacity in kJ/K and Q_s is the duty delivered by condensing steam. Both sides of Equations (4.1) and (4.2) carry units of kJ/min. The two duties are given by Equations (4.3) and (4.4).

$$Q_p = UA_p \cdot \Delta T_{\text{lm}}(T_w - T_{\text{in}}, T_w - T_p) \tag{4.3}$$

$$Q_s = UA_s \left[T_{\text{sat}}(P_{\text{sh}}) - T_w\right] \tag{4.4}$$

where UA_p and UA_s are the process-side and condensing-side conductances in kJ/min/K and T_{sat} is the shell saturation temperature. The process-side driving force is the log mean of the two terminal differences, as shown in Equation (4.5).

$$\Delta T_{\text{lm}}(a,b) = \frac{a-b}{\ln(a/b)} \tag{4.5}$$

The arithmetic mean $\frac{1}{2}(a+b)$ replaces (4.5) whenever either end falls to 1×10^{-9} K or below, or when the two ends agree to within one part in 10^6. That substitution keeps the right-hand side smooth enough for a fixed-step integrator during a transient in which the fluid is briefly hotter than the metal. Retaining the log mean rather than a single lumped ΔT is the mechanism by which the process gain falls as the outlet approaches the wall.

🕮 **Pause.** Try Exercise 4.2 before reading on.

4.3.2 Steam Side

The shell is a pressure capacitance charged by the valve and discharged by condensation, as shown in Equation (4.6).

$$C_{\mathrm{sh}}\frac{\mathrm{d}P_{\mathrm{sh}}}{\mathrm{d}t} = \dot{m}_s(x, P_{\mathrm{sh}}) - \frac{Q_s}{\lambda_{\mathrm{eff}}(P_{\mathrm{sh}})} \tag{4.6}$$

where C_{sh} is the shell steam capacitance in kg/bar, $\dot{m}_s$ is the steam admitted in kg/min and λ_{eff} is the latent heat actually delivered in kJ/kg. The shell pressure is bounded by $0.3 \le P_{\mathrm{sh}} \le P_{\mathrm{sup}}$ in two places. The derivative in (4.6) is set to zero when the state is at a bound and the derivative points outward, so the integrator cannot step through a wall and back, and the state is clipped again after each step. Saturated-steam properties are quadratics in $L = \ln P$ with P in bar absolute, fitted over 1–10 bar, as shown in Equations (4.7) and (4.8).

$$T_{\mathrm{sat}} = 99.6 + 26.772\,L + 3.519\,L^2 \quad [^\circ\mathrm{C}] \tag{4.7}$$

$$\lambda = 2258 - 67.33\,L - 16.59\,L^2 \quad [\mathrm{kJ/kg}] \tag{4.8}$$

Both fits sit within 0.4 K and 2 kJ/kg of tabulated saturated-steam properties across that range, which is tighter than any other approximation in the model. Wet steam is handled by $\lambda_{\mathrm{eff}} = \chi\lambda$ with quality χ available as an operator-set input over $[0.8, 1]$ and bounded to $[0.5, 1]$ internally. At the design point the shell settles at 2.60 bar absolute (1.59 barg) and 128.4 °C against a 4 barg header, because nothing forces the shell to sit at header pressure. The shell finds the pressure at which the steam admitted equals the steam condensed.

4.3.3 Valve, Film Coefficients and Transport Delay

Steam flow through FCV-102 obeys an installed characteristic $f(x) = x^a$ with x the fractional lift, a square-root pressure-drop law, and a cap at the header capacity, as shown in Equation (4.9).

$$\dot{m}_s = \min\left(\dot{m}_{s,\max} f(x)\sqrt{\frac{P_{\mathrm{sup}} - P_{\mathrm{sh}}}{\Delta P_{\mathrm{nom}}}},\ \frac{F_{s,\max}}{60}\right) \tag{4.9}$$

where $\dot{m}_{s,\max}$ is the valve capacity at the design pressure drop ΔP_{nom} and $F_{s,\max}$ is the header capacity in kg/h. The flow is zero when $P_{\mathrm{sup}} - P_{\mathrm{sh}}$ is not positive. The exponent a is user-adjustable. Linear trim is $a = 1$, $a = 2$ behaves like equal-percentage trim whose gain rises with lift, and $a = 0.5$ is quick-opening. The stem is a first-order actuator with $\tau_v\,\mathrm{d}x/\mathrm{d}t = x_{\mathrm{cmd}} - x$ and is clipped to 0–100 % after each step.

The process-side conductance carries the turbulent film exponent and the fouling resistance in series, as shown in Equation (4.10).

$$UA_p = \left[\frac{1}{UA_{p0}\,(F_p/F_{p0})^n} + R_f\right]^{-1}, \qquad n = 0.8 \tag{4.10}$$

where UA_{p0} is the clean conductance at the design feed flow F_{p0} and R_f is the fouling resistance in min K/kJ. Equation (4.10) is the Dittus-Boelter and Sieder-Tate scaling written at the equipment level [3, 4]. Fouling resistance is $R_f = (\phi/100)\,R_{f,\max}$ where ϕ is a percentage set directly as an operator input or accumulated by an accelerated ramp $\mathrm{d}\phi/\mathrm{d}t$ that is the fifth integrated state. Condensate flooding reduces the condensing side by $UA_s = UA_{s0}\max(0.05,\, 1 - \varphi/100)$ with φ the flooded fraction in percent.

TT-102 is located downstream of the bundle and therefore measures fluid that left the bundle θ minutes earlier, as shown in Equation (4.11).

$$\theta = \frac{V_{\mathrm{trans}}}{\dot{m}_p/\rho} \tag{4.11}$$

Table 4.2 Model Parameters. Hidden Parameters Are Withheld From the Application's Parameter Panel and Must Be Inferred From Experiments.

Symbol	Meaning	Value	Unit
Hidden, the plant to be identified			
UA_{p0}	process-side UA at design flow	1380	kJ/min/K
UA_{s0}	condensing-side UA	3726	kJ/min/K
M_p	process thermal inventory	1200	kg
$M_w C_w$	tube-metal heat capacity	500	kJ/K
C_{sh}	shell steam capacitance	3.0	kg/bar
n	tube-side film exponent	0.8	—
V_{trans}	outlet transport volume	0.1	m^3
$R_{f,\max}$	fouling resistance at 100 %	2.6×10^{-4}	min K/kJ
$\dot{m}_{s,\max}$	valve capacity at design ΔP	64.0	kg/min
ΔP_{nom}	design pressure drop	2.4	bar
τ_v	valve actuator lag	0.20	min
F_{p0}	design feed flow	20 000	kg/h
Visible, data sheet and instruments			
C_p	process heat capacity	4.18	kJ/kg/K
ρ	process density	1000	$\mathrm{kg/m}^3$
$F_{s,\max}$	steam header capacity	2600	kg/h
FT-102 span	steam flow transmitter span	4000	kg/h
τ_{TT}	TT-102 lag	0.10	min
τ_{FT}	FT-102 lag	0.05	min
a	installed valve characteristic	1.0	—
Adjustable disturbances			
ϕ	fouling	0	%
φ	condensate flooding	0	%
χ	steam quality	1.0	—
S	valve stiction deadband	0	%
b	TT-102 bias	0	°C
σ	TT-102 noise	0	°C
Numerics			
Δt	integration step (RK4)	0.005	min
Δt_c	controller sample period	0.02	min
—	historian rate	50	samples/min

where V_{trans} is the outlet transport volume in m^3 and ρ is the process density in $\mathrm{kg/m}^3$. The result is clipped to 0–6 min. The delayed temperature receives an optional bias and optional band-limited noise and then passes through a first-order transmitter lag τ_{TT} before the controller reads it. Equation (4.11) carries the principal consequence for control design, because the dead time is not a constant of the plant. It is a fixed volume divided by a variable flow, so it is inversely proportional to production rate.

Every parameter needed to rebuild the simulator is listed in Table 4.2. At the design point the exchanger holds 85.0 °C with the valve at 44.9 %, passing 1727 kg/h of steam for a duty of 1045 kW against 1252 kW available with the valve wide open. The overall conductance is 16.78 kW/K and the transport delay is 0.30 min.

4.3.4 Steady-State Initialization

Every scenario begins at an algebraic steady state for its manual valve position or its setpoint rather than at an arbitrary initial vector. Scenario 5 then sets the fluid in the bundle to 78 °C to start the product cold. Given a valve position, a bisection on P_{sh} with a nested bisection on T_w solves (4.1) through (4.6) with all derivatives set to zero, with T_p following directly from the fluid balance, and a further bisection inverts that map to find the valve position that holds a requested outlet temperature.

Bisection rather than Newton iteration is used because every residual involved is monotone and bisection cannot diverge on an extreme disturbance setting. Each bisection runs 44 iterations, which brackets the shell pressure to better than 10^{-12} bar. The same routine supplies the balancing valve position offered during manual operation and the available-duty figure reported with the performance measures, and it is refreshed no more than four times per simulated minute because it is expensive relative to a single integration step.

4.3.5 Assumptions

The model retains only the mechanisms that change how the loop must be tuned. Each assumption below is stated with the range over which it holds and the condition under which it fails.

1. *Lumped process side.* The fluid in the bundle is one well-mixed inventory with a log-mean driving force to the metal rather than a distributed axial profile. The steady-state duty and the dominant time constant are correct, and the fitted gain at the design point agrees with a steady-state sweep to within 1 %. The first few seconds of a step response are approximate, and the assumption fails outright for any question about local tube-wall temperature, vaporization inside the tubes, or axial fouling distribution.

2. *Saturated shell at a single pressure.* The shell is saturated throughout, with T_{sat} and λ from (4.7) and (4.8). This holds between 1 and 10 bar absolute, which covers the whole adjustable range of the steam header. It fails during a rapid pressure transient, when a real shell is neither uniform nor exactly saturated, and it fails for superheated supply, non-condensables and shell-side maldistribution, none of which are represented.

3. *Lumped shell capacitance.* The steam side is one capacitance in kg/bar, so the pressure state responds as a first-order lag driven by the imbalance between admission and condensation. A real shell has a fixed vapor volume in which density and temperature both change, which makes the true pressure response stiffer than this. The consequence is that the shell contributes slightly more apparent dead time in the simulator than it would on the corresponding unit.

4. *Perfect condensate removal.* The trap is not modeled. Condensate is removed as fast as it forms unless the flooding disturbance is applied, and flooding then removes condensing area in direct proportion. This is adequate for showing loss of duty and the resulting valve drift. It cannot reproduce stall, subcooling, condensate level dynamics or the differential pressure that drives a real trap.

5. *Fouling as one uniform series resistance.* The deposit is a single R_f added in series with the clean film, ramped at a constant rate. The series form is standard and the effect on UA is correct. Real deposits build unevenly, change the tube-side pressure drop as well as the heat transfer, and follow a deposition-minus-removal law with an asymptote rather than a constant rate [7]. The accelerated rate of up to 3 %/min compresses months into minutes and is not a claim about fouling kinetics.

6. *Turbulent film scaling.* The tube-side coefficient follows $F_p^{0.8}$ over the full feed range of 12 000 to 30 000 kg/h, which stays turbulent for a water-like fluid in a bundle of this duty. The correlation is extrapolated and wrong at very low throughput, where the flow would be transitional or laminar and the exponent nearer 0.33.

7. *Transport delay as a moving tap.* The outlet delay is a fixed volume divided by volumetric flow, implemented as a linearly interpolated tap on a history buffer rather than as true plug flow. This reproduces the throughput dependence exactly and is correct in steady state. During a rapid flow change the tap moves faster than fluid physically can, so the delay transitions instantaneously where a real line would blend two residence times.

8. *Constant physical properties.* C_p and ρ are constant except for the deliberate heat-capacity error and the steam correlations. Over a 40 to 95 °C span in liquid water this is accurate to a few percent. It fails for a fluid near a phase change or with a strongly temperature dependent viscosity, where the film coefficient would also move with temperature and not only with flow.

9. *Valve and instrument idealization.* FCV-102 has a first-order actuator lag, an installed characteristic x^a and an optional pure deadband for stiction [17]. Positioner dynamics, hysteresis separate from deadband, and a finite stroking rate are absent. TT-102 and FT-102 are first-order lags with optional bias and noise, so drift, quantization and thermowell fouling are not represented.

10. *Omitted entirely.* Ambient heat loss, wall thermal expansion, the process-side pump characteristic, downstream mixing volumes, tube vibration and mechanical failure other than stiction and fouling are all absent. They are omitted because none of them changes the structure of the loop the application is used to teach.

4.4 Identification from Step-Response Data

No element of the model above is first order, and the only explicit dead time in the path from valve to transmitter is the 0.30 min transport delay of (4.11). A small step in the valve nevertheless produces a response that reads as FOPDT, the model of Equation (1.1), as shown in Equation (4.12).

$$\hat{G}(s) = \frac{\hat{K}_p \, \mathrm{e}^{-\hat{\theta} s}}{\hat{\tau} s + 1} \tag{4.12}$$

where $\hat{K}_p$ is the identified gain in °C/%, $\hat{\tau}$ is the identified time constant and $\hat{\theta}$ is the identified dead time, both in minutes. The identified constants are lumped surrogates. The value of $\hat{\tau}$ combines the fluid and metal capacitances with the shell and valve dynamics, and $\hat{\theta}$ combines the true transport delay (4.11) with the valve stroke, the metal lag, the transmitter lag and the initial curvature of a higher-order response, the effective dead time that Section 1.4.4 describes. At the design point a 5 % step, fitted over the 20 min that follow it, yields $\hat{K}_p = 0.355\,^{\circ}\mathrm{C}/\%$, $\hat{\tau} = 3.02$ min and $\hat{\theta} = 0.94$ min at $R^2 = 0.9997$, against a true transport delay of 0.30 min. The scenarios that need a model preload this one with $\hat{\tau}$ rounded to 3.0 min.

Those three numbers are not properties of the exchanger. The same test repeated from the balancing valve position at 24 000 kg/h returns $\hat{K}_p = 0.22\,^{\circ}\mathrm{C}/\%$, $\hat{\tau} = 2.49$ min and $\hat{\theta} = 0.82$ min, and at 16 000 kg/h it returns $\hat{K}_p = 0.56\,^{\circ}\mathrm{C}/\%$, $\hat{\tau} = 3.82$ min and $\hat{\theta} = 1.10$ min. The gain varies by a factor of 2.6 across a feed range of 16 000 to 24 000 kg/h. Three mechanisms act together. More mass to heat per minute at the same duty is a smaller temperature rise per unit of steam, the log-mean driving force in (4.3) shifts as the outlet moves relative to the wall, and the valve runs further up its travel where the shell has pressured up, so $P_{\mathrm{sup}} - P_{\mathrm{sh}}$ shrinks and the valve contributes less loop gain through (4.9). The shell pressure drop falls from 3.45 bar at 30 % lift to 0.73 bar at full lift. Volume 1, Chapter 2 meets the same lesson on a tank whose fitted gain and time constant move with the operating level.

Both a graphical reading of the reaction curve and a least-squares fit are available for the same record. The fitter locates the largest single-sample jump in the logged valve command, takes a pre-step baseline from the twelve preceding samples, truncates the window at the next command move, and then exploits the fact that (4.12) is linear in $\hat{K}_p$ for any fixed pair $(\hat{\tau}, \hat{\theta})$. A one-parameter least squares gives the optimal gain in closed form for each candidate pair, so only a two-dimensional grid of 4200 candidates plus one local refinement of 2500 candidates is searched. The fit is reported with R^2 and RMSE alongside the manually read constants, so the difference between a graphical reading and a regression is quantified without a numerical library. The fitter returns nothing when the logged step is smaller than 0.5 % or the response window is shorter than 1.5 min.

A separate steady-state regression addresses the nonlinearity itself. Recording a sequence of settled pairs (u, T_{out}) and fitting $T_{\mathrm{out}} = b_0 + b_1 u$ gives R^2 above 0.99 over a narrow span and a poorer

fit over a wide one, because the steady-state map runs from 76.8 °C at 30 % lift to 93.9 °C at full lift while the local slope falls from 0.75 °C/% at 30 % lift to 0.06 °C/% at full lift. Up to twelve points are retained. The regression shows that a linear model identified at one operating point describes the plant only near that point, which is the argument for gain scheduling [18] in Section 1.4.13.

4.5 Controller Design

4.5.1 Velocity Form PID Algorithm

Both loops run the velocity-form PID algorithm [18] of Section 1.6.2, the one implementation that the framework apps share. It is written here as increments per scan in the symbols of this loop, because the results below cite its terms. Error and the three increments are given by Equations (4.13) and (4.14).

$$e_k = \mathrm{SP}_k - \mathrm{PV}_k \tag{4.13}$$

$$\begin{aligned} \Delta u_k^P &= K_c\,(e_k - e_{k-1}) \\ \Delta u_k^I &= \frac{K_c}{\tau_I}\, e_k\, \Delta t_c \\ \Delta u_k^D &= -\,K_c \tau_D\, \frac{\mathrm{PV}_k^f - 2\mathrm{PV}_{k-1}^f + \mathrm{PV}_{k-2}^f}{\Delta t_c} \end{aligned} \tag{4.14}$$

where K_c is in %/°C on the temperature loop, τ_I and τ_D are in minutes, and Δt_c is the control period. Derivative comes from a second difference on a filtered measurement, with $\mathrm{PV}_k^f = \alpha \mathrm{PV}_{k-1}^f + (1-\alpha)\mathrm{PV}_k$ and $\alpha = \exp(-\Delta t_c/\tau_f)$ and $\tau_f = 0.05$ min by default, which is stable at any sample period because α is never negative. Section 1.4.8 gives what this filter costs in phase. The increments are combined, the demand is clipped to the output limits and the integral is corrected by back-calculation, as shown in Equation (4.15).

$$\begin{aligned} z_k &= z_{k-1} + \Delta u_k^P + \Delta u_k^D \\ c_k &= z_k + I_{k-1} \\ u_k &= \min(u_{\max},\ \max(u_{\min},\ c_k)) \\ I_k &= I_{k-1} + \Delta u_k^I + \frac{\Delta t_c}{T_t}\,(u_k - c_k) \end{aligned} \tag{4.15}$$

Here z_k accumulates the proportional and derivative increments without limit, c_k is the demanded position built on the integral accumulated through the previous scan, and I_k also carries the starting output. These are the sum S_k, the output before limiting v_k and the integral with its bias of Equations (1.24) to (1.26). The last term drives I_k toward the value that reproduces the clipped output with tracking time $T_t = \tau_I$ ($\tau_{I,F}$ on the flow loop), removing $1/150$ of the clipped excess per scan at $\tau_I = 3$ min, so only the integral is corrected. Away from the limits the recursion is algebraically identical to the positional law with the same three gains, but not at a limit, where the integral unwinds over a time of order τ_I, the relaxation of Equation (1.18). Disabling anti-windup sets $T_t = 10^9$ min, so the integral winds up against a saturated output and reset windup can be observed rather than described. In manual operation I_k is reset every scan so that c_k equals the output in force (the valve position, or the measured steam flow for the outer loop of a cascade), and in automatic operation the manual valve setting follows u_k after every controller execution, so transfer is bumpless in both directions.

Output limits are 0–100 % on the valve and on the steam-flow setpoint of the cascade and ±50 % on the feedforward trim, with no rate limit. The anti-windup of TIC-101 knows only its own limits. A valve held wide open by FIC-102 is not reported back to it, and its setpoint can reach 100 % of the 4000 kg/h FT-102 span while the header delivers at most 2600 kg/h (65 % of span), so the outer integral can wind while the inner loop is saturated. The 0–100 % clip on feedforward plus trim has the same effect. Both loops are reverse acting in the ISA sense (Section 1.4.3), because opening the steam

valve raises the temperature, so an outlet temperature above setpoint has to close the valve. With the error written as SP − PV in (4.13), that is what a positive K_c delivers. The controller executes every 0.02 min (1.2 s), independent of the 0.005 min (0.3 s) physics step, which makes the measurement a sampled one and places a half-sample of additional lag in the loop (Section 1.4.11).

4.5.2 Three Architectures, One Actuator

The manipulated variable is the same in all four architectures, the three PID structures described here and the predictive controller of Section 4.5.4. What changes is the calculation that produces it. In the single loop, TIC-101 drives FCV-102 directly. In cascade, TIC-101 outputs a steam-flow setpoint in percent of FT-102 span and FIC-102 drives the valve with $K_{c,F} = 2\,\%/\%$ and $\tau_{I,F} = 0.25\,\text{min}$, which is about the valve stroking time. The inner loop absorbs header-pressure upsets and the valve characteristic before they reach the temperature. The separation of time scales required for cascade control is present, because the flow loop settles in a fraction of a minute against an outer time constant of 3.0 min, a ratio of about ten [1]. Volume 3, Chapter 3 measures what a cascade loses when that separation is absent. In the third architecture the steam demand is computed from the process heat balance, as shown in Equation (4.16).

$$F_{\text{ff}} = K_{\text{FF}}\,\frac{\dot{m}_p C_p\,(T_{\text{SP}} - T_{\text{in}})}{\lambda_0}, \qquad \lambda_0 = \lambda\,(2.60\,\text{bar}) \tag{4.16}$$

where K_{FF} is a dimensionless trim on the calculation and F_{ff} is converted to percent of FT-102 span. The result passes through a backward-Euler lead-lag with constants τ_{lead} and τ_{lag} and a pure delay θ_{FF}, and it is added to a temperature trim bounded to $\pm 50\,\%$ of span before FIC-102 moves the valve. Equation (4.16) is deliberately built from the nameplate C_p rather than the true one, so a heat-capacity error biases the feedforward calculation and the feedback trim is required rather than optional. Whichever structure is selected, the idle loops keep tracking, so the architecture can be changed mid-run without a bump. This mirrors the compound feedforward-plus-cascade practice reported for industrial exchanger services [8].

Pause. Try Exercise 4.1 before reading on.

4.5.3 Tuning Rules

Two rules are offered, both operating on the identified model and never on the simulator equations. Section 1.4.5 derives both and compares them on one process. The IMC-derived SIMC PI rule [12, 13] is shown in Equation (4.17).

$$K_c = \frac{\hat{\tau}}{\hat{K}_p(\tau_c + \hat{\theta})}, \qquad \tau_I = \min\!\left(\hat{\tau},\, 4(\tau_c + \hat{\theta})\right), \qquad \tau_D = 0 \tag{4.17}$$

where the closed-loop constant τ_c is selected as $2\hat{\theta}$, $\hat{\theta}$ or $0.5\hat{\theta}$ for conservative, moderate and aggressive response. The reaction-curve PID rule [11] is shown in Equation (4.18).

$$K_c = \frac{1.2\,\hat{\tau}}{\hat{K}_p\,\hat{\theta}}, \qquad \tau_I = 2\hat{\theta}, \qquad \tau_D = 0.5\hat{\theta} \tag{4.18}$$

Both results are rounded to the tuning-entry resolution and clamped to $K_c \le 40\,\%/^{\circ}\text{C}$, $0.1\,\text{min} \le \tau_I \le 30\,\text{min}$ and $\tau_D \le 3\,\text{min}$ before they are loaded. With the application's τ_c tied to $\hat{\theta}$, both rules' gains scale inversely with the identified dead time: changing 0.94 min to 0.50 min multiplies each by 1.88. Equation (4.18) places $\hat{\theta}$ in a denominator with no additive term, so a dead time read as 0.50 min instead of 0.94 min produces $K_c = 20.3\,\%/^{\circ}\text{C}$ on the preloaded model ($\hat{K}_p = 0.355\,^{\circ}\text{C}/\%$, $\hat{\tau} = 3.0\,\text{min}$), against 10.8 %/°C from the same rule on the correctly read dead time and 4.5 %/°C from the SIMC rule on the same model. The application ships that reading error as a scenario that preloads exactly the 20.3 %/°C the rule returns, and every measurement reported below is taken at that value.

> **TRY IT IN THE APP** **8 · Ziegler-Nichols, and an Optimistic Dead Time**
>
> Load the case, which preloads $K_c = 20.3\,\%/^\circ\mathrm{C}$ from a dead time read as 0.50 min, and run it to 24 min. Watch the steam valve trend after the feed turns 8 K colder at 4 min: it rings with a period of 4.3 min, each swing about 0.54 of the one before. Then Reset, enter $\hat{\theta} = 0.94\,\mathrm{min}$, apply the reaction-curve rule before 4 min and run again.

4.5.4 Model Predictive Control on the Identified Model

The fourth architecture replaces TIC-101 with a model predictive controller (MPC) that moves FCV-102 directly. Every existing case keeps its shipped architecture, and only the two cases added with the MPC select it, the comparison case of Section 4.7.2 and the tube case of Section 4.7.3. Every $T_s = 0.1\,\mathrm{min}$, or five controller executions, the MPC predicts TT-102 over a horizon, chooses a sequence of valve positions, applies the first and holds it until the next decision, the receding-horizon scheme of Section 1.6.3 [19, 20]. Its model joins the identified valve path of Equation (4.12) to a measured feed-flow path in deviation variables, as shown in Equation (4.19).

$$T_{\mathrm{out}} - T_0 = \frac{\hat{K}_p\, \mathrm{e}^{-\hat{\theta} s}}{\hat{\tau} s + 1}\,(u - u_0 + d) + \frac{K_w\, \mathrm{e}^{-\theta_w s}}{\tau_w s + 1}\, w \tag{4.19}$$

where u_0, T_0 and F_0 are the valve command, the TT-102 reading and the FT-101 reading when the estimator starts, $w = F_p - F_0$ is the measured feed deviation in kg/h, d is an unmeasured disturbance in percent of valve travel, and $K_w = -1.275 \times 10^{-3}\,^\circ\mathrm{C\,h/kg}$, $\tau_w = 3.08\,\mathrm{min}$ and $\theta_w = 0.60\,\mathrm{min}$ are the gain, lag and dead time of the feed path.

The valve path is the model entered for the tuning rules, so an identification error reaches the MPC as it reaches the PI, and while no complete model is entered the MPC makes no plan and TIC-101 keeps the valve. In the comparison case it is the preloaded fit of Section 4.4, 0.355 °C/%, 3.0 min and 0.94 min. The feed path was identified once on the simulator at the design point, with the valve held at 44.9 % and no measurement noise, by averaging the responses to feed steps of $\pm 1000\,\mathrm{kg/h}$ (root-mean-square residual 0.32 % of the response), and it is declared as constants, as the nameplate data of Equation (4.16) are. Its gain matches the steady-state slope of the process model at a fixed valve, -1.2735×10^{-3} °C h/kg, which is smaller in magnitude than the -2.25×10^{-3} °C h/kg of a fixed duty because more feed also raises the film coefficient of Equation (4.10). Both dead times are kept as records of what has already been sent, $D_u = 47$ and $D_w = 30$ executions of $\Delta t_c = 0.02\,\mathrm{min}$, and each plan starts from the state that the commands already in transit will produce, the prediction of a Smith predictor, which is exact for the model.

The disturbance d comes from a steady-state Kalman filter built on the two paths, sampled at Δt_c for a zero-order hold and augmented with d as a random walk, the usual construction for offset-free predictive control [21]. The filter runs at every execution whichever architecture holds the valve, as shown in Equation (4.20).

$$\begin{aligned}
\bar{x}_{u,k} &= \mu_u \hat{x}_{u,k-1} + \hat{K}_p(1-\mu_u)\left(u_{k-1-D_u} - u_0 + \hat{d}_{k-1}\right)\\
\bar{x}_{w,k} &= \mu_w \hat{x}_{w,k-1} + K_w(1-\mu_w)\, w_{k-1-D_w}\\
\nu_k &= T_{\mathrm{out},k} - T_0 - \bar{x}_{u,k} - \bar{x}_{w,k}\\
\hat{x}_{u,k} &= \bar{x}_{u,k} + \ell_u \nu_k, \qquad \hat{x}_{w,k} = \bar{x}_{w,k} + \ell_w \nu_k\\
\hat{d}_k &= \hat{d}_{k-1} + \ell_d \nu_k
\end{aligned} \tag{4.20}$$

where x_u and x_w are the outputs of the two paths in °C, $\mu_u = \mathrm{e}^{-\Delta t_c/\hat{\tau}} = 0.99336$ and $\mu_w = \mathrm{e}^{-\Delta t_c/\tau_w} = 0.99353$, u_{k-1-D_u} and w_{k-1-D_w} are the command and the feed reading D_u and D_w executions before the period just ended, and ν_k is the innovation. The gains $\ell_u = 0.1033$, $\ell_w = 8.4 \times 10^{-7}$ and $\ell_d = 2.667\,\%/^\circ\mathrm{C}$ follow from a measurement variance r, state noise variances of $10^{-6} r$ and a disturbance noise variance $q_d = r/\hat{K}_p^2$, with $r = 1\,^\circ\mathrm{C}^2$ because only the ratios set the gains. Every

load the model lacks, a sagging header, a colder feed, fouling, a failing trap or an error in the model itself, appears in $\hat{d}$ as the valve offset that would produce it, and holding $\hat{d}$ over the horizon removes the offset, so the estimate supplies the integral action. The comparison case has no unmeasured load, so the $\hat{d}$ of −8.28 % of valve reached at its end measures only how far the linear model, identified at 85 °C and 20 000 kg/h, misses the exchanger at 88 °C and 24 000 kg/h.

Each decision solves a quadratic program on the same model resampled at T_s, as shown in Equation (4.21).

$$\begin{aligned} \min_{v,\varepsilon} \; & \frac{T_s}{\hat{\tau}} \sum_{k=1}^{N} \left(\frac{\hat{T}_k - T_{\mathrm{SP}}}{1\,^{\circ}\mathrm{C}} \right)^2 + \kappa \sum_{j=1}^{12} \left(\frac{v_j - v_{j-1}}{5\,\%} \right)^2 + c_\varepsilon \varepsilon^2 \\ \text{s.t. } & 0 \le v_j \le 100\,\%, \qquad \varepsilon \ge 0 \\ & T_{\mathrm{lo}} - \varepsilon \le \hat{T}_k \le T_{\mathrm{hi}} + \varepsilon, \quad k = 1, 3, \ldots, N-1 \end{aligned} \tag{4.21}$$

where $\hat{T}_k$ is the TT-102 reading predicted k steps beyond the dead time with $\hat{d}$ held, $v_1, \ldots, v_{12}$ are valve positions held over twelve blocks whose lengths in steps are $(1,1,1,1,2,2,3,4,6,9,15,75)$, $N = 120$ steps covers 12 min after $\hat{\theta}$, v_0 is the command applied now, $\kappa = 1.5$ is a dimensionless move suppression, ε is one slack in °C priced at $c_\varepsilon = 10^3$ per °C^2, and $T_{\mathrm{lo}} = \min(79\,^{\circ}\mathrm{C}, T_{\mathrm{SP}} - 1\,^{\circ}\mathrm{C})$ and $T_{\mathrm{hi}} = \max(90\,^{\circ}\mathrm{C}, T_{\mathrm{SP}} + 1\,^{\circ}\mathrm{C})$ are TAL-102 and TAH-102 moved, where needed, to 1 °C from the setpoint. The factor $T_s/\hat{\tau}$ keeps the error sum from growing with the horizon, and with these scales $\kappa = 1$ prices one 5 % valve move like a 1 °C error held for one time constant. The valve range is the only hard constraint. The band is soft, so a plan always exists, and moving it off the setpoint keeps a 90 °C setpoint from sitting on TAH-102. There is no rate limit and no terminal weight, because the model is stable and $\hat{\theta} + NT_s = 12.94$ min exceeds four time constants.

> ⧖ HISTORY **Predictive Control Moves Into the Plant, 1973 to 1983**
>
> Control by a sequence of open-loop optimizations was described in the 1960s, but industry put it to work. Engineers at an oil company applied their own predictive method in 1973 and published a version built on a step-response model in 1979. A paper of 1976 described the first applications, among them a power-plant steam generator whose response time varied inversely with load, handled by running the controller with a variable sample time. These controllers penalized valve moves for a less aggressive response and some robustness to model error, and one showed a better feedforward response to inlet temperature than a PID lead-lag. In 1983 the method was posed as a quadratic program with the constraints written explicitly [20]. The predictive controller of this chapter is of that second kind.

With the preview on, the feed enters every prediction as a known input, the FT-101 readings taken θ_w before each period while the record reaches them and the present reading held thereafter, a measured disturbance rather than a forecast. With the preview off the reading is frozen and a feed change reaches the MPC only through $\hat{d}$. The feedforward of Equation (4.16) also reads TT-101, so it receives more measured information than the MPC. The dual active-set solver of the framework [22] solves the program, 13 variables and 145 inequality rows, with a ridge near 10^{-9} on the Hessian diagonal.

The move suppression and the disturbance noise were fixed once, on the comparison case of Section 4.7.2, by the rule applied to the predictive controllers of the apps. Of 55 settings (κ from 0.05 to 5, a factor of 0.01 to 100 on q_d), 25 reject no plan and move the valve no more than the IMC-PI (45.6 %), and among those within 10 % of the best true-temperature IAE (12.71 °C min) the largest κ and then the smallest factor are taken, $\kappa = 1.5$ with a factor of 1 (13.78 °C min). Decision periods of one, two and five executions give true-temperature IAE of 14.29, 13.83 and 13.78 °C min under the same rule, at about 370, 190 and 85 µs per plan in Node.js, and horizons from 6 to 30 min move the scorecard IAE only from 14.12 to 14.00 °C min.

A plan is discarded if the solver cannot certify it as optimal, or if its first move is non-finite or outside 0–100 %. TIC-101 tracks the applied command at every execution and therefore takes over

the valve bumplessly (0.026 % when a rejection is forced 8 min into the comparison case), the record is marked, and control returns to the MPC at the next acceptable plan. Loading a case solves the first plan. The MPC reads only TT-102, the command applied, FT-101, the setpoint, the alarm limits and its own settings, and its model is the learner's entry and the declared feed path, so a change to a plant parameter or a disturbance setting is model mismatch for it, as it would be on a real exchanger.

4.5.5 Alarm Band Tightened by the Learner's Fit

The plan of Equation (4.21) treats the entered model as exact, so near TAH-102 it stakes the band on the step test behind it. A second setting of the MPC first measures how far the entered model misses that step test and then keeps the reading inside the band for any valve-path gain within that misfit. This is constraint tightening, in which one nominal plan is kept and its constraints are moved in by a bound on the model's error [23, 24], rather than a plan made against every model of a set [25].

In manual operation the controller records the command it applied and the TT-102 reading at each execution, finds the step test in that record by the rules of the fitter of Section 4.4 (the largest move of at least 0.5 %, a twelve-execution baseline and at least 1.5 min of response), and replays the entered model through it, as shown in Equation (4.22).

$$\begin{aligned} \delta^2 &= \frac{\sum_i \left(T_{\mathrm{m},i} - \bar{T}_\mathrm{b} - g_i\right)^2}{\sum_i g_i^2} \\ g_i &= \hat{K}_p \Delta u_\mathrm{s} \left(1 - \mathrm{e}^{-(t_i - t_\mathrm{s} - \hat{\theta})/\hat{\tau}}\right) \end{aligned} \tag{4.22}$$

where t_s and Δu_s are the time and size of the step, $\bar{T}_\mathrm{b}$ is the mean baseline reading, $T_{\mathrm{m},i}$ is the reading at execution i of the response, and $g_i = 0$ while $t_i - t_\mathrm{s} \le \hat{\theta}$. The relative misfit δ measures the learner's model against the learner's own data, not against the hidden constants, and equals $|K_p/\hat{K}_p - 1|$ for a gain error alone on a linear plant. The step test of the tube case fits as 0.3596 °C/%, 3.14 min and 0.92 min, and with $\hat{\tau} = 3.0$ min and $\hat{\theta} = 0.94$ min the entries 0.36, 0.54 and 0.252 °C/% give δ of 1.77, 34.30 and 40.82 %, against 33.3 and 42.9 % for a pure gain error of 1.5 and 0.7 times.

Read as the relative error of the valve path, δ places the reading at prediction step k in $\hat{T}_k \pm h_k$ with $h_k = \delta|\tilde{x}_{u,k} - \hat{x}_u|$, where $\tilde{x}_{u,k}$ is the valve-path output that the plan predicts and $\hat{x}_u$ is its estimate from Equation (4.20) at the decision. The half-width is zero where the plan leaves the valve path alone. It grows with the commands still in the dead-time record, sent but not yet seen at TT-102, and with the plan's own moves, to δ times the whole change at the end. Keeping both edges in the band gives linear rows, as shown in Equation (4.23).

$$\begin{aligned} T_\mathrm{lo} - \varepsilon_1 &\le \hat{T}_k + \delta\left(\tilde{x}_{u,k} - \hat{x}_u\right) \le T_\mathrm{hi} + \varepsilon_1 \\ T_\mathrm{lo} - \varepsilon_2 &\le \hat{T}_k - \delta\left(\tilde{x}_{u,k} - \hat{x}_u\right) \le T_\mathrm{hi} + \varepsilon_2 \end{aligned} \tag{4.23}$$

for $k = 1, 3, \ldots, N-1$. They replace the band of Equation (4.21), with $c_\varepsilon(\varepsilon_1^2 + \varepsilon_2^2)$ in place of $c_\varepsilon \varepsilon^2$, each as a further output $(1 \pm \delta)x_u + x_w$ of the same model against the band shifted by $\pm\delta\hat{x}_u$. Everything else is the nominal plan's, and the tube has no weight of its own. One slack per edge for the whole horizon was kept by a rule fixed beforehand, because per-step slack gave the same minutes past the band at 1.5 to 2.1 ms a plan. The program grows from 13 variables and 145 rows to 14 and 266, and a plan takes 104 to 106 µs in Node.js (86 to 88 µs for the nominal band) and 137 to 172 µs on the frame loop of the application in Chrome, or 0.17 % of one core in real time. No step test, or a misfit of 100 % or more, leaves the nominal band, because $1 \pm \delta$ then admits a valve that does nothing or acts in reverse, and from 50 % the controller's note warns that the tube will work the valve hard. The plan display shades the tube, the misfit and the end width are reported and logged, and the misfit is computed from the controller's own record and the entered model alone.

The interval is exact for a gain error on the linear model only once the estimator has settled. Around a linear plant of the model's structure with two thirds of the entered gain ($\delta = 1/3$), each

plan applied open loop to a copy of the plant leaves 4790 of 18 060 judged predictions outside its tube, by up to 0.32 °C and none before the setpoint step, because while $\hat{d}$ still absorbs the gain error the model predicts a drift of x_u that is wrong in full and the tube covers only δ of it.

4.6 Dynamic Analysis

Three properties of the loop follow from the model and govern how it must be tuned.

The first is a dead time that moves with production rate. By (4.11), a higher throughput shortens θ, from 0.375 min at 16 000 kg/h to 0.250 min at 24 000 kg/h. A controller tuned at low rate is therefore conservative at high rate, which preserves stability margin, while a controller tuned at high rate and operated at low rate loses it.

The second is a gain that falls as the load rises. The log-mean driving force and the valve self-throttling act in the same direction, so the steady-state gain falls from 0.561 °C/% at 16 000 kg/h to 0.219 °C/% at 24 000 kg/h and to 0.147 °C/% at 27 000 kg/h, a factor of 3.8 across the range. Combined with the moving dead time, the loop gain of a fixed controller varies by more than a factor of three over the normal operating window, which is the condition that gain scheduling addresses [18]. Chapter 6 builds such a schedule for a loop whose gain moves by orders of magnitude.

The third is a capacity limit that presents as a tuning problem. When required duty approaches available duty the valve saturates at 100 % and the temperature falls short of setpoint. The symptom is a steady offset with a saturated output, which is also the symptom of a wound-up integrator, and the common response is to increase gain. Required and available duty are reported as separate trends together with the percentage of capacity in use, so the two causes can be distinguished from the record. At 27 000 kg/h with the exchanger 70 % fouled and the header at 3.2 barg, the required duty is 1411 kW against 1248 kW available, and the outlet settles at 79.8 °C with the valve wide open. Chapter 3 asks the same question, whether the setpoint is reachable at all, of a composition loop.

Two further behaviors are diagnostic problems rather than tuning problems. Fouling raises R_f slowly, the temperature stays at setpoint, and the controller output drifts upward, so the controller output rather than the controlled variable is the fouling indicator. A ramp to 100 % fouling moves the balancing valve position from 44.9 % to 71.9 % and drops the overall conductance from 16.78 kW/K to 13.30 kW/K while the outlet holds 85 °C. Valve stiction is modeled as a deadband S, where the stem is held while $|u - x| < S$ and, once moving, continues until the command is within $0.2S$, which converts a smooth controller output into the square-wave stem motion characteristic of a sticking valve [17]. The resulting limit cycle is not removable by tuning.

Pause. Try Exercise 4.3 before reading on.

> **TRY IT IN THE APP** **14 · Fouling**
>
> Load the case and press Run. The tubes foul on a ramp while the loop stays in automatic. Watch two trends: the outlet temperature stays at its setpoint, and the steam valve creeps open. The valve, not the temperature, carries the news of the fouling.

4.7 Results

Nineteen scenarios plus a graded challenge are arranged as a laboratory sequence, covering manual operation and finding the operating point, a step test and a regression comparison, a first conservative automatic loop and then a model-based one, a setpoint move and an aggressive-rule comparison, the three load disturbances, cascade against the header upset and feedforward against measured load changes, four diagnostics, a comparison of three controllers designed from one identified model, and the alarm band of that predictive controller set from the learner's own fit. Each scenario sets an initial mode, a tuning, a set of disturbance values and a list of timed events. The disturbances and faults are summarized in Table 4.5, and the three load disturbances are applied at $t = 4$ min from a settled start.

Table 4.3 Scenarios in Selector Order and Completion Requirements, First Half

	Scenario	Required to complete
1	Hand operation	From the steady 30 % start, move the valve after 1 min and hold the outlet within ±1 °C of 85 °C by hand for 3 min, then set the heat-balance position.
2	Operating point	Settle within ±0.5 °C of 85 °C for 2 min in Manual, record five or more steady points spanning at least 15 % of valve travel, then a sixth below 32 % or above 62 %.
3	Step test	Step the valve 5 % from the settled design point after 1 min, hold it 15 min, enter $\hat{K}_p$, $\hat{\tau}$ and $\hat{\theta}$ within 0.28–0.43 °C/%, 2.4–3.8 min and 0.7–1.2 min, then run the least-squares fit.
4	Reading against fit	Fit a 5 % step and then a step of at least 9 %, each over at least 10 min, and load the fit.
5	First try in Auto	Recover the cold start to 84.8 °C with $K_c \leq 1.5\,\%/°\mathrm{C}$, then Reset, select the moderate SIMC rule within 0.5 min and recover again with $K_c \geq 3\,\%/°\mathrm{C}$.
6	SIMC tuning	Switch to Manual and back to Auto after 0.5 min, hold a setpoint of 87 °C for 8 min with $\tau_I \geq 2.9$ min, then reload and repeat the step from 85 °C for 10 min with $\tau_I \leq 0.85$ min.
7	Setpoint test	Run the 85 to 90 °C step at 3 min to $t = 15$ min with $K_c \geq 4\,\%/°\mathrm{C}$, then Reset, select the conservative SIMC rule (K_c near 3.0 %/°C) before the step and repeat to 15 min.
8	ZN, dead time	Run the 8 K inlet drop at 4 min to $t = 24$ min with $K_c \geq 19\,\%/°\mathrm{C}$, then Reset, enter $\hat{\theta} = 0.94$ min and apply the ZN rule before 4 min (K_c of 9 to 13 %/°C) and repeat to 24 min.
9	Feed-flow upset	After the step to 24 000 kg/h at 4 min, hold within ±0.3 °C for 2 min, then fit a Manual step of 4 % or more over 5 min.
10	Inlet upset	After the 8 K inlet drop at 4 min, hold the outlet within ±0.2 °C for 2 min in Auto.

Tables 4.3 and 4.4 list the cases in selector order with the requirement for completing each, judged as Section 1.6.4 describes. Reset keeps the learner's own settings and a reload restores the shipped ones (Section 1.6.1). A setting chosen before Run is part of the starting state and gives the same run as the same setting carried through a Reset, so a step the learner is to watch is made after the run has started. The challenge is judged on the case as given, so changing a model parameter or a disturbance setting, or firing an extra upset, voids the attempt.

Each load disturbance has a computable magnitude. A feed step from 20 000 to 24 000 kg/h raises the required duty by 209 kW and moves the balancing valve from 44.9 % to 59.4 %. An inlet drop of 8 K raises the required duty by 186 kW and the steady steam flow by 329 kg/h, because the latent heat falls as the shell pressure rises. A header pressure drop from 4.0 to 3.2 barg at a fixed valve position cuts the steam flow from 1727 to 1594 kg/h, a fall of 7.7 %, and would settle the outlet at 81.9 °C if the controller did not act.

4.7.1 Measured Comparison of Two Tunings

Table 4.6 reports the application's own performance measures for three controllers subjected to the identical upset, a feed inlet temperature stepped from 40 °C to 32 °C at $t = 4$ min, read at $t = 10$ min. All three were produced by a rule applied to the same identified model. The outer columns are the shipped tunings of the inlet-temperature and tuning-comparison scenarios, and the middle column is the reaction-curve rule reapplied after the fitted dead time is entered, which is the second run the tuning-comparison scenario asks for.

Both reaction-curve controllers beat SIMC on every error measure, with IAE of 3.37 and 3.43 °C min against 8.61 °C min. That advantage follows from the rule, which asks for more gain and adds derivative action, and not from the misread dead time. Reading the dead time 0.44 min short inflates $1.2\hat{\tau}/(\hat{K}_p\hat{\theta})$ by a factor of 1.88, trims the largest deviation by 0.12 °C, leaves IAE slightly worse and multiplies the valve travel by 4.5. The loop rings with a period of 4.3 min, each valve swing about 0.54 of the one before, so by $t = 24$ min it has accumulated an IAE of 6.05 °C min and 187 % of travel against 3.67 °C min and 25.8 % for the correctly read dead time. The remaining margin does not survive a change of operating point. At a feed of 18 000 kg/h, with a higher process gain and a longer dead time, the same upset grows into an oscillation that strokes the valve between 14 % and 100 %, while the PID tuned on the fitted dead time remains stable at 16 000 kg/h. The consequential

Table 4.4 Scenarios in Selector Order and Completion Requirements, Second Half

	Scenario	Required to complete
11	Header sag	After the header falls to 3.2 barg at 4 min, hold the outlet within $\pm 0.2\,^{\circ}\mathrm{C}$ for 2 min with the single loop.
12	Cascade	Run the same sag in cascade and switch to the single loop after 8 min, then reload the case, set $K_{c,F} \leq 0.35\,\%/\%$ before the sag and run in cascade to $t = 12$ min.
13	Feedforward	Run both load changes for 29 min in feedforward with $K_{\mathrm{FF}} = 1$ and no lead, Reset and repeat in cascade, reload and repeat with $K_{\mathrm{FF}} \geq 1.25$, then set a lead of at least 0.35 min.
14	Fouling	Run the 2.5 %/min fouling ramp in Auto to $t = 40$ min, then set fouling to at least 145 % and run until the valve saturates.
15	Stiction	Run 40 min with the 3 % deadband, 10 min with $K_c \leq 2.3\,\%/^{\circ}\mathrm{C}$ and $\tau_I \geq 5.9$ min, 10 min in cascade, then remove it.
16	Instruments	With 0.15 °C noise and a 2 °C bias, run 20 min with $\tau_D = 0.47$ min, Reset and run 20 min with $\tau_D = 0$, reload and run 20 min with $\tau_f \geq 0.25$ min, then set the bias to −1.5 °C or below.
17	Capacity limit	Saturate the valve for 1 min and run past the cleaning at 20 min to 28 min, then Reset, switch anti-windup off at the start and run to 28 min.
18	Three controllers	Run the setpoint step to 88 °C at 3 min and the feed step to 24 000 kg/h at 15 min to $t = 30$ min under the MPC, then under the IMC-PI after a Reset, then under the feedforward with that PI as trim after a Reset.
19	Tube from the fit	Run the step test, the switch to Auto at 11 min, the setpoint step to 89 °C and the feed drop to 17 000 kg/h to $t = 40$ min with the tube, Reset with $\hat{K}_p \geq 0.5\,^{\circ}\mathrm{C}/\%$ and repeat, then Reset with the nominal band and repeat.
20	Challenge	Hold TT-102 within $\pm 1\,^{\circ}\mathrm{C}$ of 85 °C for the full 60 min through five load and supply upsets, with no alarm, no time saturated and less than 250 % valve travel, with the model and disturbances as given.[a]

Every case is an experiment except case 17, a demonstration, and case 20, the graded challenge. [a] With the shipped single-loop tuning ($K_c = 1\,\%/^{\circ}\mathrm{C}$, $\tau_I = 10$ min) the worst error is 4.58 °C, with 18 % valve travel and no saturation. With the suggested SIMC trim on the heat-balance feedforward ($K_c = 4.5\,\%/^{\circ}\mathrm{C}$, $\tau_I = 3$ min, $K_{\mathrm{FF}} = 1$) it is 0.66 °C, 0.34 °C inside the band, with 154 % travel and no saturation. The architecture alone is enough: feedforward with the shipped gains passes at 0.75 °C and 69 % travel.

Table 4.5 Load-Disturbance Scenarios and the Mechanism Each Exercises

Scenario	Change	What it exercises
Feed flow	20 000 → 24 000 kg/h	simultaneous change in gain, dead time and residence time
Inlet temperature	40 → 32 °C	duty step not visible at TT-102 for θ minutes
Steam header	4.0 → 3.2 barg	actuator-side upset detected first by the inner loop
Fouling ramp	R_f increasing	offset-free CV, drifting MV
Stiction	$S > 0$	limit cycle not removable by tuning
Capacity	duty demand > duty available	saturation, not detuning

error was in the identification, and the remaining difference, 24.8 % of valve travel against 17.4 % for SIMC, is a design choice about how much valve motion the service warrants. Figure 4.2 draws the table, and Figure 4.5 shows the misread case in the application at $t = 9$ min.

Performance is reported with the scorecard measures of Section 1.6.4 rather than one score. They are the IAE accumulated from the start of the run, the total valve travel, the fraction of time the output remained at a limit and, after a setpoint step only, percentage overshoot past the step target and settling time against a 1 °C band. The largest deviation is carried in the exported record. The trade is explicit on this service, because a tuning that reduces IAE on the header-pressure upset generally does so by increasing valve travel, which is packing and seat wear.

Table 4.6 Measured Performance After an 8 K Inlet-Temperature Drop, Read at $t = 10$ min. All Three Controllers Come From the Same $\hat{K}_p = 0.355$, $\hat{\tau} = 3.0$ min. Time Outside $\pm 1\,^{\circ}$C Is Taken From the Exported Record.

	SIMC PI $\hat{\theta} = 0.94$	ZN PID $\hat{\theta} = 0.94$	ZN PID $\hat{\theta} = 0.50$
K_c (%/°C)	4.5	10.8	20.3
τ_I (min)	3.0	1.88	1.0
τ_D (min)	0	0.47	0.25
IAE (°C min)	8.61	3.37	3.43
Max. deviation (°C)	2.16	1.36	1.24
Outside $\pm 1\,^{\circ}$C (min)	4.92	1.48	0.84
Valve travel (% of span)	17.4	24.8	112
Time saturated (%)	0	0	0

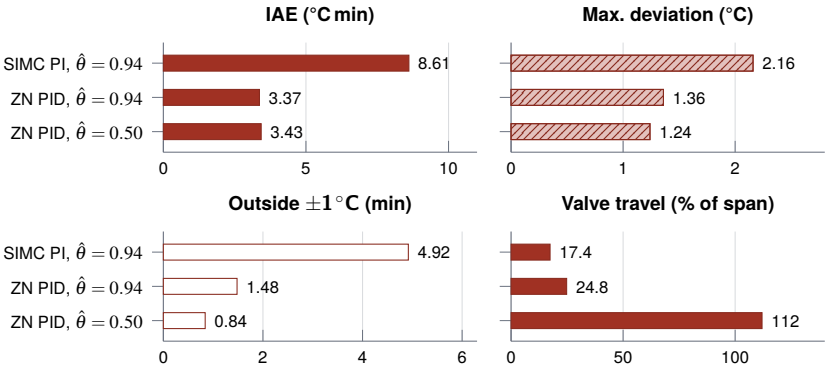


Figure 4.2 The three tunings of Table 4.6, all from the same identified model, after the 8 K inlet drop, read at $t = 10$ min: integrated absolute error and valve travel (solid bars), largest deviation (hatched) and time outside $\pm 1\,^{\circ}$C (open). Notice that the misread dead time leaves the error measures nearly where the correct reading puts them and multiplies the valve travel by 4.5.

> **+ DID YOU KNOW? A Bolder Tuning That Only Wears the Valve**
>
> From the simulator: after the 8 K inlet drop, the reaction-curve PID tuned on a dead time read as 0.50 min instead of 0.94 min reaches an IAE of 3.43 against 3.37 °C min at $t = 10$ min, while its valve travel rises from 24.8 to 112 % of span (Table 4.6).

4.7.2 Three Designs From One Identified Model

The three designs that one identification supports, the IMC-PI of Equation (4.17), the same PI as the trim of the heat-balance feedforward, and the MPC of Section 4.5.4, are compared in Table 4.7. Each case is loaded with its own seed and instruments and the structure is chosen before the run, so every design meets the same events. The PI keeps the tuning that the case ships, the feedforward rows use the default lag of 0.1 min and $K_{FF} = 1$ with that PI as the trim, while the MPC runs with the settings of Section 4.5.4 in every case, never retuned. The case added for the comparison preloads the 5 % fit and its SIMC tuning ($K_c = 4.5\,\%/^{\circ}$C, $\tau_I = 3.0$ min), raises the setpoint from 85 to 88 °C at 3 min and the feed from 20 000 to 24 000 kg/h at 15 min, and runs for 30 min.

On the setpoint half of the added case the model is the only information any design holds. The IMC-PI and the MPC stay close, with scorecard IAE of 7.17 and 8.84 °C min at 15 min and overshoots of 0.12 and 0.09 °C, while the feedforward carries the setpoint into its heat balance, opens the valve

Table 4.7 IMC-PI, Heat-Balance Feedforward and MPC From the Same Identified Model, With the Same Seed and Events in Each Case

Case	Design	IAE (°C min)	OS (°C)	e_{max} (°C)	$t_{\pm1}$ (min)	Travel (%)	Sat (%)	Plans
18, SP, feed	IMC-PI	20.64	0.12	1.98	8.34	45.6	0	—
	FF + FIC	17.21	2.34	0.99	4.72	230.6	0	—
	MPC	14.05	0.09	1.08	4.90	40.9	0	300
9, feed	IMC-PI	9.70	—	1.84	3.90	14.6	0	—
	FF + FIC	2.34	—	0.64	0	36.4	0	—
	MPC	2.51	—	0.86	0	19.3	0	300
	MPC[a]	5.53	—	1.69	2.58	17.9	0	300
7, SP	IMC-PI	13.22	0.10	—	3.56	38.1	0	—
	FF + FIC	18.30	2.43	—	6.30	142.5	24.3	—
	MPC	16.21	0.17	—	4.34	30.7	0	150
6, trap	IMC-PI	3.64	—	0.71	0	5.7	0	—
	FF + FIC	1.99	—	0.40	0	22.1	0	—
	MPC	2.22	—	0.66	0	7.9	0	300
8, $\hat{\theta} = 0.5$	ZN PID	6.05	—	1.24	0.86	186.8	0	—
	IMC-PI	6.25	—	1.81	2.60	28.2	0	—
	FF + FIC	12.72	—	1.11	1.98	315.8	0	—
	MPC	6.66	—	2.02	2.84	34.8	0	240
	IMC-PI[b]	11.78	—	2.16	4.90	17.7	0	—
	MPC[b]	7.19	—	1.99	3.30	23.1	0	240
7, SP[c]	IMC-PI	12.92	0.70	—	2.62	71.0	0	—
	MPC	14.51	0.31	—	3.62	43.9	0	150
17, capacity	IMC-PI	118.20	—	5.18	26.46	19.1	80.9	—
	MPC	114.22	—	5.18	21.90	25.9	74.3	280
14, fouling	Cascade	0.89	—	0.07	0	27.8	0	—
	MPC	7.94	—	0.39	0	23.2	0	400
19[d]	FF + FIC	8.40	—	0.66	0	153.7	0	—
	MPC	18.38	—	1.33	2.18	63.1	0	600

IAE and travel from the scorecard over 30, 30, 15, 30, 24, 15, 28, 40 and 60 min in case order. OS is the overshoot of TT-102 past a new setpoint, e_{max} its largest deviation after the load, $t_{\pm1}$ the time with the true product more than 1 °C off, Sat the share of time at a valve limit, Plans those of the MPC, none rejected. Case 6 adds a failing steam trap at 4 min, and in case 8 and row *c* every design comes from the misread fit. [a]Preview off. [b]Fitted $\hat{\theta} = 0.94$ min. [c]Designed from the 24 000 kg/h fit. [d]Challenge, suggested settings, for information.

by 28.0 % at the first execution after the step against 13.5 and 2.4 %, reaches 99.4 % at 4.64 min and overshoots by 2.34 °C. On the feed half the measurement decides. The PI waits for TT-102 and dips 1.98 °C, while the feedforward and the MPC act on FT-101 at once and dip 0.99 and 1.08 °C, the MPC with 22.6 % of valve travel after 15 min against 70.9 %, as shown in Figure 4.3. Over the case the MPC reaches 14.05 °C min against 20.64 and 17.21 °C min with the least travel of the three, and the steam used differs by less than 2 % (1015, 1032 and 1021 kg).

Rerunning the MPC on the added case with one setting changed at a time separates what the model contributes from what the optimization contributes. A move suppression of 0.5 gives an IAE of 11.72 °C min with 57.1 % of travel, and 10 gives 20.05 °C min with 33.5 %. Zero lets the valve chatter through 4214.5 % of travel although every plan stays optimal. The preview switched off raises the IAE from 14.05 to 17.13 °C min. A dead time entered as $\hat{\theta} = 0.5$ min gives the MPC 12.27 °C min with 56.5 % of travel, against 13.87 °C min with 82.0 % for the IMC-PI from the same reading.

The measured feed step of case 9 repeats the second half. The feedforward has both the lowest IAE, 2.34 against 2.51 °C min for the MPC and 9.70 for the PI, and the smaller peak, 0.64 against 0.86 °C, because its inner flow loop moves the steam within a fraction of a minute where the move suppression spreads the correction of the MPC. Without the preview the MPC reaches 5.53 °C min, so the preview accounts for 3.02 of its 7.19 °C min gain over the PI and the estimator for the rest. The failing trap enters on the steam side, where FT-102 at a fixed valve falls 54 kg/h within 0.2 min while the outlet has not yet moved, so the feedforward with its flow loop holds both the smallest peak (0.40 against 0.66 and 0.71 °C) and the lowest IAE (1.99 against 2.22 and 3.64 °C min).

The valve limit of case 17 is where the constraint enters the plan. Both designs hold the valve at

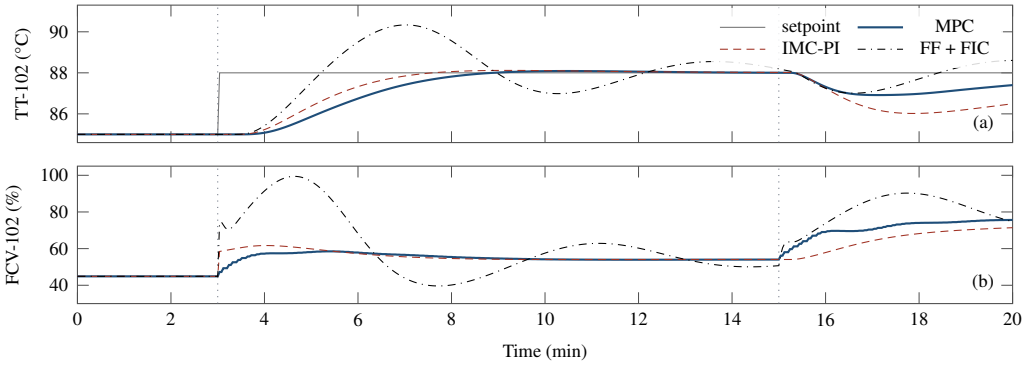


Figure 4.3 The added comparison case under the MPC (thick solid), the IMC-PI (dashed) and the heat-balance feedforward (dash-dotted), all three from the same identified model, to $t = 20$ min. The setpoint (thin gray) rises from 85 to 88 °C at 3 min and the feed from 20 000 to 24 000 kg/h at 15 min (dotted vertical lines). (a) TT-102 and setpoint. (b) Valve command. At $t = 20$ min the scorecard IAE is 12.72, 14.20 and 15.70 °C min and the valve travel 40.2, 41.8 and 215.1 %.

full lift from the start and spend the first 19.98 min outside the band, and 207 of the 280 plans of the MPC have the upper bound active on the first move, with up to 12 active-set iterations. After the cleaning at 20 min the MPC brings the valve off its stop 0.82 min after the event against 2.66 min for the PI, overshoots by 0.97 against 1.85 °C and spends 1.92 against 6.48 min outside the band, which is the whole of the difference in $t_{\pm 1}$. The PI unwinds its integral through the back-calculation of Equation (4.15) on a time scale of τ_I, while the MPC holds no integrator and plans again against the bound from a revised $\hat{d}$ at every decision.

The MPC loses the plain setpoint step of case 7, 16.21 against 13.22 °C min. Its first move is 3.98 % against the proportional kick of 22.50 %, a minute later the valve stands at 65.5 against 73.0 %, and TT-102 comes within 0.1 °C of 90 °C at 9.40 against 8.80 min, because the move suppression chosen to keep the travel of the MPC within that of the PI on the added case also slows its first moves. The model's gain is a smaller part. The steady-state balance gives a secant gain of 0.277 °C/% between 85 and 90 °C, and with only $\hat{K}_p$ changed to 0.28 °C/% the MPC reaches 15.35 and the IMC-PI designed from it 12.30 °C min, so the gap between them stays near 3 °C min. The 24 000 kg/h fit of Section 4.4 used on the same step improves both designs, and the PI stays ahead, 12.92 against 14.51 °C min, at the price of 0.70 against 0.31 °C of overshoot and 71.0 against 43.9 % of travel.

A dead time read as 0.5 min in case 8 helps the PI rules more than the MPC. The rules turn the short $\hat{\theta}$ into larger gains (20.3 and 8.5 %/°C for the reaction-curve PID and the IMC-PI, against 10.8 and 4.5 on the fitted dead time), which on this unmeasured inlet drop give IAE of 6.05 and 6.25 against 6.66 °C min for the MPC, although the reaction-curve PID moves the valve 186.8 % against 34.8 %. The Kalman gains do not depend on $\hat{\theta}$, so the misreading only shortens the record the MPC plans through and moves its IAE from 7.19 to 6.66 °C min. On the fitted dead time the MPC beats the IMC-PI, 7.19 against 11.78 °C min, but not the reaction-curve PID of the preceding comparison (3.67 °C min, 25.8 % of travel).

The fouling ramp of case 14 is lost to the shipped cascade, 7.94 against 0.89 °C min, although neither leaves the 1 °C band. A random-walk disturbance with a fixed filter gain trails a ramp, and the MPC runs on average 0.14 °C below setpoint between 5 and 15 min and 0.29 °C between 25 and 40 min while $\hat{d}$ falls from -3.3 percent of valve at 10 min to -23 percent at 40 min, whereas the inner loop of the cascade answers the steam flow that fouling removes, 159 kg/h over the run at the design valve position, as soon as FT-102 registers it. On the challenge with its suggested settings, reported for information only, the MPC peaks at 1.33 °C, outside the 1 °C band, 2.2 min after the feed inlet temperature falls at 12 min, against 0.66 °C for the suggested feedforward, because the inlet

Table 4.8 Five Designs on the Tube Case at Three Entered Gains, From 11 to 40 min With the Same Seed and Events

Design	t_{TAH} (min)	Peak (°C)	IAE (°C min)	Steam (kg)	Travel (%)	Sat (min)
$\hat{K}_p = 0.252$ (0.7×), $\delta = 40.8\,\%$						
IMC-PI	2.96	90.52	19.30	863.7	74.9	0
FF + FIC	4.60	91.62	28.64	866.0	339.7	2.66
MPC, nominal	0	89.74	16.79	855.0	62.7	0
MPC, tube	0	89.62	16.36	854.5	64.1	0
MPC, band $-h_N$	0	89.57	23.58	850.2	191.2	0.30
$\hat{K}_p = 0.36$ (1.0×), $\delta = 1.8\,\%$						
IMC-PI	4.42	90.70	21.77	864.2	51.0	0
FF + FIC	3.36	91.53	22.07	863.4	198.8	1.96
MPC, nominal	0	89.92	19.33	855.7	53.2	0
MPC, tube	0	89.92	19.33	855.7	53.2	0
MPC, band $-h_N$	0	89.92	19.33	855.7	53.2	0
$\hat{K}_p = 0.54$ (1.5×), $\delta = 34.3\,\%$						
IMC-PI	6.52	90.91	26.41	865.1	35.6	0
FF + FIC	3.74	91.42	22.84	864.0	162.4	0.68
MPC, nominal	1.88	90.16	22.53	856.2	50.1	0
MPC, tube	1.14	90.12	21.76	855.5	61.3	0
MPC, band $-h_N$	0	89.26	25.66	849.9	217.1	0

t_{TAH} is the time at or above TAH-102 and Sat the time at a valve limit. Band $-h_N$ is the nominal band moved in by the tube's end half-width. IMC-PI gains 6.3, 4.4 and 3.0 %/°C, $\tau_I = 3$ min. Each MPC row made 290 plans, none rejected. No row reached TAHH-102, TAL-102 or FAL-102.

temperature, which the feedforward reads at TT-101, reaches the MPC only through TT-102 and $\hat{d}$.

On the other shipped upsets the MPC lies between the single loop and the structure built for each (IAE 4.01 against 6.60 and 0.78 °C min for the single loop and the cascade on the header sag, 7.19 against 11.67 and 4.81 °C min for the single loop and the feedforward on the colder feed), it beats the single loop under stiction (5.88 against 10.03 °C min). Outside case 17 no bound was active in any plan, so there the MPC acted as a fixed linear law and its results come from the model, the preview and the estimator rather than from constraint handling. All 3260 of its plans in Table 4.7 were accepted, at about 85 µs each in Node.js where no bound was active and 150 to 250 µs on case 17, and at 0.25 to 0.30 ms in Chrome, or 0.3 % of one processor core in real time (wall-clock values on one desktop computer).

4.7.3 Alarm Band From the Learner's Fit

The tube case starts in manual operation, steps FCV-102 by 5 % at 1 min as the step test, hands the valve to the MPC at 11 min, raises the setpoint from 85 to 89 °C, one degree under TAH-102, at 14 min, and drops the feed from 20 000 to 17 000 kg/h at 28 min, which heats the outlet toward the alarm. Every design is judged from 11 to 40 min with the same seed, instruments, $\hat{\tau}$ and $\hat{\theta}$, with $\hat{K}_p$ entered at 0.7, 1.0 and 1.5 times the 0.36 °C/% of the step test, as reported in Table 4.8. The PI rows use Equation (4.17) with $\tau_c = \hat{\theta}$ on the same entry, the feedforward row $K_{FF} = 1$ with that PI as the trim, and the MPC rows the settings of Section 4.5.4. The last row of each group is the nominal MPC retuned once by the tube's own measure, its band moved in at every decision by the half-width that the tube plan reaches at its end.

With the shipped entry of 0.355 °C/% the misfit is 1.03 %, the tube is 0.02 to 0.04 °C wide, and both bands give the same run (peak 89.91 °C, IAE 19.22 °C min), as they do at the step test's gain. At 0.7 times neither band reaches TAH-102, and the tube lowers the peak by 0.12 °C and the IAE by 2.6 % for 2.3 % more travel. At 1.5 times, δ is 34.3 %, the tube widens to ±1.49 °C at the end of the plan, and TT-102 crosses TAH-102 about 2.1 min after the feed drop under both bands. The tube shortens the time above it from 1.88 to 1.14 min with 3.4 % less IAE and 22 % more travel, as shown in Figure 4.4, but does not remove it.

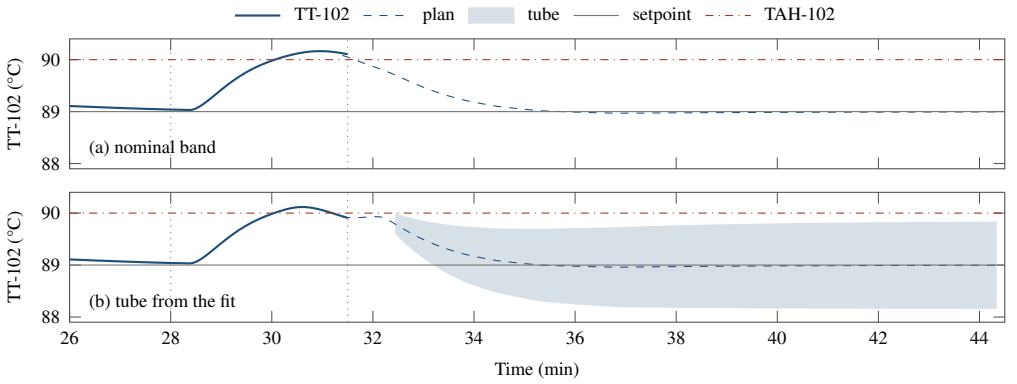


Figure 4.4 The tube case with $\hat{K}_p = 0.54$, 1.5 times the step test's gain, paused at $t = 31.5$ min (dotted vertical lines, with the feed drop at 28 min), with TT-102 to that instant (thick solid) and the reading expected by the plan made at 31.4 min (dashed). The setpoint is the thin gray line, TAH-102 the dash-dotted line and the tube the shaded band. (a) Nominal band, TT-102 at 90.10 °C with TAH-102 active and FCV-102 at 44.0 %. (b) Tube from the learner's fit, TT-102 at 89.91 °C, last at 90 °C at 31.18 min, FCV-102 at 40.9 % and a tube widening from ±0.20 °C to ±0.84 °C.

The remaining error is the feed path's. Applied open loop to a copy of the exchanger, the plans made at the feed drop (28.0 to 28.4 min) under-predict TT-102 by 0.26 °C 2.1 min ahead and by 0.57 °C 3.4 min ahead even with the step test's gain, inside tubes of 0.02 to 0.03 °C, because the feed path of Equation (4.19), identified at 20 000 kg/h, is off at 17 000 kg/h. With the right gain the feedback corrects this in time (peak 89.92 °C). At 1.5 times every correction is a third too small, the same plans miss by 0.50 and 1.11 °C, and their tubes of 0.34 and 0.72 °C cover the gain's share but not the feed path's. The crossing is not decided in the dead time, since closing FCV-102 at any plan up to 29.4 min would have held TT-102 at or below 89.97 °C, and a tube twice as wide removes it (peak 89.87 °C) while one three times as wide crosses again for 1.04 min.

The nominal band moved in by the tube's end half-width does remove the crossing at 1.5 times (peak 89.26 °C), at 3.5 times the tube's travel and 18 % more IAE, and at 0.7 times it saturates the valve for 0.30 min and raises the IAE from 16.36 to 23.58 °C min. The tube wins on IAE and travel and the constant back-off on the alarm. The PID structures, the cascade included (5.52 to 6.92 min, not tabulated), cross TAH-102 at every entry, and to stay under it the IMC-PI needs a setpoint of 88.00 to 88.50 °C and the feedforward 87.25 to 88.00 °C. Over 23 faults at their extremes in place of the feed drop, the nominal band and the tube spend 8.78 and 8.82 min above TAH-102 in all at 1.0 times the gain and 14.78 and 12.44 min at 1.5 times, against 26.78 to 100.24 min for the PID structures. No MPC run reaches TAHH-102, and the tube spends longer than the nominal band at a limit in 4 of 46 runs, by 0.04 to 0.10 min.

A wide tube works the valve hard. Its travel is 109 against 55 % for the nominal band at a misfit of 53 % and 539 against 76 % at 65 %, and at 70 % its plans hold FCV-102 at a stop for 1.90 min and set off FAL-102 twice, although it never spends longer above TAH-102. The MPC of Section 4.5.4 has a limit of its own. With $\hat{K}_p = 2.0$, 5.6 times the step test's gain, the nominal band moves FCV-102 through 1665 % of travel in the window, 6.90 min of it at a stop, with five FAL-102 alarms, in a cycle of plan and estimator with a period of about 2.4 min (the IMC-PI from that entry moves 15.8 % but spends 9.50 min above TAH-102). Around a linear plant of the model's structure the same controller cycles at a gain ratio of 5.56 (729 % of travel) but not at 5.0 (64 %), nor at 5.56 with $\kappa = 5$ or a factor of 0.1 on q_d (43 and 38 %), so the margin against an overestimated gain, about a factor of five, belongs to the design and not to the exchanger. The tube lowers it, with 2422 % of travel at $\hat{K}_p = 1.5$ where the nominal band moves 70 %.

4.7.4 Numerical Implementation

The application runs on the engine of Section 1.6.1, with its three clocks set as follows. The physics advances by classical fourth-order Runge-Kutta on a fixed 0.005 min step, the controller executes every 0.02 min, and the historian records every fourth physics step, 50 samples per simulated minute, the rate the application declares. The trajectory does not depend on frame rate and the noise is seeded, so two machines produce identical exports and noisy scenarios replay exactly.

Within a control execution the order is fixed, instruments first with the transport delay ahead of the bias, the noise and the transmitter lag, then the available-duty refresh, then the feedforward calculation, then the estimator of the predictive controller, then the controllers, then the stiction model, then integration. Placing the delay line ahead of everything else preserves the measurement chain, because no controller reads a state variable directly (the feedforward takes its latent heat at the design shell pressure, which no transmitter reads). The exported record carries eighty-six columns sampled every 0.02 min (1.2 s), including the true and the measured temperature, the commanded and the actual valve position, the separate proportional, integral and derivative contributions, the duty ledger, every disturbance, the estimate and plan of the predictive controller, and the misfit and widths of its tube, which is sufficient for an offline identification or grading task in any language. Forty self-checks run inside the application. Eighteen cover the steam-table fits, the log-mean fallback, the zero-derivative condition at the design steady state, the sign of every disturbance, the flow dependence of gain and dead time, the fouling and flooding relations, and the two tuning rules. Seven test the predictive controller, including its first move against a least-squares closed form, a forced rejection handed to TIC-101 without a bump, and a check of its source for any read of the plant state. Fourteen test the tube, including δ against a replay of the logged step test and each half-width against a recomputation, and one checks that the feedforward reads neither the shell pressure nor the steam quality.

4.8 Technology in Practice

The loop of Section 4.3 reads TT-102 through a first-order lag and moves FCV-102 through another. Figure 4.6 draws it with the devices an exchanger in a plant would carry in the place of the schematic's symbols.

What Measures It

An outlet temperature in a plant is read by a resistance temperature detector or a thermocouple, held in a thermowell. Thermocouples are the most rugged and wide-ranging of the contact sensors, and resistance detectors need no reference compensation and are more linear. Their accuracy varies with the type and the quality of the installation [26], so no single figure applies. The lag τ_{TT} of Table 4.2 stands for the element and its well, and the bias b for a lead-wire or calibration error that no controller can see.

> **TECHNOLOGY** **Sensor: A Resistance Thermometer in a Thermowell**
>
> A resistance temperature detector is a winding of pure metal wire, usually platinum, whose resistance always rises with temperature. It is relatively insensitive but very linear. The resistance of its lead wires adds to the reading and is taken as a falsely high temperature, a steady bias on the trend. The thermowell that protects the element adds a first-order lag through its own mass and heat capacity. An air gap between element and well adds a second lag, and multiple lags raise the phase shift of a feedback loop and may lead to oscillation [26]. How fast the reading answers is therefore set mostly by the installation, not by the element.

What Moves It

FCV-102 of Equation (4.9) is a first-order stem with a lag of 0.20 min, an installed characteristic x^a and an optional deadband S. A plant valve in this service is pneumatic, with a positioner that closes an inner loop on the stem [5], and the deadband stands for the friction that the positioner cannot

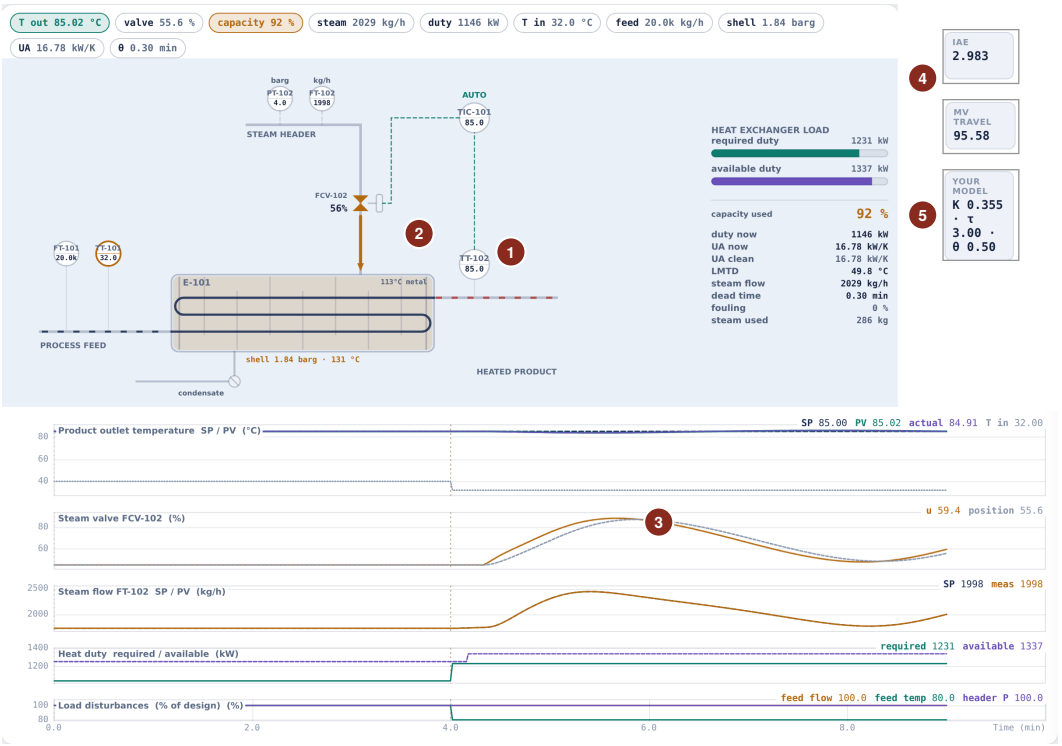


Figure 4.5 Tuning comparison after an inlet-temperature disturbance, at $t = 9$ min. A Ziegler-Nichols PID computed from (4.18) with the dead time read as $\hat{\theta} = 0.5$ min instead of the fitted 0.94 min ($K_c = 20.3\,\%/°\text{C}$, $\tau_I = 1.0$ min, $\tau_D = 0.25$ min) meets a feed 8 K colder at $t = 4$ min. The outlet stays within 1.24 °C of setpoint (Table 4.6) while the valve and the steam flow ring about the new resting position, the steam flow rising 386 kg/h above its final 2056 kg/h and falling 288 kg/h below it, where the SIMC tuning makes a single monotone valve move. IAE is 2.983 °C min and valve travel 95.6 % of span at this instant, reaching the 3.43 °C min and 112 % of Table 4.6 at $t = 10$ min. The identified constants are $\hat{K}_p = 0.355$, $\hat{\tau} = 3.00$ and $\hat{\theta} = 0.50$, which is where the error entered. The picture is a composite: the readouts and plant view, three scorecard tiles moved into the empty band beside them, and the trends. The callouts mark (1) TT-102, (2) FCV-102, (3) the ringing valve trend, (4) the IAE and valve travel tiles and (5) the identified model with its short dead time.

overcome. The heat-balance feedforward of Equation (4.16) also has its plant form. A feed flow transmitter and a summing block change the steam at once with the load, and because the steam heats more metal than the liquid does, the steam path is the slower one and the feedforward needs dynamic compensation [26], which is the role of τ_{lead} and τ_{lag}.

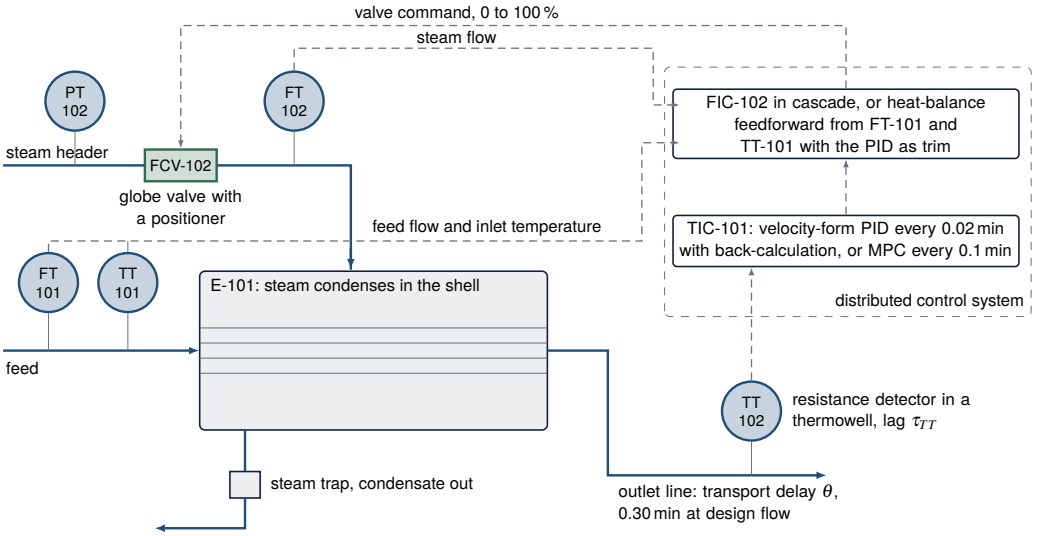


Figure 4.6 The loop of the chapter with plant devices in place of the schematic's symbols. A resistance detector in a thermowell reads TT-102 at the end of the outlet line, and the controllers on a distributed control system move valve FCV-102 through its positioner, either directly, through the steam flow loop FIC-102, or through the heat-balance feedforward. Notice that the transport delay and the thermowell both lie between the bundle and the controller, so the controller sees every move of the valve late.

> **TECHNOLOGY** **Final Element: The Steam Valve and Its Positioner**
>
> Steam, not the process liquid, is the stream to throttle, since changing the liquid's flow would upset the units around the exchanger [26]. A pneumatic valve is chosen for its failure mode first. An air-to-open valve fails closed on loss of air, and the risk of the process, not habit, should decide that position [26]. A positioner closes an inner loop on the stem to reduce the effects of friction and nonlinearity [5]. Friction it cannot overcome makes the stem stick and jump, and integral action then hunts above and below the setpoint [26].

Where the Controller Runs

A distributed control system runs production at one site in chemical manufacturing, refining and similar industries. Its loops are often nested, with the setpoint of one taken from another, and they run with cycle times from milliseconds to minutes [27]. TIC-101 and FIC-102 execute every 0.02 min (1.2 s) and the predictive controller decides every 0.1 min, both well inside that range. The cascade of the chapter is a remote setpoint from one PID block to another. Industrial practice places model predictive control one level above the PID loops that move the valves [20], while the MPC of Section 4.5.4 moves FCV-102 directly and hands it to TIC-101 when a plan is rejected.

IN PRACTICE The Plant Step Test

Before a model-based controller is commissioned, engineers run a plant test. A pre-test steps each manipulated variable to adjust the instruments and the existing PID loops, finds each output's time to steady state and gives data for a first model. The formal test runs around the clock with engineers watching. Each variable is stepped 8 to 15 times with a signal-to-noise ratio of at least six, which can take 5 to 15 days. No PID configuration or tuning change is allowed during the test, and operators may intervene only to avoid critical situations [20].

What Is Modernizing It

Three developments touch this loop. The first is the smart positioner. It records actuator pressure against stem position, and this valve signature identifies excessive packing friction, trim interference and poor seating [26], the faults behind the stiction of case 15. The second is the soft sensor for fouling. Regression and neural-network models estimate the fouling resistance of a steam heater from routine readings of flows and temperatures [16], the role the Conclusion gives to an online estimate of R_f. The third is predictive control on identified models. It originated in the process industries, where the processes are often slow enough to allow the computation [5], and its identification packages rely mainly on step-like or pseudo-random binary test signals [20], the industrial form of the step test of Section 4.4.

4.9 Limitations

The simulator is a teaching instrument and not a design tool. The lumped formulation cannot represent axial profiles, shell-side maldistribution, baffle effects or two-phase behavior on the tube side, so it must not be used to size equipment. Disturbances are deterministic steps and ramps, scripted by a scenario or set by the learner, rather than stochastic plant behavior. Two of them are stylized. A fouling ramp of 2.5 %/min would not occur in a physical system, and it serves to show the drift of a controller output under a slowly changing plant. A step in header pressure of 20 % is larger than most plants would see, and it separates the disturbance the inner flow loop can reject from the two it cannot. Sensors are ideal apart from the explicit lag, bias and noise that a user enables. The condensate system is reduced to one flooding parameter, and no equipment fault beyond fouling, flooding, stiction and transmitter bias is represented. The steam correlations are valid over 1 to 10 bar absolute and nowhere else, and the film correlation is extrapolated below the turbulent range. The fixed-step integrator and the fixed control period make every run reproducible, which a real distributed control system is not. Each simplification that was retained, the log-mean driving force, the flow-dependent film coefficient, the square-root valve law and the transport delay, was retained because removing it removes a behavior the loop must be tuned around. Section 4.3.5 states where each assumption fails.

The predictive controller plans on a fixed linear model, a FOPDT valve path from wherever the learner identified it and a feed path identified once at the design point. It omits the gain that falls with load and outlet temperature (0.56 to 0.22 °C/% between 16 000 and 24 000 kg/h, a secant gain of 0.277 °C/% between 85 and 90 °C), the dependence of a fitted gain on the direction of the step (0.355 and 0.440 °C/% for +5 and −5 % steps), the dead time that falls with feed rate, the available duty and stiction, and it sees the inlet temperature and the header pressure only through $\hat{d}$. Nothing re-identifies it on line, and its random-walk disturbance trails a ramp.

The tube inherits the reach of one valve step at one operating point and in one direction. It bounds a relative error of the valve path near that point, not an error in the declared feed path, a dead time that moves with flow or an unmeasured load, and for a gain error it is exact only once the estimator has settled. A misfit of 100 % or more builds no tube, and from about 70 % the tube drives the valve into its stops.

⚠ OPEN CHALLENGE Knowing When a Step Test Is Enough

A survey of industrial practice found that identification rests on step-like test signals that need the close attention of experienced engineers. No tool tells whether the data represent the process dynamics well enough for the controller, so plants are generally over-tested, which raises the cost of the project. Adaptive predictive controllers that keep their models current were proposed, but few reached the market, which reflects how hard adaptive control is in practice [20]. Progress would be a test that stops when the model is good enough and a controller that follows the load.

4.10 Discussion

Heat exchange accounts for most of the energy transferred within a process plant and for a large share of the energy lost. Pinch analysis established how strongly the recovery achievable in a network depends on how its exchangers are matched [28], and that recovery is realized only if each service holds its duty during upsets, which is a control problem. Fouling turns the same argument into a direct cost. An industrial survey of New Zealand plants quantified the combined capital and operating penalty of exchanger fouling across a national industrial base [29], and much of that penalty accrues through lost duty and extra fuel rather than through cleaning. Reading fouling from a drifting controller output, the task of scenario 14, is a maintenance-planning skill before it is a control skill, and the rise in balancing valve position from 44.9 % to 71.9 % across a full fouling ramp is the signal a monitoring system would trend.

Automation of these services also changes what an operator is asked to do. A cascade rejects the header-pressure upset almost completely, holding TT-102 within 0.16 °C of setpoint where the single loop dips 1.22 °C, and a heat-balance feedforward added on top answers measured load changes before they reach the outlet. That is desirable until the assumed heat capacity is wrong and the feedback trim has been absorbing the resulting bias over a long period. A heat-capacity error of $\pm 12\,\%$ is available as a disturbance for that reason, and (4.16) uses the nameplate value so the error is not silently corrected. Increasing automation does not remove the need to understand the loop. It moves the required understanding from moment-to-moment control to recognizing when the automation is compensating for a fault that ought to be corrected.

⚠ OPEN CHALLENGE Seeing Fouling Before It Costs Duty

Fouling remains the main unresolved problem in heat transfer and an almost universal one in the design and operation of exchangers. The deposit adds a thermal resistance, narrows the flow passage and raises the cost of maintenance. Neural-network and regression models now estimate the fouling resistance from routine plant readings to plan cleaning schedules [16], and online determination of that resistance in shell-and-tube exchangers is under study [30]. Progress would be an estimate trusted enough to schedule a cleaning by.

4.11 Conclusion

This chapter studied a steam-heated shell-and-tube exchanger whose plant constants are hidden, so the controller is built from an experiment. The simulator integrates five states with a log-mean driving force, a tube-side film coefficient that scales as $F_p^{0.8}$, saturated-steam correlations and a transport delay of 0.30 min at design flow that varies inversely with production rate. A 5 % valve step at the design point identifies $\hat{K}_p = 0.355\,^{\circ}\mathrm{C}/\%$, $\hat{\tau} = 3.02$ min and $\hat{\theta} = 0.94$ min, three lumped surrogates for five dynamic elements, and the same test at 16 000 and 24 000 kg/h moves the identified gain by a factor of 2.6. A dead time read 0.44 min short raises the reaction-curve gain from the 10.8 %/°C that the fitted dead time warrants to the 20.3 %/°C the scenario loads, where the SIMC rule on the same fit returns 4.5 %/°C. After an 8 K inlet drop both reaction-curve tunings reach an IAE near 3.4 °C min at $t = 10$ min against 8.61 °C min for SIMC, so the error advantage belongs to the rule. The misreading adds nothing to it, multiplies the valve travel by 4.5 and leaves a loop that grows into a valve-limited oscillation when the feed falls to 18 000 kg/h. Around that comparison the application arranges load

upsets that motivate cascade and feedforward, and four fault conditions whose symptoms resemble a tuning problem without being one.

A linear MPC built on the same three identified numbers, with FT-101 as a preview and the valve range as a hard limit, reaches 14.05 °C min on the added comparison case against 20.64 for the IMC-PI and 17.21 for the feedforward, and it loses the setpoint step alone (16.21 against 13.22 °C min) and the fouling ramp. A tube sized from the learner's own step-test misfit costs nothing with a correct fit and, with the gain entered 1.5 times too high, shortens the time above TAH-102 from 1.88 to 1.14 min where a band moved in by its end width removes it at 3.5 times the valve travel, because the error that remains lies in the feed path and not in the gain.

The model supports control methods beyond the four architectures. Equations (4.1) to (4.11) are a small differential-algebraic system with measured inputs, so they can be posed directly as the prediction model of a nonlinear model predictive controller [31, 32]. The linear MPC of Section 4.5.4 plans against the 0–100 % valve span and coordinates the steam valve with the measured feed rate, but it knows neither the available duty of 1252 kW at design flow nor the gain that falls with load, and a nonlinear predictive controller on the full model would add both. Moving horizon estimation or an extended Kalman filter driven by TT-102, FT-101 and FT-102 would infer the fouling resistance R_f and the tube-metal temperature T_w, neither of which is measured. An online estimate of R_f turns the drifting controller output of scenario 14 into a soft sensor for cleaning schedules. A fuller model would add a distributed tube-side model for axial profiles, a condensate system with stall dynamics, a deposition-minus-removal fouling law, and scenarios at more than one production rate to make gain scheduling measurable. The lesson of the exchanger is that the three identified numbers belong to an operating load as much as to the plant, and that the experiment behind them decides more than the tuning rule applied to them.

KEY POINTS

- A small step of the steam valve gives three numbers for tuning, but they belong to one feed rate as much as to the exchanger.
- The fitted dead time of 0.94 min exceeds the 0.30 min transport delay because it lumps the valve, the metal and the transmitter.
- A dead time read 0.44 min short raises the reaction-curve gain from 10.8 to 20.3 %/°C, adds nothing to the error advantage and multiplies the valve travel by 4.5.
- Fouling shows in the controller output, which drifts upward while the temperature stays at setpoint.

Where else this applies. The same step test and tuning rules meet a tank in Volume 1, Chapter 2 and a heated device in Volume 1, Chapter 4.

References

[1] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, III, *Process Dynamics and Control*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.

[2] B. W. Bequette, *Process Control: Modeling, Design, and Simulation*. Upper Saddle River, NJ: Prentice Hall, 2003.

[3] F. W. Dittus and L. M. K. Boelter, "Heat transfer in automobile radiators of the tubular type," *International Communications in Heat and Mass Transfer*, vol. 12, no. 1, pp. 3–22, 1985.

[4] E. N. Sieder and G. E. Tate, "Heat transfer and pressure drop of liquids in tubes," *Industrial & Engineering Chemistry*, vol. 28, no. 12, pp. 1429–1435, 1936.

[5] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed. Princeton, NJ: Princeton University Press, 2021.

[6] D. J. Correa and J. L. Marchetti, "Dynamic simulation of shell-and-tube heat exchangers," *Heat Transfer Engineering*, vol. 8, no. 1, pp. 50–59, 1987.

[7] H. Müller-Steinhagen, "Heat transfer fouling: 50 years after the Kern and Seaton model," *Heat Transfer Engineering*, vol. 32, no. 1, pp. 1–13, 2011.

[8] W. Miao and B. Xu, “Application of feedforward cascade compound control based on improved predictive functional control in heat exchanger outlet temperature system,” *Applied Sciences*, vol. 13, no. 12, p. 7132, 2023.

[9] A. Vasičkaninová, M. Bakošová, A. Mészáros, and J. J. Kleměs, “Neural network predictive control of a heat exchanger,” *Applied Thermal Engineering*, vol. 31, no. 13, pp. 2094–2100, 2011.

[10] S. Bennett, “The past of PID controllers,” *Annual Reviews in Control*, vol. 25, pp. 43–53, 2001.

[11] J. G. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” *Transactions of the ASME*, vol. 64, no. 8, pp. 759–765, 1942.

[12] D. E. Rivera, M. Morari, and S. Skogestad, “Internal model control: PID controller design,” *Industrial & Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252–265, 1986.

[13] S. Skogestad, “Simple analytic rules for model reduction and PID controller tuning,” *Journal of Process Control*, vol. 13, no. 4, pp. 291–309, 2003.

[14] B. W. Bequette, “Process control practice and education: Past, present and future,” *Computers & Chemical Engineering*, vol. 128, pp. 538–556, 2019.

[15] J. A. Rossiter, B. Pasik-Duncan, S. Dormido, L. Vlacic, B. Jones, and R. Murray, “A survey of good practice in control education,” *European Journal of Engineering Education*, vol. 43, no. 6, pp. 801–823, 2018.

[16] R. Jradi, C. Marvillet, and M. R. Jeday, “Estimation and sensitivity analysis of fouling resistance in phosphoric acid/steam heat exchanger using artificial neural networks and regression methods,” *Scientific Reports*, vol. 13, p. 17889, 2023.

[17] M. A. A. S. Choudhury, N. F. Thornhill, and S. L. Shah, “Modelling valve stiction,” *Control Engineering Practice*, vol. 13, no. 5, pp. 641–658, 2005.

[18] K. J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2006.

[19] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[20] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[21] G. Pannocchia and J. B. Rawlings, “Disturbance models for offset-free model-predictive control,” *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[22] D. Goldfarb and A. Idnani, “A numerically stable dual method for solving strictly convex quadratic programs,” *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[23] L. Chisci, J. A. Rossiter, and G. Zappa, “Systems with persistent disturbances: predictive control with restricted constraints,” *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.

[24] D. Q. Mayne, M. M. Seron, and S. V. Raković, “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[25] M. V. Kothare, V. Balakrishnan, and M. Morari, “Robust constrained model predictive control using linear matrix inequalities,” *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[26] T. R. Kuphaldt, *Lessons in Industrial Instrumentation*, version 2.33 ed. Open textbook, Creative Commons Attribution 4.0 license, ibiblio.org/kuphaldt/socratic/sinst, 2024.

[27] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, “Guide to operational technology (OT) security,” National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-82r3, 2023.

[28] B. Linnhoff and E. Hindmarsh, “The pinch design method for heat exchanger networks,” *Chemical Engineering Science*, vol. 38, no. 5, pp. 745–763, 1983.

[29] R. Steinhagen, H. Müller-Steinhagen, and K. Maani, “Problems and costs due to heat exchanger fouling in New Zealand industries,” *Heat Transfer Engineering*, vol. 14, no. 1, pp. 19–30, 1993.

[30] M. Markowski and P. Trzcinski, “Comparison of the analytical model with neural network model on the case of heat exchanger behavior under fouling,” *Scientific Reports*, vol. 15, p. 42078, 2025.

[31] J. D. Hedengren, R. Asgharzadeh Shishavan, K. M. Powell, and T. F. Edgar, “Nonlinear modeling, estimation and predictive control in APMonitor,” *Computers & Chemical Engineering*, vol. 70, pp. 133–148, 2014.

[32] L. D. R. Beal, D. C. Hill, R. A. Martin, and J. D. Hedengren, “GEKKO optimization suite,” *Processes*, vol. 6, no. 8, p. 106, 2018.

4.12 Exercises

4.1 Concept check · 1 min

TIC-101 can drive the steam valve directly, send a steam-flow setpoint to FIC-102, or trim a heat-balance feedforward. Which variable is the manipulated variable of the exchanger in every one of these structures?

(a) The steam-flow setpoint of FIC-102, in percent of the FT-102 span
(b) The position of the steam valve FCV-102, in percent of its travel
(c) The pressure of the steam header at PT-102, in bar gauge
(d) The process feed flow measured at FT-101, in kilograms per hour

4.2 Concept check · 2 min

Equation (4.1) is the energy balance on one of the five inventories of the model. Which inventory is it, and where does its heat come from?

(a) The tube metal, which receives the duty Q_s from the condensing steam
(b) The steam in the shell, which is charged by the flow through FCV-102
(c) The process fluid in the bundle, which receives the duty Q_p from the tube metal
(d) The process fluid in the bundle, which receives the duty Q_s straight from the steam

4.3 Concept check · 2 min

The tubes foul slowly while the loop holds 85 °C in automatic at a constant feed rate and inlet temperature. Which signal shows the fouling?

(a) TT-102 drifts below its setpoint as the conductance of the bundle falls
(b) The transport delay to TT-102 grows as the deposit builds in the tubes
(c) TT-101 at the feed inlet falls as less heat reaches the product
(d) The valve position drifts upward while TT-102 stays on its setpoint

4.4 Theory and calculation · 5 min

Exchanger E-101 heats 20 000 kg/h of process liquid with $C_p = 4.18\,\mathrm{kJ/kg/K}$ from 40 to 85 °C. The shell settles at 2.60 bar absolute.

(a) Find the duty in kilowatts. *Checkpoint:* 1045 kW.

(b) Find the latent heat from Equation (4.8) and the steam flow that supplies the duty. *Checkpoint:* 2178 kJ/kg and 1727 kg/h.

(c) Find the transport delay to TT-102 at this feed rate and at 30 000 kg/h, with $V_{\mathrm{trans}} = 0.1\,\mathrm{m}^3$ and $\rho = 1000\,\mathrm{kg/m}^3$. *Checkpoint:* 0.30 and 0.20 min.

4.5 Theory and calculation · 8 min

A valve step at the design point is fitted as $\hat{K}_p = 0.355\,^{\circ}\mathrm{C}/\%$, $\hat{\tau} = 3.0\,\mathrm{min}$ and $\hat{\theta} = 0.94\,\mathrm{min}$.

(a) Find the SIMC PI settings of Equation (4.17) with $\tau_c = \hat{\theta}$. *Checkpoint:* $K_c = 4.5\,\%/^{\circ}\mathrm{C}$ and $\tau_I = 3.0\,\mathrm{min}$.

(b) Find the reaction-curve PID settings of Equation (4.18). *Checkpoint:* $K_c = 10.8\,\%/^{\circ}\mathrm{C}$, $\tau_I = 1.88\,\mathrm{min}$ and $\tau_D = 0.47\,\mathrm{min}$.

(c) The dead time is misread as 0.50 min. Find both sets of settings again, with $\tau_c = \hat{\theta}$ as before, and say what changes in each. *Checkpoint:* SIMC 8.5 and 3.0 min. Reaction curve 20.3, 1.0 min and 0.25 min.

4.6 Theory and calculation · 6 min

The clean conductances of Table 4.2 are $UA_{p0} = 1380\,\mathrm{kJ/min/K}$ and $UA_{s0} = 3726\,\mathrm{kJ/min/K}$, and they act in series.

(a) Find the overall conductance of the clean exchanger at the design feed rate, in kW/K. *Checkpoint:* 16.78 kW/K.

(b) Find it again at 100 % fouling, where $R_f = 2.6 \times 10^{-4}\,\mathrm{min\,K/kJ}$, and the loss in percent. *Checkpoint:* 13.30 kW/K, a loss of 20.7 %.

(c) Find it for the clean exchanger at 24 000 kg/h from Equation (4.10), and compare its rise with the rise in duty. *Checkpoint:* 18.63 kW/K, a rise of 11 % against 20 % in duty.

4.7 App challenge · 9 min

Load scenario 3, Step test, and change no setting. The run starts in Manual at a steady state with FCV-102 at 44.9 %. Use the default seed. This scenario has no measurement noise. Press Run, let about one minute pass and press Step valve +5%. Run until 20 min after the step and export the CSV file. Measure times from the last row in which `valve_cmd_pct` still reads 44.9.

(a) Read TT-102, column `temp_pv_C`, before the step and 20 min after it, and compute the process gain. *Checkpoint:* 85.01 and 86.78 °C, so 0.354 °C/%, within 0.005.

(b) Find the times at which TT-102 has covered 28.3 and 63.2 % of that change, and apply Equation (1.9). *Checkpoint:* 2.00 and 3.96 min, within 0.05 min, so $\hat{\tau} = 2.94\,\mathrm{min}$ and $\hat{\theta} = 1.02\,\mathrm{min}$.

(c) Press Fit FOPDT to my step test and read the Regression tile. Compare the three numbers with the two-point fit and with the transport delay of 0.30 min. *Checkpoint:* 0.355, 3.02 and 0.94, within 0.002, 0.06 min and 0.02 min.

4.8 App challenge · 8 min

Load scenario 8, Ziegler-Nichols, and an optimistic dead time. The loop is in Auto with the reaction-curve tuning from a dead time read as 0.5 min, and the feed inlet falls from 40 to 32 °C at 4 min. Use the default seed and run each case to 24 min. For each run read the IAE and the valve travel on the scorecard and find the lowest TT-102 in the exported file.

(a) Run the scenario as shipped. *Checkpoint:* 6.05 °C min, 187 % and 83.76 °C.

(b) Press Reset. Before Run, set Identified dead time to 0.94 min and select ZN PID under Tune from my model, which loads $K_c = 10.8\,\%/°\mathrm{C}$, $\tau_I = 1.88\,\mathrm{min}$ and $\tau_D = 0.47\,\mathrm{min}$. Run again. *Checkpoint:* 3.67 °C min, 25.8 % and 83.64 °C.

Accept an IAE within 0.1 °C min, a travel within 5 % and a temperature within 0.03 °C. State what the misread dead time bought and what it cost.

4.9 App challenge · 8 min

Load scenario 17, Capacity limit. The feed is 27 000 kg/h, the tubes are 70 % fouled and the header is at 3.2 barg. At 20 min the exchanger is cleaned and the header returns to 4.0 barg. Use the default seed, run each case to 28 min and export the CSV file.

(a) Run the scenario as shipped. Read TT-102 in the row at 19.9 min, and the required and the available duty in the columns `duty_required_kW` and `duty_available_kW`. *Checkpoint:* 79.82 °C, 1411 kW and 1248 kW.

(b) In the same file find the time at which `valve_cmd_pct` first falls below 100 and the highest TT-102 after it. *Checkpoint:* 22.66 min, within 0.05 min, and 86.85 °C.

(c) Press Reset, set Anti-windup to Off at the Advanced level and run again. Read the valve command and TT-102 at 28 min, and compare the column `I_term` at 19.9 min in the two files. *Checkpoint:* 100 % and 87.70 °C. The integral term is 254.6 % against 100.0 %.

Accept a temperature within 0.03 °C and a duty within 2 kW. Say why no tuning removes the offset before 20 min, and what the anti-windup changes after it.

4.10 Application area · 3 min

At the design load a ramp to 100 % fouling moves the balancing valve position from 44.9 to 71.9 % and the overall conductance from 16.78 to 13.30 kW/K, while the loop holds 85 °C.

(a) Say what has happened to the duty over the ramp, and why the outlet temperature is of no use as a fouling monitor. *Checkpoint:* The duty is unchanged at 1045 kW.

(b) Name two other faults of the model that raise the valve position at a constant load, and the instrument or check that tells each from fouling. *Checkpoint:* For example a sagging header, seen at PT-102.

4.11 Application area · 3 min

In the cascade TIC-101 writes the setpoint of FIC-102 in percent of the 4000 kg/h span of FT-102, limited to 0–100 %. The steam header delivers at most 2600 kg/h.

(a) Express the header capacity and the design steam flow of 1727 kg/h in percent of span. *Checkpoint:* 65 % and 43.2 %.

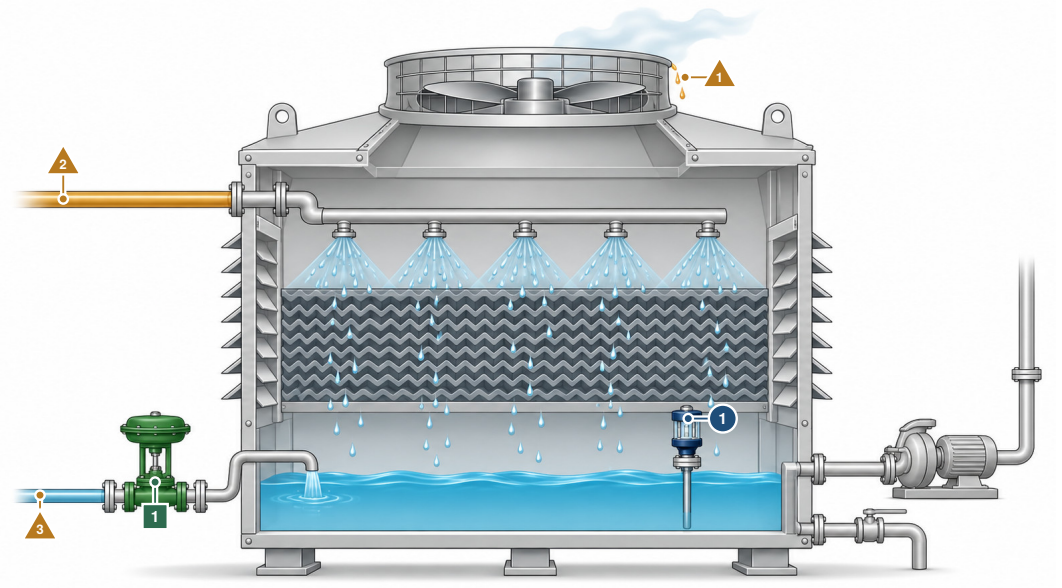
(b) During a large upset the setpoint from TIC-101 passes 65 %. Say what each integral does, and what an engineer would configure to prevent it. *Checkpoint:* The outer integral winds.

4.12 Review the agent · 5 min

An AI assistant is asked to tune TIC-101 at the design point without a step test and replies as follows. "The dead time of this loop is the transport delay from the bundle to TT-102, $\theta = V_{\text{trans}}\rho/\dot{m}_p = 0.1 \times 1000/333.3 = 0.30$ min. This is exact physics, so it is better than a dead time read from a trend. With $K_p = 0.355\,^{\circ}\text{C}/\%$ and $\tau = 3.0$ min the reaction-curve rule gives $K_c = 1.2 \times 3.0/(0.355 \times 0.30) = 33.8\,\%/^{\circ}\text{C}$, $\tau_I = 0.60$ min and $\tau_D = 0.15$ min, and the loop will reject loads about three times better than with a dead time of 0.94 min." Find the flaw in the proposal and say how it could be checked.

CHAPTER 5

Cooling Tower Makeup Simulator for Identification and Tuning of Integrating Loops



The controller moves 1 the makeup valve

The controller reads 1 the basin level

What disturbs it 1 water carried off as vapor and drift 2 more heat in the return water 3 a sag in the makeup supply pressure

Notice that nothing that drains the basin depends on its level, so the makeup valve alone must match the losses or the level keeps moving. Illustration, not to scale.

IN PLAIN TERMS **Keeping a Cooling Tower's Pool Topped Up**

The main idea. A cooling tower cools water by letting some of it evaporate, and more leaves as spray and drained water. A controller opens a refill valve to keep the pool under the tower at a steady depth.

The challenge. None of these losses changes with the depth, so any mismatch makes the pool rise or fall at a steady rate. One quick test of the valve sets up the controller, and a delay read too short makes the valve swing from shut to wide open.

Why it matters. The refill water is treated and paid for, and a low pool can stop the pumps that cool the plant.

THREE WAYS TO READ THIS CHAPTER

Explore	12 min	the first page, the fact on water use and the history of the float regulator (Section 5.2), the try-it segment of case 6 (Section 5.5.2), the fact on reset time (Section 5.5.3), the try-it segment of case 1 (Section 5.6) and the key points
Study	60 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	20 min	the first page, the results (Section 5.6), Technology in Practice (Section 5.7), Limitations (Section 5.8) and the open challenge

The control question A cooling tower basin loses water by evaporation, blowdown and drift, and none of the three depends on the level, so the level ramps where a drained tank would settle. What does a step test measure on such a plant, and how do its two numbers become a controller that neither cycles nor wears out the makeup valve?

Learning objectives

- Explain from the water balance why the basin level integrates, and compute the valve position that balances the losses.
- Identify an integrating gain and an apparent dead time from the slope of a step response.
- Compute SIMC and Ziegler–Nichols settings from the identified model, and predict the result of a dead time read too short.
- Predict the proportional offset after a load change, and explain why the reset time of an integrating loop must be long.
- Distinguish a makeup capacity limit from a tuning problem.

Prerequisites Sections 1.4.4, 1.4.5 and 1.4.6 (step test, tuning rules, integrating process), 1.4.7 (ultimate gain) and 1.4.9 (windup). Chapter 4 runs the same workflow on a plant that settles. Section 1.6.3 is needed only for the predictive controller.

The app https://apmonitor.com/apps/cooling-tower/

See also Volume 1, Chapter 2 for the tank that regulates itself and Volume 1, Chapter 9 for another inventory that integrates. Volume 9, Chapter 2 for predictive control of a delayed integrating plant and Volume 9, Chapter 4 for offset-free estimation.



5.1 Introduction

A gravity-drained tank, the plant of Volume 1, Chapter 2, is the first level loop most process control courses present, and it is atypical. Its outlet flow rises with the head above the nozzle, so an inflow that is slightly too large raises the level until the outflow grows to match, and the tank reaches a steady state without control. Such a plant is self-regulating. A cooling tower basin has no such mechanism. Water leaves the basin as vapor carried off by the fan, as a bleed stream through the blowdown valve, and as droplets entrained past the drift eliminators, and the rate of each is fixed by the process heat load, by the position of the blowdown valve and by the airflow. None of the three is a function of the basin level. The consequence is a plant with a pole at the origin, a level that ramps rather than settles, and an operator who can hold the basin steady only by matching two flows exactly and indefinitely.

The same structure appears throughout a process plant. Surge drums, storage tanks, boiler drums, hoppers and reactor holdups behave the same way whenever the discharge is set by a pump or a downstream controller rather than by hydrostatic head, and integrating inventories are the ordinary case in a process plant rather than the exception [1, 2]. The water level of the pool in Volume 1, Chapter 9 is a pure integrator for the same reason. The physical origin is conservation of mass with outflows independent of the stored inventory, the structure textbooks treat as the integrating or pure capacity process [3, 4]. What changes for the engineer is the whole identification and tuning sequence. An open-loop step on a self-regulating plant is read for a steady-state gain, a time constant and a dead time. An open-loop step on an integrating plant never reaches a steady state, so the reading is a change of slope and a delay, the fitted model is integrating plus dead time (IPDT), and the tuning rules that follow place the dead time alone in the denominator of the gain [5, 6]. Stability of the closed loop then rests entirely on the ratio of the closed-loop speed to that dead time [7].

Simulators of cooling towers are usually built to answer thermal questions. The Merkel enthalpy-difference method and the more complete Poppe and effectiveness-NTU formulations predict outlet water temperature against wet-bulb temperature and air flow [8, 9], and design packages extend that to fill selection and fan power [10]. The water side receives less attention even though it is where the control loop lives [11]. However, a student who is handed an IPDT transfer function and asked for PI settings never performs the experiment that produces the two numbers, and the arithmetic of the

tuning rule is the easy part. This chapter works with an interactive cooling tower basin whose holdup, valve capacity, transport delay and actuator lag are hidden, so that the model must be identified from a measured slope and an identification error appears as sustained oscillation rather than as a small correction. The scenarios cover integrating dynamics, dead time in series with an actuator lag, open-loop identification, model-based PID tuning, anti-windup and the diagnosis of an actuator capacity limit, principles a standard curriculum develops with self-regulating examples [3, 7]. Section 1.4.6 introduces the integrating process with dead time and its tuning rule, and this chapter carries both through on one plant.

The chapter is organized as follows. Section 5.2 reviews evaporative cooling, blowdown and makeup level control. Section 5.3 gives the three-state water and solids balance with its instrument chain and its actuator model, written out in the detail needed to rebuild the simulator. Section 5.4 derives the transfer function from valve to level, identifies it from the slope of a step response and shows why the apparent dead time exceeds the transport delay. Section 5.5 sets out the level controller, the two tuning rules computed from the identified model, the proportional offset, the role of the reset time and a linear model predictive controller (MPC) built on the same identified integrating model. Section 5.6 measures these designs on a scenario sequence that begins with manual operation and ends at a makeup capacity limit that no tuning removes. Section 5.7 sets the loop in its plant hardware, Sections 5.8 and 5.9 state what the model omits and what level control costs in water, and Section 5.11 holds the exercises.

5.2 Background

Evaporative cooling is a mature technology used at large scale. A tower rejects heat by evaporating roughly one percent of the circulating flow for every 10 °F of cooling range, which is why the water side of a tower is a consumption problem and not merely a recirculation problem. Thermoelectric generation, refining, chemicals and data centers all depend on it, and a review of operational water factors for electricity generation puts recirculating cooling in a class where consumption per unit of output is far higher than for once-through cooling [12]. Every gallon that evaporates leaves its dissolved solids behind in the basin, so a second stream has to be discarded to keep the concentration bounded, and a third is lost as fine droplets past the eliminators.

> **DID YOU KNOW? Where a Building's Water Goes**
>
> In commercial and institutional buildings, cooling towers are often the largest single use of water, at 20 to 50 percent or more of a facility's total. Drift, the mist carried off in the air, is 0.05 to 0.2 percent of the flow through a tower, and modern drift eliminators can cut it below 0.005 percent [13].

Analysis of the thermal side matured first. Merkel's enthalpy-driving-force treatment of the 1920s was reduced to the tower characteristic and the demand curves that designers still use [8], and later work quantified how far that treatment departs from a full heat and mass balance and how the evaporated fraction should be corrected for it [9, 11]. The water balance followed the chemistry rather than the thermodynamics. The ratio of basin conductivity to makeup conductivity defines the cycles of concentration, and the blowdown flow that holds a target ratio is fixed once evaporation and drift are known. Raising the target reduces water use and moves the basin closer to scaling and corrosion limits, which is why blowdown moved from a fixed manual bleed to a conductivity switch that opens a valve only when the basin passes its limit. Blowdown streams are concentrated enough that their treatment and reuse are studied in their own right [14].

Makeup level control followed a different path. The earliest arrangement is a mechanical float valve in the basin, which is a proportional controller whose gain is set by the float geometry and whose offset is rarely measured. Pneumatic and then electronic level transmitters replaced the float, and the loop became one line in a distributed control system. The difficulties characteristic of level loops survived the change of hardware. A basin level offers no natural damping, its measurement is

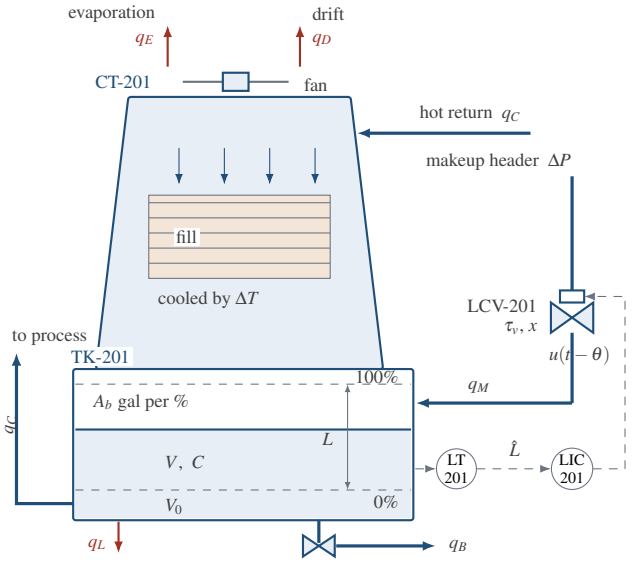


Figure 5.1 Cooling tower CT-201 and basin TK-201 with the makeup, blowdown, drift and evaporation streams. Circulating water q_C returns hot, is cooled through the fill by ΔT and falls into the basin, which holds volume V above the transmitter zero at conductivity C and volume V_0 below it. Makeup q_M enters through valve LCV-201, whose stem x follows the delayed controller output $u(t-\theta)$ with lag τ_v against header differential ΔP. LT-201 reports the level $\hat{L}$ over a span holding A_b gallons per percent, and LIC-201 drives the valve.

noisy because the surface is disturbed by the returning water, and the loops are frequently detuned on purpose so that the inventory absorbs upsets rather than passing them downstream [2]. Tuning rules written specifically for integrator-plus-dead-time plants appeared for the same reason [1, 6], and identification methods for plants with a pole at the origin are still an active subject [15], because the settled step response that a self-regulating plant provides is not available and the model must be constructed from a slope.

> **HISTORY** **The Float Regulator, About 270 BC to 1791**
>
> Around 270 BC the Greek inventor Ktesibios built a float regulator for a water clock. It held the water in a tank at a constant depth, so that a steady flow from its bottom filled a second tank at a constant rate, and the level in the second tank told the time. The float worked an inflow valve: as the level fell, the valve opened and refilled the tank, as the ball and cock of a flush toilet does today. The device returned in the late 1700s, when the boilers of steam engines and household water reservoirs both needed a held level. A patent of 1758 describes a float valve on a steam boiler, and by 1791 the float regulator was in common use in steam engines [16]. The first makeup controller of a cooling tower basin was the same device.

5.3 Process Model

Tower CT-201 is an induced-draft counterflow unit that circulates 3000 gpm of water through a process returning it 10 °F hotter at design load. Cooled water falls into basin TK-201, which feeds the circulating pump. Makeup comes from a treated-water header through control valve LCV-201, blowdown leaves through a valve of its own, and level transmitter LT-201 measures a 1.2 ft band on a 250 ft^2 basin. Figure 5.1 shows the layout, the instrument tags and each symbol used in the equations that follow. LT-201 supplies the controlled variable in percent of that span, and LCV-201 position is the only manipulated variable in the application.

Table 5.1 Classification of Variables. Nominal Values Are Those of the Design Operating Point at Which Every Scenario Starts.

Symbol	Role	Nominal	Range
L (LT-201)	CV, measured	60 %	30–85 % SP
u (LCV-201)	MV	42.5 %	0–100 %
heat load	DV, measured	100 %	50–150 %
ΔP	DV, measured	100 %	40–120 %
blowdown lift	DV, measured	100 %	0–300 %
m_D drift	DV, unmeasured	1×	1–40×
q_L leak	DV, unmeasured	0 gpm	0–15 gpm
V	state	1350 gal	0–2250 gal
x	state	42.5 %	0–100 %
C	state	2000 µS/cm	$\geq C_M$

Table 5.1 classifies what the loop interacts with. Three alarm limits bound normal operation, a high level at 85 %, a low level at 40 % and a pump-suction trip at 20 %. None of the three stops the run. Each one latches an annunciation, so a test that carried the basin outside its limits leaves a record. The plant is expressed in the units of utility operation, flows in gpm, volumes in gal, level and valve position in percent, conductivity in µS/cm and time in minutes. Three states are integrated.

5.3.1 Water Balance on the Basin

The basin is a single well-mixed inventory whose level is linear in volume over the transmitter span, and the balance that governs it is Equation (5.1).

$$A_b \frac{\mathrm{d}L}{\mathrm{d}t} = q_M - q_E - q_B - q_D - q_L \tag{5.1}$$

where A_b is the basin holdup per percent of span in gal/%, L is the level in percent, q_M is makeup, q_E is evaporation, q_B is blowdown, q_D is drift and q_L is a basin leak, all in gpm. Both sides carry units of gal/min. The state actually integrated is the volume above the transmitter zero, $V = A_b L$, in gallons.

Nothing on the right-hand side of (5.1) is a function of L, and that absence is the subject of this chapter. Substituting a head-driven outlet $q_{\text{out}} = k\sqrt{L}$ would add a term that grows as the tank fills, would linearize to a first-order lag, and would give the tank a steady state of its own. The cooling tower basin has no such term, so a constant imbalance produces a constant slope and the level travels until it reaches the overflow weir at 100 % or the empty basin at 0 %. Both walls are enforced twice. The derivative in (5.1) is set to zero when the state sits at a bound and the imbalance points outward, so the fixed-step integrator cannot pass through the wall and return, and the state is clipped again after every step. Volume clipped off at the top latches a basin-overflow annunciation. While the level sits on the weir, the surplus that the blocked derivative discards is recomputed after each step and metered as a spill rate and a totalized spill volume, and a deficit at the empty basin is reported the same way, so the water sent to drain is accounted for during a sustained overflow.

🕮 **Pause.** Try Exercise 5.1 before reading on.

5.3.2 Evaporation, Blowdown and Drift

Evaporation is written in the form a tower vendor uses, as a fraction of the circulating flow proportional to the cooling range, as shown in Equation (5.2).

$$q_E = 0.001\, f_E\, q_C\, \Delta T, \qquad \Delta T = \Delta T_{\text{nom}} \frac{\text{heat load}}{100\,\%} \tag{5.2}$$

where q_C is the circulation rate in gpm, ΔT is the cooling range in °F, f_E is the fraction of the duty rejected as latent heat, and the coefficient 0.001/°F carries the units. The heat load is an operator-set

input that moves ΔT and therefore q_E with no dynamics between them. Drift is the entrained liquid that escapes the eliminators, taken as a fixed fraction of circulation in Equation (5.3).

$$q_D = f_D\, q_C\, m_D \tag{5.3}$$

where f_D is the design drift fraction and m_D is a multiplier that represents damaged or fouled eliminators. Blowdown is sized from the cycles of concentration $N = C/C_M$ that the water treatment program holds, as shown in Equation (5.4).

$$q_{B,0} = \max\left(0,\ \frac{q_E}{N-1} - f_D q_C\right), \qquad q_B = q_{B,0}\,\frac{\text{lift}}{100\,\%} \tag{5.4}$$

where N is floored at 1.05 and the design drift $f_D q_C$ is credited against the required bleed because drift removes solids as effectively as blowdown does. The multiplier m_D deliberately does not appear in (5.4), so raising the drift on a fixed bleed dilutes the basin and pulls the achieved cycles below target, which is what a real fixed bleed does. In the alternative blowdown mode the lift term is replaced by an on-off valve that passes $3q_{B,0}$ while open and nothing while shut, switched on basin conductivity with a hysteresis band of $\pm 1.5\,\%$ about NC_M.

At the design point the three losses are $q_E = 25.50\,\text{gpm}$, $q_B = 6.225\,\text{gpm}$ and $q_D = 0.15\,\text{gpm}$, which total 31.875 gpm.

5.3.3 Makeup Valve, Transport Delay and Level Transmitter

The makeup valve is linear in stem position with a square-root pressure term, and its stem follows a delayed command through a first-order actuator lag, as shown in Equations (5.5) and (5.6).

$$\tau_v \frac{\mathrm{d}x}{\mathrm{d}t} = u(t-\theta) - x \tag{5.5}$$

$$q_M = q_{M,\max}\,\frac{x}{100}\sqrt{\frac{\Delta P}{\Delta P_{\text{nom}}}} \tag{5.6}$$

where x is the stem position in percent, u is the controller output in percent, θ is the combined transport and measurement dead time in minutes, τ_v is the actuator lag in minutes, $q_{M,\max}$ is the valve capacity at full lift and nominal differential, and $\Delta P/\Delta P_{\text{nom}}$ is the header pressure ratio. Delaying the command in (5.5) rather than delaying the flow is equivalent and keeps the delay line out of the right-hand side, which the fixed-step integrator evaluates four times per step. The delay line stores each controller output and is read at every controller execution, so a command issued at one execution, the first of a run included, reaches the stem exactly 0.50 min later and is held over the following 0.02 min scan. The stem is clipped to 0–100 % after each step. Equation (5.6) makes the valve gain a function of header pressure, so a model identified at 100 % pressure overstates the makeup available at 40 % by a factor of $1/\sqrt{0.4} = 1.58$.

The level transmitter is the only instrument the controller reads, and it is sampled at each controller execution. It adds a constant bias, then noise made by passing a Gaussian sequence through a first-order filter of 0.15 min and rescaling it so that the stationary standard deviation matches the specified value, then a fixed transmitter filter of time constant τ_S, as shown in Equation (5.7).

$$\tau_S \frac{\mathrm{d}\hat{L}}{\mathrm{d}t} = L + b + n(t) - \hat{L} \tag{5.7}$$

where $\hat{L}$ is the measured level, b is the transmitter bias and n is the noise sequence. The filter is advanced by the exact zero-order-hold update $\hat{L}_k = \alpha \hat{L}_{k-1} + (1-\alpha)(L + b + n)$ with $\alpha = \exp(-\Delta t/\tau_S)$, which is stable at any sample period. A separate first-order filter of time constant 1.0 min produces a smoothed rate of change of $\hat{L}$, reported in %/min as the instrument for reading a step test. The makeup, evaporation and blowdown flow indications carry no delay and no filter.

Table 5.2 Model Parameters. Hidden Constants Are Never Displayed and Are the Quantities the Step Test Recovers.

Symbol	Meaning	Value	Unit	Visibility
q_C	circulation rate	3000	gpm	adjustable
ΔT_{nom}	design cooling range	10	°F	adjustable
f_E	evaporation factor	0.85	—	adjustable
f_D	drift fraction	5×10^{-5}	—	adjustable
N	target cycles	5	—	adjustable
C_M	makeup conductivity	400	µS/cm	adjustable
V_0	volume below zero	6000	gal	adjustable
τ_S	transmitter filter	0.05	min	adjustable
—	high-level alarm	85	%	adjustable
—	low-level alarm	40	%	adjustable
—	pump-suction trip	20	%	adjustable
A_b	holdup per percent	22.5	gal/%	hidden
$q_{M,\max}$	makeup valve capacity	75	gpm	hidden
θ	makeup dead time	0.50	min	hidden
τ_v	valve actuator lag	0.33	min	hidden

5.3.4 Dissolved Solids and the Conductivity Switch

Evaporation removes water and leaves its solids behind, so the basin concentrates until the streams that carry solids out balance the solids arriving in the makeup. The solids balance is Equation (5.8).

$$\frac{\mathrm{d}(CV_t)}{\mathrm{d}t} = C_M q_M - C\,(q_B + q_D + q_L) \tag{5.8}$$

where C is the basin conductivity in µS/cm, C_M is the makeup conductivity and $V_t = \max(100\,\text{gal},\ V_0 + V)$ is the total wetted volume with V_0 the basin volume below the transmitter zero. Evaporation appears nowhere on the right of (5.8), which is exactly why the basin concentrates. Expanding the product and substituting (5.1) gives the form that is integrated, as shown in Equation (5.9).

$$\frac{\mathrm{d}C}{\mathrm{d}t} = \frac{q_M\,(C_M - C) + q_E\,C}{V_t} \tag{5.9}$$

The state is floored at C_M, since the basin cannot be cleaner than its makeup. At the design point (5.9) is stationary at $C = NC_M = 2000\,\mu\text{S/cm}$, which is one of the tests in the verification suite shipped with the application. In the on-off blowdown mode this second loop becomes a square-wave load on the level loop, and the period of the square wave is set by V_t, which is why the cycling scenario reduces V_0 from 6000 gal to 1500 gal.

5.3.5 Parameters, Initialization and Numerics

The constants are listed in Table 5.2. Eleven of them are adjustable. Four are hidden and never displayed, and those four set the dynamics that the step test of Section 5.4.2 recovers.

Every scenario opens at a genuine steady state. The volume is initialized to $V(0) = A_b L_{\text{SP}}$, the conductivity to $C(0) = NC_M$, and the valve position to the opening that balances the present losses exactly, as shown in Equation (5.10).

$$x(0) = \frac{100\,(q_E + q_B + q_D + q_L)}{q_{M,\max}\sqrt{\Delta P / \Delta P_{\text{nom}}}} \tag{5.10}$$

At the design point (5.10) returns 42.5 %, and the same expression supplies the flow-balancing aid available in manual mode. Two scenarios override the initial state, one starting the level 8 % below setpoint at $V = 1170$ gal and one starting the conductivity high at 2100 µS/cm.

The integrator is classical fourth-order Runge-Kutta at a fixed 0.005 min (0.3 s) step. LIC-201 executes on a 0.02 min (1.2 s) period, and the historian records one sample at the same 0.02 min interval, 50 samples for every simulated minute. The default seed of the noise generator is 20260825. Section 1.6.1 describes the three clocks, the seeded noise and the speed setting, which this app shares with the others.

5.3.6 Assumptions

The model is deliberately thin on the thermal side and detailed on the water side, and each simplification fails somewhere.

One well-mixed basin, linear in volume. The transmitter span is treated as a prism of constant area, and the water held up in the fill, the distribution deck and the return piping is ignored except through the discrete second-pump event, which removes 3 % of span (67.5 gal) instantly. Surface waves from the returning water and wind are represented only by the measurement noise term. The assumption fails on a basin with a sloped floor or a sump, where the gallons per percent change with level.

Evaporation is algebraic in the heat load. Equation (5.2) has no thermal capacitance, no wet-bulb dependence and no transport lag through the fill, so a heat-load change reaches the water balance at once. A real tower takes a minute or two to respond, and f_E itself falls in cold or humid weather because more of the duty leaves as sensible heat [9, 11]. The assumption overstates how quickly a load upset appears as a level ramp.

Cycles of concentration are the only chemistry. Scaling indices, corrosion, biological control, pH and chemical feed are outside the model, and conductivity stands in for total dissolved solids. The model therefore cannot be used to choose a target N, only to show what holding one costs in water.

The valve is ideal apart from lag and pressure. There is no stiction, no hysteresis, no equal-percentage trim and no finite stroking rate, and the pressure term in (5.6) assumes the valve does not itself destroy the differential it works against. Any of those would change the identified gain and would add a limit cycle that tuning cannot remove.

Dead time is one lumped constant. The delay in (5.5) covers piping transport, valve positioner response and measurement, and it is held fixed. In a plant the transport component falls as makeup flow rises.

The numerics are more repeatable than a control system. An identical trajectory on every machine is convenient and false, because a real level input arrives with jitter, a coarser resolution and a scan order that moves.

5.4 Dynamic Analysis

5.4.1 Transfer Function From Valve to Level

Linearizing (5.1), (5.5) and (5.6) about any operating point gives the plant in Equation (5.11), because the losses contribute nothing to the derivative with respect to L.

$$\frac{L(s)}{U(s)} = \frac{K_I\,\mathrm{e}^{-\theta s}}{s\,(\tau_v s + 1)}, \qquad K_I = \frac{q_{M,\max}/100}{A_b}\sqrt{\frac{\Delta P}{\Delta P_{\mathrm{nom}}}} \tag{5.11}$$

where K_I is the integrating gain in %/min of level per percent of valve. At the design condition $K_I = 0.75/22.5 = 0.0333\,\%/(\mathrm{min}\,\%)$, and the verification suite confirms that value directly by evaluating the derivative at a valve position 10 % above balance. The linearization in (5.11) does not depend on the level, which distinguishes the basin from a drained tank. A gravity-drained tank of area A and outlet coefficient k linearizes to $K_p/(\tau_p s + 1)$ with $\tau_p = 2A\sqrt{L_s}/k$ and a gain that both depend on the operating level. The basin has one gain, one delay and no process time constant, and none of the three moves with level.

Two consequences follow immediately. The steady-state gain of (5.11) is infinite, so there is no process gain to read from a settled step response and no self-regulation. The open-loop phase begins

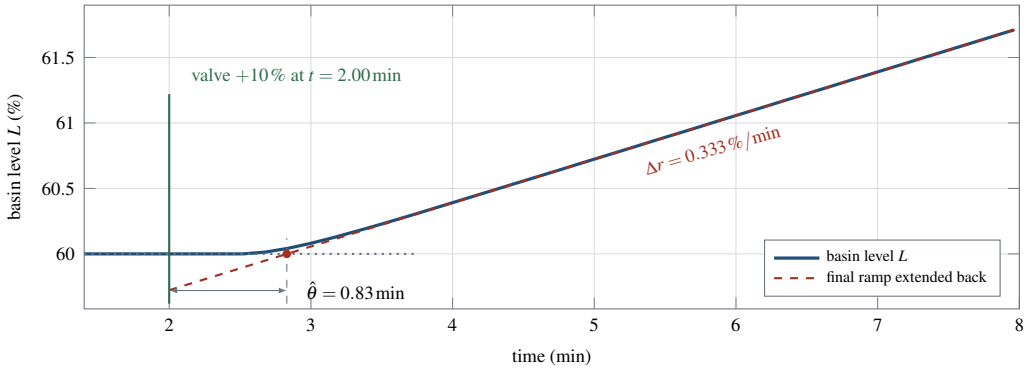


Figure 5.2 Identification of the basin from one open-loop step, recorded by the application. The makeup valve is commanded 10 % open at $t = 2.00$ min from the balanced position. The level departs from its pre-step value by 0.01 % 0.65 min later and reaches a straight ramp of 0.3333 %/min, giving $\hat{K}_I = 0.0333\,\%/(\text{min}\,\%)$. Extending that ramp backward to the pre-step level places the intersection 0.83 min after the step, against a true transport delay of 0.50 min.

at -90° and falls monotonically, so the loop has phase margin only because the dead time is small compared with the speed the tuning asks for.

5.4.2 Identification From the Step-Response Slope

The identification is a 10 % open step on the makeup valve from the balanced position, applied as a calibrated command so that the step size is exact. It is the integrating counterpart of the step test of Section 1.4.4. What is read from the trend is a change of slope and a delay, not a gain and a time constant, as shown in Equation (5.12).

$$\hat{K}_I = \frac{\Delta r}{\Delta u}, \qquad \hat{\theta} = t_\times - t_{\text{step}} \tag{5.12}$$

where Δr is the change in level rate in %/min, Δu is the step in valve percent, and $t_\times$ is the time at which the final ramp, extended backward, crosses the pre-step level. The gain $\hat{K}_I$ is the K' of Equation (1.13). The construction is shown against data recorded by the application in Figure 5.2, which plots the basin level L that the application trends beside the transmitter reading $\hat{L}$. Figure 5.5 shows the application during the same test.

The slope fitted by least squares from 6 to 12 min is 0.3333 %/min against a commanded step of 10 %, which gives $\hat{K}_I = 0.0333\,\%/(\text{min}\,\%)$ and matches the true K_I of (5.11) to four significant figures. The procedure does not require the level to settle, and waiting longer adds no information, because the ramp continues at the same slope until the basin overflows. The application grades the entered $\hat{K}_I$ and $\hat{\theta}$ against the plant and reports each error as a percentage.

5.4.3 Why the Apparent Dead Time Exceeds the Transport Delay

The plant in (5.11) is not IPDT. It carries a pure delay of 0.50 min and, in series with it, a valve lag of 0.33 min. A step test cannot separate the two, because a first-order lag on the input of an integrator produces exactly the same asymptotic ramp as a pure delay of the same magnitude, offset by τ_v. Integrating (5.5) and (5.1) for a step of size Δu at $t = 0$ gives Equation (5.13).

$$L(t) = L_0 + K_I \Delta u \left[t - \theta - \tau_v \left(1 - \mathrm{e}^{-(t-\theta)/\tau_v} \right) \right], \quad t \geq \theta \tag{5.13}$$

The exponential decays, and the surviving asymptote is a straight line of slope $K_I \Delta u$ that crosses L_0 at $t = \theta + \tau_v$. The effective dead time an IPDT fit recovers is therefore the sum, as shown in

Equation (5.14).

$$\theta_{\text{eff}} = \theta + \tau_v = 0.50 + 0.33 = 0.83\,\text{min} \tag{5.14}$$

The value read from Figure 5.2 is 0.83 min, in agreement with (5.14). The same construction applied to the transmitter reading $\hat{L}$ places the intersection at 0.89 min, because the 0.05 min transmitter filter of (5.7) and the 0.02 min scan at which $\hat{L}$ is refreshed add their own lag to the apparent dead time. An alternative and easier reading is the moment the trend visibly departs from the flat line, which occurs 0.65 min after the step at a resolution of 0.01 % and tends toward the true transport delay of 0.50 min as the resolution of the trend improves. That reading is wrong for tuning purposes even though it is correct as a statement about the plant, and Section 5.6 measures what the difference costs. The grading compares $\hat{\theta}$ against θ_{eff} of (5.14) rather than against θ, for that reason. Section 1.4.4 reports the same reading on a self-regulating plant, where a small lag is read almost entirely as dead time.

🕮 **Pause.** Try Exercise 5.2 before reading on.

5.5 Controller Design

5.5.1 Velocity Form PID Algorithm

LIC-201 runs the velocity-form algorithm that the framework apps share (Section 1.6.2), written for this loop in Equations (5.15) and (5.16). The derivative acts on the measurement, so moving the setpoint alone cannot kick the valve (Section 1.4.8).

$$\begin{aligned} e_k &= L_{\text{SP},k} - \hat{L}_k \\ \Delta u_k^P &= K_c\,(e_k - e_{k-1}) \\ \Delta u_k^I &= \frac{K_c}{\tau_I}\,e_k\,\Delta t_c \\ \Delta u_k^D &= -\,K_c\,\tau_D\,\frac{\hat{L}_k^f - 2\hat{L}_{k-1}^f + \hat{L}_{k-2}^f}{\Delta t_c} \end{aligned} \tag{5.15, 5.16}$$

The controller is reverse acting, because the process gain is positive and a rising level calls for less makeup (Section 1.4.3). Derivative comes from a second difference on a filtered measurement, with $\hat{L}_k^f = \alpha \hat{L}_{k-1}^f + (1-\alpha)\hat{L}_k$ and $\alpha = \exp(-\Delta t_c/\tau_f)$, where the filter constant τ_f defaults to 0.05 min. The proportional and derivative increments are formed as differences of the complete terms $K_c e_k$ and $-K_c\tau_D(\hat{L}_k^f - \hat{L}_{k-1}^f)/\Delta t_c$, which equals (5.16) while the gains are constant and makes a retuned gain act at once. The increments are combined, the demand is clipped to the valve span, and the integral is then corrected by back-calculation [17], as shown in Equation (5.17).

$$\begin{aligned} s_k &= s_{k-1} + \Delta u_k^P + \Delta u_k^D \\ c_k &= s_k + I_{k-1} \\ u_k &= \min(100,\ \max(0,\ c_k)) \\ I_k &= I_{k-1} + \Delta u_k^I + \frac{\Delta t_c}{\tau_t}\,(u_k - c_k) \end{aligned} \tag{5.17}$$

Here s_k accumulates the proportional and derivative increments without limit, c_k is the position the algorithm demands with the integral accumulated through the previous scan, u_k is that demand clipped to the valve span, and $\tau_t = \tau_I$ is the tracking time of the back-calculation. These are Equations (1.24) to (1.26) with s_k for S_k, c_k for v_k and no bias. Section 1.4.9 explains why the integral alone is corrected and the running sum is never truncated. With $\tau_t = \tau_I$ the correction cancels Δu_k^I exactly when the clipped excess $c_k - u_k$ equals the proportional term $K_c e_k$, so during a sustained saturation without derivative action the integral holds at the applied output instead of growing, as Section 5.6 measures. The bias is zero and I is seeded at start-up with the initial valve command, so the first

scan contributes the whole positional proportional term. The output carries no rate limit. Transfer between manual and automatic is bumpless in both directions through the tracking of Section 1.4.10. Execution is every 0.02 min, independent of the 0.005 min physics step. Tuning constants are bounded to $0 \le K_c \le 60\,\%/\%$, $0.2\,\text{min} \le \tau_I \le 60\,\text{min}$ and $0 \le \tau_D \le 2\,\text{min}$. The top of the τ_I range is read as an infinite reset time, which removes both the integral increment and the back-calculation and leaves a proportional-plus-derivative loop.

5.5.2 SIMC and Ziegler-Nichols Settings

Both rules read $\hat{K}_I$ and $\hat{\theta}$ exactly as they were entered and never consult the hidden constants, so an identification error propagates directly into the tuning. Equation (5.18) is the SIMC setting for an integrator with dead time [5], which is Equation (1.14) with $\hat{K}_I$ for K'.

$$K_c = \frac{1}{\hat{K}_I(\tau_c + \hat{\theta})}, \qquad \tau_I = 4(\tau_c + \hat{\theta}), \qquad \tau_D = 0 \tag{5.18}$$

where τ_c is the closed-loop time constant and the application uses $\tau_c = \hat{\theta}$, the usual fast-but-robust choice. The reaction-curve PID settings [18] follow from the ultimate gain and period of an IPDT plant under proportional control (Section 1.4.7). The phase of $K_I e^{-\theta s}/s$ reaches -180° at $\omega_u = \pi/(2\theta)$, at which the magnitude is $2K_I\theta/\pi$, giving Equation (5.19).

$$K_u = \frac{\pi}{2K_I\theta}, \qquad P_u = \frac{2\pi}{\omega_u} = 4\theta \tag{5.19}$$

Substituting (5.19) into the classical $0.6K_u$, $P_u/2$, $P_u/8$ settings of Table 1.2 gives the rule the application implements, as shown in Equation (5.20).

$$K_c = \frac{0.94}{\hat{K}_I\hat{\theta}}, \qquad \tau_I = 2\hat{\theta}, \qquad \tau_D = 0.5\hat{\theta} \tag{5.20}$$

Results are rounded to a resolution of 0.1 on K_c and τ_I and 0.01 on τ_D, and clamped to the tuning bounds before they are loaded. With $\hat{K}_I = 0.0333$ and $\hat{\theta} = 0.83$, the effective dead time of (5.14) that the identification of Figure 5.2 recovers to within 0.01 min, (5.18) returns $K_c = 18.1\,\%/\%$ and $\tau_I = 6.6\,\text{min}$, while (5.20) returns $K_c = 34.0\,\%/\%$, $\tau_I = 1.7\,\text{min}$ and $\tau_D = 0.42\,\text{min}$. These are the settings the scenarios preload, and the identified gain is entered to four decimals, so the step-test reading reproduces both.

Equation (5.20) has $\hat{\theta}$ alone in a denominator, with no additive term to soften a bad reading. Reading the dead time as the visible departure of the trend, 0.50 min instead of 0.83 min, raises the gain to 56.4 %/% with $\tau_I = 1.0\,\text{min}$ and $\tau_D = 0.25\,\text{min}$, the tuning scenario 6 loads (from the exact $K_I = 1/30$, while an entry of 0.0333 gives 56.5). Equation (5.19) evaluated with the effective dead time of (5.14) gives $K_u = \pi/(2 \cdot 0.0333 \cdot 0.83) = 56.8\,\%/\%$, so within the IPDT model the rule assumes, the optimistic reading places the proportional gain at 99 % of the ultimate gain. The simulated loop has more margin than that, because part of its apparent dead time is the 0.33 min lag rather than a pure delay. A proportional-only test of the loop, transmitter filter included, reaches sustained oscillation at $K_c = 65.5\,\%/\%$ (beyond the tuning bound of 60) with a period of 3.39 min, so 56.4 is 86 % of the actual ultimate gain. The cycling reported in Section 5.6 comes from pairing that gain with the reset time $\tau_I = 2\hat{\theta} = 1.0\,\text{min}$ that the same reading error sets, since the same gain and derivative settle with $\tau_I = 1.7\,\text{min}$ or with no integral action at all. Equation (5.18) is far less sensitive, because its denominator is $\tau_c + \hat{\theta} = 2\hat{\theta}$ and the same error moves K_c by the same factor from a much lower starting value.

Table 5.3 Effect of Reset Time Alone on a 5 % Setpoint Move at $t = 1$ min, $K_c = 18.1\,\%/\%$ and $\tau_D = 0$ Throughout, Read at $t = 26$ min. Settling Is Into the $\pm 2\,\%$ Band.

τ_I (min)	6.6	2.0	1.0
Overshoot (%)	26.2	49.7	53.7
Settling time (min)	2.38	6.44	13.48
IAE (% min)	18.00	22.76	38.37
Valve travel (% of span)	132.3	249.3	625.2
Time saturated (%)	8.9	12.1	33.8

TRY IT IN THE APP 6 · Aggressive Tuning, Optimistic Dead Time

Load the case, which preloads $K_c = 56.4$, $\tau_I = 1.0$ min and $\tau_D = 0.25$ min from a dead time read as 0.50 min, and run it to 30 min. After the heat load rises at 5 min, watch the makeup valve trend: from 20 min it swings between 4.7 and 100 % while the level stays between 59.09 and 60.84 %. Then Reset, enter $\hat{\theta} = 0.83$ min and choose the ZN PID rule before 5 min.

5.5.3 Proportional Offset and Reset Time

The pole at the origin also changes what proportional action alone can achieve. With $\tau_I \to \infty$ and the delay neglected, the loop transfer from setpoint to level is $K_c K_I/(s + K_c K_I)$, which is unity at $s = 0$, so a proportional controller on an integrating process tracks a setpoint change with no offset at all. A self-regulating plant cannot do that (Equation (1.3)). The offset reappears for load changes, and its size is worth computing because it explains the peak deviations reported later. Writing a sustained load as an equivalent valve demand d in percent, the steady level satisfies Equation (5.21).

$$L_\infty - L_{\mathrm{SP}} = -\frac{d}{K_c} \tag{5.21}$$

A 30 % rise in heat load raises evaporation from 25.50 to 33.15 gpm and, through (5.4), blowdown from 6.225 to 8.14 gpm, a total of 9.5625 gpm that the valve must supply. Dividing by the valve capacity of 0.75 gpm per percent gives $d = 12.75\,\%$, so (5.21) predicts a residual deviation of 0.70 % at $K_c = 18.1$ and 0.375 % at $K_c = 34$ before integral action removes it. With the reset time at the top of its range, so that the loop is proportional only, the level settles 0.704 % and 0.375 % below setpoint at those two gains after the load step, as (5.21) predicts.

Integral action is therefore needed only to erase (5.21), and it carries a cost. A PI controller on (5.11) places two open-loop poles at the origin, so the open-loop phase starts at -180° and the loop is stable only because the PI zero at $1/\tau_I$ supplies phase lead below the crossover frequency. Shortening τ_I moves that zero above the crossover, the phase lead is lost, and the loop oscillates slowly about the setpoint instead of settling. This is the mechanism behind the long reset times that tuning rules written for integrating plants recommend [1, 6], and behind the factor of four in (5.18). Section 1.4.6 finds the same trend on the running example of Chapter 1. Table 5.3 measures it on a 5 % setpoint move with the proportional gain held at 18.1 %/%.

Cutting τ_I from the SIMC value of 6.6 min to 1.0 min multiplies the settling time by 5.7, from 2.38 to 13.48 min, and the valve travel by 4.7, from 132.3 to 625.2 % of span, on a loop whose proportional gain never changed. At $\tau_I = 1.0$ min the level stays inside the band after 13.48 min but never stops rolling, with an amplitude of 1.9 % about the setpoint and the valve swinging between 0 and 94 % from 16 to 26 min, so its travel keeps growing with the length of the run. This failure mode is particular to integrating processes, and it reverses the intuition a first-order plant teaches, where reducing τ_I shortens the response over a wide range.

> ✦ DID YOU KNOW? **A Shorter Reset Time That Slows the Loop**
>
> From the simulator: with the gain held at 18.1 %/% on a 5 % setpoint move, cutting the reset time from 6.6 to 1.0 min lengthens the settling time from 2.38 to 13.48 min and raises the valve travel from 132.3 to 625.2 % of span (Table 5.3).

5.5.4 Model Predictive Control on the Identified Integrator

The same two identified numbers also serve as the prediction model of a third design, an MPC that moves LCV-201 directly. Only the comparison case, scenario 10, starts under the MPC. It is the receding-horizon controller of Section 1.6.3 [19, 20], and it makes a decision every $T_s = 0.1$ min, five controller executions. Its model is Equation (5.11) with the actuator lag folded into the dead time, the IPDT form the step test identifies, plus an unmeasured input disturbance, as shown in Equation (5.22).

$$\hat{L} - \hat{L}_0 = z, \qquad \frac{\mathrm{d}z}{\mathrm{d}t} = \hat{K}_I \left[u(t - \hat{\theta}) - u_0 + \delta \right] \tag{5.22}$$

where u_0 and $\hat{L}_0$ are the command and the LT-201 reading when the estimator starts, z is the predicted level deviation in percent and δ is a load the model cannot measure, expressed in percent of valve. On an integrator a constant δ is a ramp rather than an offset, so a heat-load change, a leak, a sagging header and an error in $\hat{K}_I$ all register as makeup that is not arriving, and the load d of Equation (5.21) appears as $\delta = -d$. The MPC reads $\hat{K}_I$ and $\hat{\theta}$ as they were entered for the rules, so a misread step test misleads it as it misleads them, and with either entry at zero the PID keeps the valve. A least-squares IPDT fit to ± 10 % valve steps on the simulator, without noise, returns $\hat{K}_I = 0.03324$ and $\hat{\theta} = 0.873$ min on the transmitter with a root-mean-square residual of 0.19 % of the response, the transient of the lag that the IPDT form cannot follow, and the comparison case uses the 0.0333 and 0.83 min read from Figure 5.2. The dead time is a record of the last $D = \mathrm{round}(\hat{\theta}/\Delta t_c) = 42$ commands sent.

A steady-state Kalman filter on (z, δ), the integrator sampled at Δt_c and augmented with δ as a random walk, which is the usual route to offset-free predictive control [21] and the subject of Volume 9, Chapter 4, runs at every execution whichever design holds the valve, as shown in Equation (5.23).

$$\begin{aligned} \bar{z}_k &= \hat{z}_{k-1} + \hat{K}_I \Delta t_c \left(u_{k-1-D} - u_0 + \hat{\delta}_{k-1} \right) \\ \nu_k &= \hat{L}_k - \hat{L}_0 - \bar{z}_k \\ \hat{z}_k &= \bar{z}_k + \ell_z \nu_k, \qquad \hat{\delta}_k = \hat{\delta}_{k-1} + \ell_\delta \nu_k \end{aligned} \tag{5.23}$$

where ν_k is the innovation and the gains $\ell_z = 0.1064$ and $\ell_\delta = 8.977$ percent of valve per percent of level follow from noise variances r, $10^{-6} r$ and $\rho\, r / \hat{K}_I^2$ on the measurement, z and δ, with $\rho = 0.1$, which gives the estimation error a time constant of 0.36 min. At rest the plan holds the valve at $u_0 - \hat{\delta}$, so $\hat{\delta}$ supplies the integral action, and after the heat load of scenario 5 it settles at -12.750 %, the $-d$ computed for Equation (5.21).

Each decision starts from the deviation z_0 that the D commands in transit will leave, as a Smith predictor would, and solves the quadratic program of Equation (5.24) on the model resampled at T_s.

$$\begin{aligned} \min_{v,\varepsilon} \; & \frac{T_s}{1\,\mathrm{min}} \sum_{j=1}^{N_p} \left(\frac{\hat{L}_0 + z_j - L_{\mathrm{SP}}}{1\,\%} \right)^2 \\ & + \lambda \sum_{i=1}^{12} \left(\frac{v_i - v_{i-1}}{5\,\%} \right)^2 + c_\varepsilon \varepsilon^2 \\ \text{s.t. } & z_j = z_{j-1} + \hat{K}_I T_s \left(v_{i(j)} - u_0 + \hat{\delta}_k \right) \\ & 0 \le v_i \le 100\,\%, \qquad \varepsilon \ge 0 \\ & L_{\mathrm{lo}} - \varepsilon \le \hat{L}_0 + z_j \le L_{\mathrm{hi}} + \varepsilon, \quad j = 1, 4, \ldots, 148 \end{aligned} \tag{5.24}$$

Table 5.4 Scenario Sequence and the Mechanism Each Exercises

#	Scenario	What it exercises
1	Operate by hand, $u = 30\,\%$	no self-regulation, exact flow matching
2	Step test from balance	IPDT identification from a slope
3	First automatic loop, $K_c = 3$	proportional recovery from 8 % low
4	SIMC PI installed	model-based tuning, reset time
5	$+30\,\%$ heat load at $t = 5$	load rejection on an integrator
6	Ziegler-Nichols from $\hat{\theta} = 0.50$	identification error, not rule error
7	On-off conductivity blowdown	square-wave load from a second loop
8	Level noise $\sigma = 0.4\,\%$	derivative action on a ramping PV
9	130 % load, 40 % pressure, 12 gpm leak	makeup capacity limit, anti-windup
10	Three controllers from one model	SIMC, ZN and MPC from one fit, move suppression
11	Graded challenge, four upsets	the whole workflow under time

where z_j is the deviation j steps beyond the dead time, $v_1, \ldots, v_{12}$ are valve positions held over blocks of $(1,1,1,1,2,2,3,4,6,9,15,105)$ steps that cover $N_p = 150$ steps or $T_p = 15\,\text{min}$, $v_{i(j)}$ is the block containing step j, v_0 is the command applied now, $\lambda = 0.3$ is the move suppression, ε is one slack priced at $c_\varepsilon = 10^3$ per $\%^2$, and $L_{\text{lo}} = \min(40\,\%, L_{\text{SP}} - 2\,\%)$ and $L_{\text{hi}} = \max(85\,\%, L_{\text{SP}} + 2\,\%)$ are the level alarms kept 2 % from the setpoint. With these scales $\lambda = 1$ charges a 5 % valve move like a 1 % level error held for one minute. Only the valve range is a hard constraint. There is no terminal weight, because the last block holds the valve for 10.5 min and a plan that leaves the basin ramping pays for the ramp over that whole block. The dual active-set solver [22] that the apps share handles the resulting 13 variables and 125 inequality rows.

The period, the horizon, λ and ρ were set once, on the comparison case, by the tuning rule common to the predictive controllers of the apps. Of 75 settings (λ from 0 to 10, ρ from 0.01 to 100), 53 finish the case without a rejected plan and with at most the 150.2 % of valve travel of the SIMC PI, and among those within 10 % of the lowest IAE (13.06 % min) the largest λ and then the smallest ρ give $\lambda = 0.3$ and $\rho = 0.1$ (14.06 % min). A solution is discarded if the solver does not certify it as optimal, if its first move is not finite, or if that move leaves the 0–100 % range, and the PID, held on the applied command at every execution as in manual mode, takes the valve without a bump (0.034 % when a rejection is forced) until the next acceptable plan. The MPC reads only LT-201, the applied command, the setpoint, the two alarm limits and its own tuning, so the plant of Equation (5.11), the identified pair of Equation (5.12) and the model (5.22) stay separate, and a change to the header or to a hidden constant is model mismatch for the MPC.

5.6 Results

Eleven scenarios, listed in Table 5.4, are arranged as a laboratory sequence, and Table 5.5 gives the requirement for completing each. Section 1.6.4 describes what a scenario fixes and how its completion is judged. Reset restarts a case from $t = 0$ with the reader's own settings kept, while reloading the case restores its shipped settings, so the comparisons in Table 5.5 are made by changing one setting between restarts. The challenge is judged on the case as given, and changing a model parameter or a disturbance voids the attempt. A run exports as a CSV record at one sample every 0.02 min (1.2 s) of simulated time. The record holds the true level beside the measured one and the commanded valve beside the actual stem, so the slope fit behind Figure 5.2 can be reproduced in a spreadsheet or in Python.

Scenario 1 makes the point with arithmetic before any controller is involved. A valve at 30 % delivers 22.5 gpm against 31.875 gpm of losses, a deficit of 9.375 gpm that by (5.1) is a level ramp of –0.417 %/min. From 60 % the basin reaches the pump-suction trip at 20 % in 96.8 min and empties in 144.8 min, the ramp times plus the 0.83 min effective dead time, and no valve position other than 42.5 % stops the motion. Scenario 3 shows the same physics under proportional-plus-reset

Table 5.5 Scenarios and Completion Requirements

#	Scenario	Kind	Required to complete
1	Operate by hand	Experiment	In Manual, hold the level rate within ±0.05 %/min of zero for 1 min, change the heat load, rebalance the valve, and hold the rate within ±0.01 %/min for 0.5 min at the new load.
2	Step test	Experiment	In Manual, step the valve from balance to at least 52 % after the first minute and hold it for 3 min, then enter $\hat{K}_I$ within 25 % of K_I (0.025 to 0.0417) and $\hat{\theta}$ within 25 % of $\theta + \tau_v$ (0.62 to 1.04 min).
3	First automatic loop	Experiment	Recover from 8 % low to 59.8 % with $K_c = 3$ in Auto, then Reset, apply the SIMC rule to the identified model before $t = 0.5$ min, and recover again with $K_c \geq 9$.
4	SIMC tuning, reset time	Experiment	Raise the setpoint to 65 % in Auto after 0.5 min and run 5 min with $\tau_I \geq 6$ min, then repeat the step in runs of the reloaded case with $\tau_I = 2$ min for 8 min and 1 min for 15 min.
5	Heat-load disturbance	Experiment	Run through the +30 % heat-load step at 5 min until the level has stayed within 0.1 % of setpoint for 2 min.
6	Optimistic dead time	Experiment	Run the loaded tuning ($K_c \geq 50$, $\tau_I \leq 1.05$ min) to $t = 25$ min, then Reset, apply the ZN rule from $\hat{\theta} \approx 0.83$ min before the step at 5 min, and run to $t = 25$ min with $25 \leq K_c \leq 45$ and $\tau_D > 0$.
7	On-off blowdown	Experiment	With conductivity-switched blowdown, run past 18 min until the bleed has been shut for 5 min, then switch to continuous blowdown and run 10 min more.
8	Noisy level	Experiment	With $\sigma = 0.4$ %, run 10 min at $\tau_D = 0.4$ min and $\tau_f \leq 0.06$ min, 10 min after Reset with $\tau_D = 0$, and 10 min from the reloaded case with $\tau_f \geq 0.3$ min, then raise the noise above 0.45 %.
9	Valve saturation	Demonstration	Hold the valve at 100 % for 1 min while the leak is on, run past the repair at 18 min until the level is back above 59.5 %, then Reset with the header pressure restored to at least 95 % before $t = 0.5$ min and, from the repair at 18 min, hold the level within 2 % of setpoint, unsaturated, for 10 min.
10	Three controllers	Experiment	Run the setpoint step at 1 min and the heat-load step at 10 min to $t = 20$ min under the MPC with $\lambda \geq 0.25$, again after a Reset with $\lambda \leq 0.05$, then under the SIMC PI after a Reset with $K_c \geq 9$.
11	Challenge	Challenge	At $t = 45$ min, a worst level deviation over the whole run of at most 2 % from the 60 % setpoint, no overflow, no time saturated and less than 250 % of valve travel, with the model parameters and disturbances as given.[a]

[a] With the shipped tuning ($K_c = 3$, $\tau_I = 30$ min) the worst deviation is 5.73 %, at $t = 25$ min, and the case fails although the valve never saturates and travels only 29 %. With the suggested SIMC tuning ($K_c = 18.1$, $\tau_I = 6.6$ min) the worst deviation is 0.70 % with 66 % of travel and no saturation, inside the band with 1.30 % to spare.

control. Starting 8 % low with $K_c = 3$ and $\tau_I = 30$ min, the loop enters the ±2 % band for good after 10.58 min, first reaches 60 % after 17.2 min and accumulates an IAE of 58.37 % min by $t = 20$ min, then overshoots to 61.41 % at 34.4 min, 17.6 % of the initial error. The SIMC tuning ($K_c = 18.1$, $\tau_I = 6.6$ min) reaches 60 % after 5.0 min and settles within 3.98 min after a peak of 61.75 % at 7.1 min, with an IAE of 33.27 % min at $t = 20$ min, while resting against a valve limit for 20.9 % of those 20 min. The saturation figure counts time at either end of the travel, because a makeup valve pinned shut costs the loop as much authority as one pinned open.

Table 5.6 Measured Performance After a +30 % Heat-Load Step at $t = 5$ min, Read at $t = 20$ min. All Four Controllers Act on the Same Plant With $K_I = 0.0333\,\%/(\text{min}\,\%)$ and $\theta_{\text{eff}} = 0.83$ min.

	Default	SIMC PI $\hat{\theta} = 0.83$	ZN PID $\hat{\theta} = 0.83$	ZN PID $\hat{\theta} = 0.50$
K_c (%/%)	3	18.1	34	56.4
τ_I (min)	30	6.6	1.7	1.0
τ_D (min)	0	0	0.42	0.25
IAE (% min)	30.90	4.42	1.04	4.72
ISE (%^{2} min)	75.30	2.01	0.25	1.89
Max. deviation (%)	3.06	0.70	0.40	0.64
Valve travel (% of span)	12.23	20.47	31.68	474.7
Time saturated (%)	0	0	0	0

TRY IT IN THE APP 1 · Operate the Basin by Hand

Load the case and press Run without touching the valve. It stays at 30 % while evaporation, blowdown and drift carry away more water than it supplies. Watch the basin level trend and the Level rate tile: the level falls on a straight line at –0.417 %/min and never levels off. Only a valve at 42.5 % would stop it.

5.6.1 Load Rejection With Four Tunings

Four controllers meet the identical upset in Table 5.6, which reports the performance measures the application computes for itself. The heat load steps from 100 to 130 % at $t = 5$ min from a settled start and every figure is read at $t = 20$ min. Two of the four follow from applying a rule to a correct identification, one is the cautious default tuning, and one follows from supplying Equation (5.20) with the dead time a trend appears to show.

Reading Table 5.6 across the first three columns gives a clean and unsurprising trade. More gain buys a smaller deviation and pays for it in valve travel, with the maximum deviation falling from 3.06 to 0.40 % while the travel rises from 12.23 to 31.68 % of span. The measured peaks also confirm Equation (5.21), since the proportional offset predicted for a 12.75 % load is 0.70 % at $K_c = 18.1$ against a measured 0.70 %, and 0.375 % at $K_c = 34$ against a measured 0.40 %.

The fourth column is a different kind of result. Its IAE of 4.72 % min is indistinguishable from the SIMC value of 4.42 % min and its peak deviation is smaller, yet its valve travel is 474.7 % of span against 20.47 %, a factor of 23. The loop is not rejecting the disturbance. It is cycling. Continuing both runs to $t = 40$ min separates them completely, because the SIMC IAE grows from 4.42 to 4.65 % min and its travel from 20.47 to 20.73 % while the aggressive loop grows from 4.72 to 15.64 % min and from 474.7 to 1485 %, and begins to hit the valve limits for 7.2 % of the time. Over $20 \le t \le 30$ min the makeup valve swings between 4.7 and 100.0 % with a period of 3.75 min, against the 3.32 min that (5.19) predicts for P_u at $\theta_{\text{eff}} = 0.83$ min and the 3.39 min of the proportional-only test. The level itself only moves between 59.09 and 60.84 %, which is the reason the error measures look respectable and the reason a cycling level loop can persist unnoticed in operation.

The two Ziegler-Nichols columns separate identification error from rule error. Both were produced by the same rule and the same arithmetic. One is stable and the fastest controller in the table, and the other carries 99 % of the ultimate gain the IPDT model predicts together with a reset time cut by the same factor, so the degradation comes from the identification and not from the rule. Rules that place $\hat{\theta}$ in an additive denominator, such as (5.18), are tolerant of the same reading error by construction. Figure 5.3 draws the four columns of Table 5.6 side by side.

5.6.2 Derivative Action on a Noisy Level

Scenario 8 raises the level noise to $\sigma = 0.4$ % and switches derivative action on at $\tau_D = 0.4$ min. Read at $t = 30$ min, the valve travel is 11 791 % of span with an IAE of 2.27 % min, and the valve touches

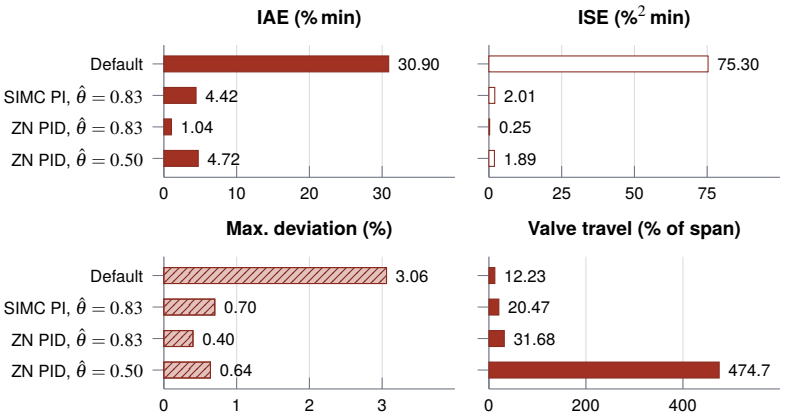


Figure 5.3 The four tunings of Table 5.6 after the +30 % heat-load step, read at $t = 20$ min: integrated absolute error and valve travel (solid bars), integrated squared error (open) and largest deviation (hatched). Notice that the tuning from the misread dead time keeps its error measures close to the SIMC values while its valve travel is 23 times larger, the signature of a cycling loop.

a limit for 1.6 % of the run. Removing derivative action entirely gives 1669 % of travel and an IAE of 2.43 % min, so derivative action obtained a 6.6 % reduction in accumulated error at a cost of 7.1 times the valve motion. Raising the derivative filter from its default 0.05 min to the maximum 0.5 min recovers most but not all of the difference, at 2982 % of travel. The mechanism is in (5.11). Derivative action anticipates, and the level of an integrating plant is already a ramp whose slope changes only when a flow changes, so the derivative term amplifies the measurement noise without adding useful anticipation. The same term adds useful phase lead on the temperature loop of Section 1.4.8, and the scenario supplies the contrast.

Pause. Try Exercise 5.3 before reading on.

Scenario 7 adds the second loop. With the blowdown valve switched by conductivity, an on-off valve passing three times the design bleed lands on the level loop as a repeating load step of 18.7 gpm at every switching, with the bleed open for 9.2 min and shut for 17.6 min once the first cycle has passed. Read at $t = 30$ min the level loop holds a sawtooth with a maximum deviation of 1.35 % and an IAE of 16.83 % min, using 79.05 % of valve travel and never saturating. The interaction is one-way and the basin absorbs it, which is what a buffer volume exists to do.

5.6.3 Makeup Capacity Limit

Scenario 9 applies a 130 % heat load, a 12 gpm basin leak and a makeup header at 40 % of nominal pressure at the same time. By (5.6) the valve can deliver at most $75\sqrt{0.40} = 47.4$ gpm, and the losses total 53.4 gpm, so the level falls at −0.267 %/min with the valve wide open and no controller action can arrest it. Read at $t = 30$ min the measured valve travel is 0.00 % with 100 % of the time saturated, an IAE of 81.68 % min and a maximum deviation of 4.80 %. A valve travel of exactly zero on a loop with a large error is the diagnostic signature of a capacity problem rather than a tuning problem, and separating the two is a routine diagnostic task in practice. Chapter 3 draws the same line between an error that feedback removes and one that an actuator bound imposes. The leak is repaired at $t = 18$ min, after which the same fully open valve delivers a surplus of 6.0 gpm and the level climbs back into the 2 % band 28.52 min after the run started. Throughout the saturated period the back-calculation term of (5.17) holds the integral at the applied 100 %, its fixed point with $\tau_t = \tau_I$, so the valve stays fully open until the measured level reaches the setpoint at 36.1 min, and the level peaks only 0.58 % above setpoint at 39.8 min. A counterfactual run with the back-calculation removed winds the integral to 324 % by $t = 30$ min, holds the valve open until 55.8 min and overshoots to

Table 5.7 Measured Comparison of SIMC PI, ZN PID and MPC Designed From One Identified Integrating Model

Case	Design	IAE (% min)	OS (%)	t_s (min)	e_{max} (%)	Travel (%)	Sat (%)	Plans
10, SP, load	SIMC PI	18.02	26.2	2.38	0.54	150.2	11.6	—
	ZN PID	16.91	39.0	2.38	0.58	227.4	16.3	—
	MPC	14.06	0.5	2.68	0.57	144.1	1.5	200
	MPC, $\lambda = 0$	12.95	6.1	2.38	0.50	2659.4	79.0	200
5, load	SIMC PI	4.42	—	—	0.70	20.5	0	—
	ZN PID	1.04	—	—	0.40	31.7	0	—
	MPC	1.58	—	—	0.56	27.1	0	200
6, $\hat{\theta} = 0.5$	ZN PID	15.64	—	—	0.96	1485.3	7.2	—
	SIMC PI	1.70	—	—	0.55	37.7	0	—
	MPC	2.04	—	—	0.58	61.5	0	400
	MPC, $\lambda = 0$	17.83	—	—	0.97	2456.4	45.5	400
3, 8 % low	SIMC PI	33.27	21.9	—	—	78.9	20.9	—
	MPC	25.09	0.0	—	—	94.5	11.5	200
8, noise	PID, $\tau_D = 0.4$	2.27	—	—	0.21	11790.9	1.6	—
	PI, $\tau_D = 0$	2.43	—	—	0.20	1669.1	0	—
	MPC	4.92	—	—	0.47	1412.4	0	300
	MPC, $\lambda = 3$	3.25	—	—	0.30	741.6	0	300
7, blowdown	SIMC PI	16.83	—	—	1.35	79.0	0	—
	MPC	6.15	—	—	1.10	104.5	0	300
9, capacity	SIMC PI	90.31	—	—	4.80	16.9	80.2	—
	MPC	87.11	—	—	4.80	16.4	74.7	450

IAE, travel and Sat are scorecard values over 20, 20, 40, 20, 30, 30 and 45 min in case order. OS is the overshoot in percent of the step (case 3: of the initial error), t_s the time from the step until the level stays within ±2 %, e_{max} the largest deviation after the load, Sat the percentage of time at either valve stop, and Plans the number of MPC decisions, all accepted. In case 6 every design comes from $\hat{\theta} = 0.50$ min.

65.56 %. A recursion that absorbed the whole clipped excess at every scan would instead unwind the integral to 14 % while saturated, release the valve at 29.5 min and approach the setpoint from below with no overshoot, reaching 59.5 % 0.8 min later than the shipped controller. With the header restored before the run, the same tuning rides out the repair with a 0.88 % peak and no saturation.

The graded challenge applies four upsets over 25 min, a +30 % heat load at $t = 4$, a 5 gpm leak at $t = 12$, a 25 % loss of header pressure at $t = 20$ and a return to normal load at $t = 25$, and is judged at $t = 45$ min on the whole run. The level must stay within ±2 % of the 60 % setpoint with no overflow, no time saturated and less than 250 % of valve travel, with the model parameters and disturbances as given. Starting from the cautious default the challenge is not met, with a worst deviation of 5.73 % at $t = 25$ min, nearly three times the band, an IAE of 134.62 % min and 29 % of valve travel. The SIMC tuning obtained from the step test meets every part of the target on the same disturbance sequence, with a worst deviation of 0.70 %, an IAE of 12.62 % min, 66 % of valve travel and no saturation, which leaves 1.30 % of margin on the band and 184 points on the travel limit.

5.6.4 Three Designs From One Identified Model

The SIMC PI, the ZN PID and the MPC all follow from $\hat{K}_I = 0.0333$ and $\hat{\theta} = 0.83$ min and are compared in Table 5.7. Every case starts from its own seed and instruments, the design is selected before the run, the PI rows keep the tuning of the case or the gains a rule installs, the MPC keeps the settings of Section 5.5.4 everywhere, and no design receives a measured disturbance. Scenario 10 raises the setpoint from 60 to 65 % at 1 min and the heat load by 30 % at 10 min, and every row is reproduced exactly after a restart from $t = 0$.

A 5 % rise of an integrating level needs a fixed volume of makeup, $5/\hat{K}_I = 150$ % min of valve above the balance, and the designs differ in when they deliver it. The SIMC PI kicks the valve by $18.1 \times 5 \approx 90$ %, holds it at 100 % from 1.02 to 3.32 min and overshoots to 66.31 %. The MPC counts the water already in transit, holds 100 % only from 1.72 to 2.00 min and starts closing at 2.02 min with the level still at 60.26 %, so the level peaks at 65.03 %, as shown in Figure 5.4. Its IAE is 14.06 against 18.02 and 16.91 % min for SIMC and ZN, with less travel and 1.5 against 11.6 and 16.3 %

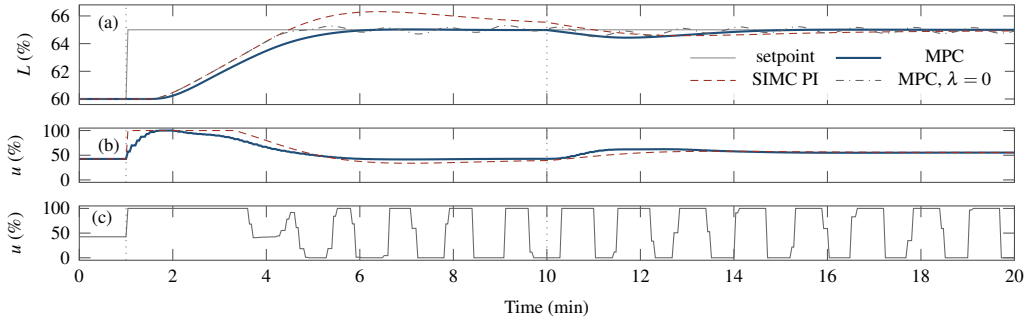


Figure 5.4 Comparison case under the MPC, the SIMC PI and the MPC with $\lambda = 0$, all three designed from $\hat{K}_I = 0.0333\,\%/(\text{min}\,\%)$ and $\hat{\theta} = 0.83\,\text{min}$. The setpoint rises from 60 to 65 % at 1 min and the heat load by 30 % at 10 min (dotted vertical lines). (a) True level under the MPC (thick solid), the SIMC PI (dashed) and the MPC with $\lambda = 0$ (dash-dotted), with the setpoint (thin solid). (b) Valve command under the MPC (thick solid) and the SIMC PI (dashed). (c) Valve command under the MPC with $\lambda = 0$. At $t = 20\,\text{min}$ the scorecard IAE is 14.06, 18.02 and 12.95 % min and the travel 144.1, 150.2 and 2659.4 %.

of the time at a limit, but the approach without overshoot enters the ±2 % band 0.30 min later. The setpoint step of scenario 4 alone gives the same order (12.49 against 18.00 and 16.08 % min).

Move suppression is what keeps the plan usable on an integrator. With $\lambda = 0$, 158 of the 200 plans put the first move on a bound and the valve strokes from one limit to the other every 1.23 min, at a limit for 79.0 % of the run with 2659 % of travel for an IAE only 8 % lower, as shown in Figure 5.4(c). The strokes come from the actuator lag folded into $\hat{\theta}$. By Equation (5.13) a valve move Δu carries the level ahead of the IPDT prediction by up to $K_I \Delta u\, \tau_v/\text{e}$, 0.41 % for a full stroke, the filter books that excursion as a change of δ within its 0.36 min time constant, and a plan with free moves answers with a full stroke back, which a hundredfold smaller ρ also prevents. Without the lag (0.01 min, with the delay raised to keep $\theta + \tau_v$) the same plan travels 298 %, $\lambda = 0.05$ alone brings the shipped plant to 165 %, and on the misread fit of scenario 6, $\lambda = 0$ turns a settled MPC into a cycle between the stops. Larger prices trade error for travel in the usual way. Scenario 10 repeated with a move suppression of 0.1, 1 and 3 gives an IAE of 12.86, 18.11 and 25.29 % min with 154.9, 110.4 and 78.6 % of travel. The model sets whether the plan is right, and the price on valve moves sets whether the valve can live with it.

Three cases are lost outright. On the unmeasured load of scenario 5 the ZN PID keeps the lower IAE and dip (1.04 against 1.58 % min, 0.40 against 0.56 %). With $K_c = 34$ and derivative action it has the valve past the new balance of 55.25 % half a minute after the load, while the move suppression holds the MPC at 46.1 % although $\hat{\delta}$ has already covered 63 % of the load. With $\hat{\theta}$ read as 0.50 min the SIMC rule returns a faster loop ($K_c = 30$) that rejects the load best of all (1.70 against 2.04 % min), while the MPC settles with its travel more than doubled against the correct fit (61.5 against 27.1 %). On the noisy level of scenario 8 the filter passes the noise into $\hat{\delta}$ (standard deviation 17.7 % of valve) and the held moves swing the valve slowly (standard deviation of one-minute averages 7.75 against 2.73 % for the PI without derivative), which the basin integrates into a root-mean-square level error of 0.21 against 0.10 % and an IAE of 4.92 against 2.43 % min.

The MPC wins where the plan can use what it knows about the integrator. From 8 % low it refills without overshoot (peak 60.00 against 61.75 %, IAE 25.09 against 33.27 % min). Under the on-off blowdown $\hat{\delta}$ swings over 27.3 % of valve, about the 24.9 % that one switching of the 18.7 gpm bleed represents, so each switching is canceled as a step of δ (IAE 6.15 against 16.83 % min for 32 % more travel). At the capacity limit the MPC has no integrator to unwind and leaves 100 % at 33.62 min with the level at 59.36 %, 2.48 min before the PI, which lowers the recovery peak from 60.58 to

60.13 %, with 335 of its 450 plans on the upper bound and at most 12 active-set iterations. On the challenge, reported for information, both designs meet the target, the MPC with an IAE of 5.63 against 12.62 % min for the suggested SIMC tuning.

A dead time read too long is tolerated far less well by the MPC than by the SIMC rule. On scenario 10 run to 40 min at $\lambda = 0.3$ the MPC settles for $\hat{\theta}$ up to 1.1 min, hunts at 1.15 and 1.2 min and cycles between the stops from 1.25 min, and at $\lambda = 3$ it cycles from 1.7 min, while the SIMC PI, whose gain falls as $1/(2\hat{K}_I\hat{\theta})$, settles up to 3 min. The valve travel over those 40 min is 1228 % with $\hat{\theta}$ of 1.2 min and 3307 % with 1.5 min, while the SIMC PI from the same readings settles. With $\hat{\theta}$ longer than the plant's delay the level answers before the model expects it and the filter books the early water as a load (after a 10 % valve step with $\hat{\theta} = 3$ min, $\hat{\delta}$ reaches 10.0 % of valve by 3.5 min), so the plan counts the same water twice. The cycle disappears when the plant's delay is lengthened to match $\hat{\theta}$ and, over the setpoint step, when the estimate is replaced by the bare measurement, and the onset lies where $\hat{\theta}$ exceeds 0.83 min by about the 0.36 min time constant of the filter. Read short, the order reverses, and at $\hat{\theta} = 0.2$ min the MPC hunts while the SIMC PI cycles at its gain bound of 60. All 3920 plans of these runs, scenario 4 and the challenge included, were optimal and the soft band never became active, at about 0.1 ms per plan in Node.js and 0.28 to 0.34 ms in Chrome, 0.3 % of one processor core at the default speed (wall-clock values on one desktop computer).

5.7 Technology in Practice

The loop of Section 5.3 reads LT-201 through a noise term and a transmitter filter and moves LCV-201 through a delay and a lag. Figure 5.6 draws it with the devices a plant basin would carry in the place of the schematic's symbols.

What Measures It

A basin level is read by a hydrostatic transmitter near the floor of the basin or by an echo transmitter mounted above the water. No range, accuracy or response time is given here, because none was found in an open source that names no maker. The noise $n(t)$ of Equation (5.7) stands for the waves that the falling return water raises on the surface, the filter τ_S for the damping that the transmitter or a stilling well adds, and the bias b for a calibration error that no controller can see.

> **TECHNOLOGY** **Sensor: A Level Transmitter Over a Splashing Basin**
>
> A hydrostatic transmitter infers level from the pressure at the bottom of the liquid column, which grows with the height and the density of the liquid, so its calibration rests on that density. An echo transmitter times a sound or radio wave from the top of the vessel to the surface and back. It is immune to density, but foam can fool it, and liquid pouring in through the space above, as the return water rains into a basin, may also cause problems. A turbulent surface is especially hard on echo sensors. A stilling well, a vertical pipe around the sensor, shields it from the splashing and the waves in the rest of the vessel [23].

What Moves It

LCV-201 of Equation (5.6) is a linear valve whose stem follows the delayed command through a lag, and whose flow falls with the square root of the header pressure. That pressure term is the plant's dependence of flow on supply pressure, and the clipped stem is a valve that cannot reach the position the controller asks for. A controller with reset action then winds up in a futile attempt to drive the valve where it cannot go, unless the final element feeds its real position back to the PID block by back-calculation [23], the role of Equation (5.17). The blowdown valve of case 7 is a second final element. A conductivity controller measures the tower water continuously and opens it only when the conductivity passes its set point [13].

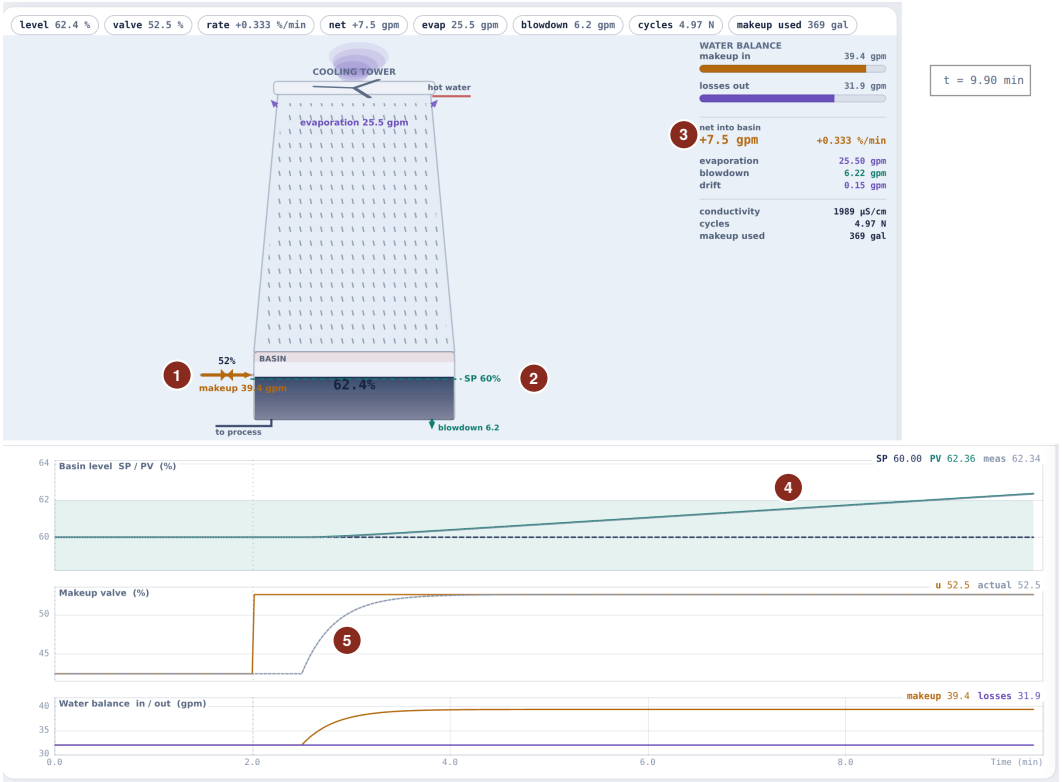


Figure 5.5 Step-test identification in progress in scenario 2 at $t = 9.9$ min, a composite of the application window: the plant view with its readouts above the three trends, and the elapsed-time readout of the header in the gray frame at the upper right. The basin starts in balance at 42.5 % valve (Table 5.1), the makeup valve is commanded 10 % open at $t = 2.00$ min, and the stem follows 0.50 min later with the 0.33 min actuator lag (Table 5.2). The level climbs on a straight ramp of 0.333 %/min to 62.36 % with no sign of settling, while makeup at 39.4 gpm exceeds the 31.9 gpm of losses by 7.5 gpm. Callouts: (1) makeup valve LCV-201, (2) the basin level against its setpoint, (3) the water balance, makeup in against losses out, (4) the level ramp on the trend and (5) the valve command step with the delayed and lagged stem.

TECHNOLOGY **Final Element: The Makeup Valve and Its Failure Position**

A pneumatic valve with a spring goes to a known position, open or closed, when its air fails, while an electric actuator usually stays where it was. An air-to-open valve fails closed, and the risks of the process, not habit, should decide which way it fails [23]. On a basin, a valve that fails shut lets the level fall toward the pump suction, and one that fails open fills the basin to the weir. The flow at a given stem position also depends on the supply pressure upstream, so a sagging header calls for a new valve position [23]. Water guidance asks that the fill valve cut off cleanly when the basin is full [13].

Where the Controller Runs

A distributed control system runs production at one site in power generation, refining and chemical manufacturing, and a programmable logic controller with three-mode PID can serve as the main controller of a smaller system. A tower that cools a building is more often on a building automation system, which runs heating, ventilation, air conditioning and other utility systems. Loops on these

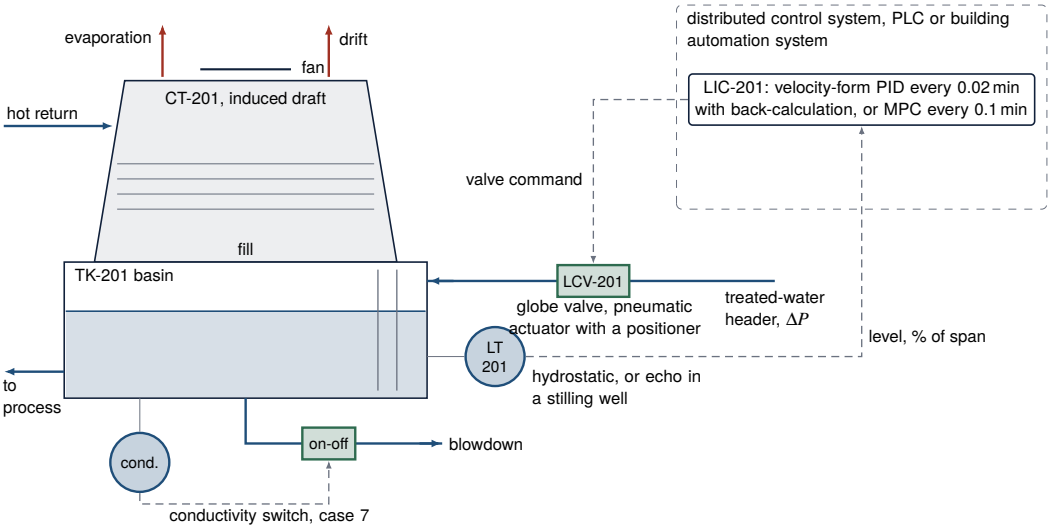


Figure 5.6 The makeup loop of the chapter with plant devices in place of the schematic's symbols. LT-201 reads the basin level, either by hydrostatic pressure or by an echo instrument in a stilling well, and LIC-201 on the control platform moves LCV-201 through its positioner. A conductivity switch opens the blowdown valve in case 7. Notice that nothing that leaves the basin is set by the level, so the makeup valve is the only device that can hold it.

platforms run with cycle times from milliseconds to minutes [24]. LIC-201 executes every 0.02 min (1.2 s) and the predictive controller decides every 0.1 min, both well inside that range. Industrial practice places model predictive control one level above the PID loops that move the valves [20], while the MPC of Section 5.5.4 moves LCV-201 directly and hands it to the PID when a plan is rejected.

> **IN PRACTICE** **Tuning a Level Loop in Service**
>
> An instrument handbook sets out the practice in a few rules. Explore the process with the controller in manual, because in automatic the response mixes the process with the controller's own action. If a step of the output makes the level ramp, the process integrates, and it may be controlled mainly by proportional action, with minimal integral action to remove offset. The lags in the loop and the measurement noise set how much gain it tolerates, since proportional action copies noise onto the output and keeps the valve stem jumping. Integral action is still needed when an outflow or the supply pressure changes [23].

What Is Modernizing It

Three developments touch this loop. The first is automation of the water side. Controllers now handle conductivity and blowdown, pH and chemical dosing, many of them with water meter inputs and alarm relays, and they hold the conductivity much closer to its set point than manual or timed blowdown [25]. The second is fieldbus, which lets the PID function block run inside the field devices, in the transmitter or in the valve positioner [23]. The third is the echo level instrument. In some industries guided-wave radar transmitters have been replacing older displacement level instruments, even at high pressure [23].

5.8 Limitations

The model is not intended for equipment selection. There is no thermal model of the tower at all, so a cold-water temperature, an approach to wet bulb or a fan power cannot be predicted, and the

Merkel and effectiveness-NTU formulations a design calculation requires [8–10] are deliberately omitted. Evaporation is one algebraic expression in the heat load with the weather frozen, which understates how much a real makeup loop moves across a day. Chemistry is one conductivity state, which puts scaling, corrosion and biological control out of reach. The valve is ideal apart from its lag and its pressure term, and stiction would add a limit cycle that looks like the one in Table 5.6 while having a different cause. Chapter 4 includes a case with valve stiction. Upsets arrive as commanded steps and timed events and never as the drifting behavior of a plant. Repeatability is itself unrealistic, because freezing the integration step and the controller period reproduces one trajectory on every machine while a distributed control system does no such thing. The four features that survived the simplification, the pure integrator, a transport delay in series with an actuator lag, the square-root valve characteristic and the on-off second loop, all survived because a controller has to be designed around each of them. Where the individual assumptions break is set out in Section 5.3.6.

Three elements of the scenarios are stylized on purpose. A basin leak that appears instantly at 12 gpm and is repaired 18 min later is not something a physical basin does, but it isolates a demand that exceeds the actuator and then stops exceeding it. A drift multiplier reaching 40× design is far beyond what a damaged eliminator produces, although it does separate a loss the blowdown calculation credits from one it does not. The heat load arrives as a step because a step is the disturbance whose response can be read off a trend, while a real load ramps.

The predictive controller plans on a fixed IPDT model from the learner's identification. It omits the actuator lag as a lag, which is the source of the strokes at $\lambda = 0$, the transmitter filter, the pressure term of Equation (5.6), which lowers the true gain to 0.0289 %/(min %) at the 75 % header of the challenge, the weir and the empty basin, and the on-off blowdown, which it sees only through $\hat{\delta}$. Nothing re-identifies the model on line, and a dead time read about 0.4 min too long makes it cycle.

5.9 Discussion

Basin level control carries a direct water cost. Recirculating evaporative cooling consumes water at a rate that dominates the water footprint of the plant it serves [12], and at the design point of this chapter the tower evaporates 25.5 gpm, bleeds 6.2 gpm and drifts 0.15 gpm, so the makeup meter runs at 31.9 gpm continuously. All of the makeup stream is softened, dosed and purchased. Overshooting the level costs twice, first because the surplus is eventually blown down or spilled over the weir along with the chemistry dissolved in it, and second because the treated blowdown stream is itself concentrated enough to be a disposal problem [14].

The cycles of concentration set the trade. Raising N from 5 to 7 reduces the required bleed in (5.4) from 6.225 to 4.10 gpm at the same duty, a saving of 6.7 % on total makeup, and moves the basin conductivity from 2000 to 2800 µS/cm, closer to the scaling limit the treatment program allows. The application exposes both N and C_M as adjustable parameters so that the trade can be computed rather than asserted, and the on-off blowdown mode shows what holding a target conductivity does to the neighboring level loop. A level controller tuned for a fast setpoint response is poorly suited to water-efficient operation. A loop slow enough that the basin absorbs the blowdown cycles and the load steps without moving the makeup valve reduces water and chemical use, which is the reasoning behind detuning level loops deliberately [2], and the reason Table 5.6 reports valve travel beside IAE rather than underneath it.

> **OPEN CHALLENGE** **Running More Cycles of Concentration**
>
> Many cooling towers run at two to four cycles of concentration, while six or more might be possible, and going from three to six cuts the makeup by 20 percent and the blowdown by half [13]. The limit is the water itself. As minerals concentrate, the potential for scale and corrosion rises, and treatment pushes those thresholds back without removing them. Reverse osmosis can recover much of the blowdown, but it is difficult to operate correctly and expensive to buy and run [25]. Progress would be a treatment and control scheme that holds high cycles without scale, so that less treated water is bought and less is sent to drain.

The model also supports control methods beyond the implemented loops. Beyond the linear MPC of Section 5.5.4, the three-state balance with a hard makeup limit is suited to a nonlinear MPC, which would carry the header pressure into the valve capacity of scenario 9 and coordinate makeup and blowdown as one two-input problem instead of two independent loops [26]. Moving horizon estimation or Kalman filtering on the same equations can estimate the unmeasured leak and drift disturbances from the level and conductivity measurements, so the capacity diagnosis of scenario 9 becomes a soft sensor for water loss. An economic formulation can trade makeup and treatment cost against the conductivity limit directly rather than through a fixed cycles target. The model equations transfer to the Gekko Optimization Suite in Python with the four hidden constants recast as estimated parameters [27].

5.10 Conclusion

This chapter studied a cooling tower basin as a laboratory for the class of plants that have no steady state of their own. The simulator integrates three states, the basin volume, the makeup valve stem and the circulating-water conductivity, with evaporation set by the cooling range, blowdown sized for a target cycles of concentration, drift proportional to circulation, and a makeup valve carrying a 0.50 min transport delay, a 0.33 min actuator lag and a square-root pressure characteristic. Four constants that fix those dynamics are hidden. A 10 % open-loop valve step identifies $\hat{K}_I = 0.0333\,\%/(\mathrm{min}\,\%)$ from a slope of 0.3333 %/min and, from the backward extension of that ramp, $\hat{\theta} = 0.83\,\mathrm{min}$, the $\theta + \tau_v$ that an integrating fit is bound to report.

SIMC settings from those two numbers reject a +30 % heat load with an IAE of 4.42 % min and 20.5 % of valve travel, while the same reaction-curve rule fed a dead time read as 0.50 min produces a gain of 56.4, 99 % of the ultimate gain the IPDT model predicts, paired with a reset time of 1.0 min, and the loop falls into a sustained 3.75 min cycle costing 475 % of valve travel for no reduction in IAE. Holding the gain fixed and cutting the reset time from 6.6 to 1.0 min multiplies the settling time by 5.7 and the valve travel by 4.7, which is the failure mode particular to a controller with an integrator in front of a plant that already has one.

A linear MPC planned on the same two numbers lowers the IAE of the setpoint and load case of scenario 10 from 18.02 to 14.06 % min with less valve travel, provided its move suppression is not zero, and it loses the unmeasured load step to the ZN PID (1.58 against 1.04 % min). A fuller model would add a wet-bulb-dependent evaporation model, valve stiction as a separate source of cycling, a scaling index alongside conductivity, and a variable-speed fan so that the thermal and water loops can be studied together. The basin remains a compact case for learning to identify and tune a process that never settles.

KEY POINTS

- A cooling tower basin loses water in ways that do not depend on its level, so the level ramps whenever makeup and losses differ.
- One valve step gives the integrating gain from a slope and an apparent dead time of 0.83 min, the transport delay plus the valve lag.
- Fed a dead time read as 0.50 min, the Ziegler-Nichols rule sets a gain of 56.4, and the loop cycles at 474.7 % of valve travel.
- With the gain fixed, cutting the reset time from 6.6 to 1.0 min multiplies the settling time by 5.7.
- A valve held wide open with zero travel marks a capacity limit, not a tuning problem.

Where else this applies. The pool of Volume 1, Chapter 9 integrates for the same reason.

References

[1] B. D. Tyreus and W. L. Luyben, “Tuning PI controllers for integrator/dead time processes,” *Industrial & Engineering Chemistry Research*, vol. 31, no. 11, pp. 2625–2628, 1992.

[2] E. C. Horton, M. W. Foley, and K. E. Kwok, “Performance assessment of level controllers,” *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 7-9, pp. 663–684, 2003.

[3] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, III, *Process Dynamics and Control*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.

[4] B. W. Bequette, *Process Control: Modeling, Design, and Simulation*. Upper Saddle River, NJ: Prentice Hall, 2003.

[5] S. Skogestad, “Simple analytic rules for model reduction and PID controller tuning,” *Journal of Process Control*, vol. 13, no. 4, pp. 291–309, 2003.

[6] A. Visioli, “Optimal tuning of PID controllers for integral and unstable processes,” *IEE Proceedings – Control Theory and Applications*, vol. 148, no. 2, pp. 180–184, 2001.

[7] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed. Princeton, NJ: Princeton University Press, 2021.

[8] D. R. Baker and H. A. Shryock, “A comprehensive approach to the analysis of cooling tower performance,” *Journal of Heat Transfer*, vol. 83, no. 3, pp. 339–349, 1961.

[9] J. C. Kloppers and D. G. Kröger, “A critical investigation into the heat and mass transfer analysis of counterflow wet-cooling towers,” *International Journal of Heat and Mass Transfer*, vol. 48, no. 3-4, pp. 765–777, 2005.

[10] N. Milosavljevic and P. Heikkilä, “A comprehensive approach to cooling tower design,” *Applied Thermal Engineering*, vol. 21, no. 9, pp. 899–915, 2001.

[11] B. A. Qureshi and S. M. Zubair, “Prediction of evaporation losses in wet cooling towers,” *Heat Transfer Engineering*, vol. 27, no. 9, pp. 86–92, 2006.

[12] J. Macknick, R. Newmark, G. Heath, and K. C. Hallett, “Operational water consumption and withdrawal factors for electricity generating technologies: a review of existing literature,” *Environmental Research Letters*, vol. 7, no. 4, p. 045802, 2012.

[13] U.S. Environmental Protection Agency, “WaterSense at work: Best management practices for commercial and institutional facilities, section 6.3: Cooling towers,” U.S. Environmental Protection Agency, WaterSense, Washington, DC, Tech. Rep., Oct. 2012.

[14] N. E. Koeman-Stein, R. J. M. Creusen, M. Zijlstra, C. K. Groot, and W. B. P. van den Broek, “Membrane distillation of industrial cooling tower blowdown water,” *Water Resources and Industry*, vol. 14, pp. 11–17, 2016.

[15] T. Liu and F. Gao, “Identification of integrating and unstable processes from relay feedback,” *Computers & Chemical Engineering*, vol. 32, no. 12, pp. 3038–3056, 2008.

[16] F. L. Lewis, “Introduction to modern control theory,” in *Applied Optimal Control and Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 1992, ch. 1, section “A Brief History of Automatic Control,” reprinted by permission at lewisgroup.uta.edu/history.htm.

[17] K. J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2006.

[18] J. G. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” *Transactions of the ASME*, vol. 64, no. 8, pp. 759–765, 1942.

[19] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[20] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[21] G. Pannocchia and J. B. Rawlings, “Disturbance models for offset-free model-predictive control,” *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[22] D. Goldfarb and A. Idnani, “A numerically stable dual method for solving strictly convex quadratic programs,” *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[23] T. R. Kuphaldt, *Lessons in Industrial Instrumentation*, version 2.33 ed. Open textbook, Creative Commons Attribution 4.0 license, ibiblio.org/kuphaldt/socratic/sinst, 2024.

[24] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, “Guide to operational technology (OT) security,” National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-82r3, 2023.

[25] Federal Energy Management Program, "Cooling towers: Understanding key components of cooling towers and how to improve water efficiency," U.S. Department of Energy, prepared by Pacific Northwest National Laboratory, Fact sheet DOE/PNNL-SA-75820, Feb. 2011.

[26] J. D. Hedengren, R. Asgharzadeh Shishavan, K. M. Powell, and T. F. Edgar, "Nonlinear modeling, estimation and predictive control in APMonitor," *Computers & Chemical Engineering*, vol. 70, pp. 133–148, 2014.

[27] L. D. R. Beal, D. C. Hill, R. A. Martin, and J. D. Hedengren, "GEKKO optimization suite," *Processes*, vol. 6, no. 8, p. 106, 2018.

5.11 Exercises

5.1 Concept check · 2 min

The basin water balance is $A_b \frac{\mathrm{d}L}{\mathrm{d}t} = q_M - q_E - q_B - q_D - q_L$. The makeup valve is left in Manual at an opening that supplies slightly less than the losses. What does the level do, and why?

(a) It falls at a constant rate until it meets a limit, because no term on the right depends on the level

(b) It settles at a lower level, because the evaporation rate falls as the basin holds less water

(c) It settles at a lower level, because the blowdown is sized in proportion to the basin inventory

(d) It falls at a rate that fades with time, because the valve passes more water as the level drops

5.2 Concept check · 2 min

After the 10 % valve step of scenario 2, the level trend visibly leaves its flat line 0.65 min after the step. The final ramp, extended backward, meets the old level 0.83 min after the step. Which value belongs in the tuning rules, and why?

(a) 0.65 min, because the first movement marks the moment the makeup starts to reach the basin

(b) 0.83 min, because the valve lag adds its full length to the delay that an integrating fit carries

(c) 0.65 min, because the ramp extension also counts the transmitter filter, which acts outside the loop

(d) 0.83 min, because the transport delay of the makeup line grows while the valve is opening

5.3 Concept check · 1 min

Which controller does the evidence of this chapter support for the noisy basin level of scenario 8?

(a) Proportional only, because the integrator in the basin removes the offset after any load change

(b) PID with a derivative time of 0.4 min, because derivative action anticipates the level ramp

(c) PI with a long reset time and no derivative, because derivative mostly turns noise into valve travel

(d) PI with a short reset time, because faster integral action removes the load offset much sooner

5.4 Theory and calculation · 5 min

The tower circulates 3000 gpm over a design cooling range of 10 °F, with an evaporation factor of 0.85, a drift fraction of 5×10^{-5} and a target of 5 cycles of concentration. The makeup valve passes 75 gpm wide open, and the basin holds 22.5 gal per percent of level.

(a) Find the evaporation, drift and blowdown flows at design load and the valve position that balances them. *Checkpoint:* 25.50, 0.15 and 6.225 gpm, balanced at 42.5 %.

(b) Find the integrating gain K_I from the valve capacity and the holdup. *Checkpoint:* 0.0333 %/min per percent of valve.

(c) Scenario 1 holds the valve at 30 %. Find the level rate, and the time for the level to fall from 60 % to the pump-suction trip at 20 %. Neglect the dead time. *Checkpoint:* −0.417 %/min and 96 min.

5.5 Theory and calculation · 7 min

Equation (5.13) gives the level after a valve step of $\Delta u = 10\,\%$ at $t = 0$, with $K_I = 0.0333\,\%/\text{min}$ per percent, $\theta = 0.50\,\text{min}$ and $\tau_v = 0.33\,\text{min}$.

(a) Show that the final ramp, extended backward, crosses the starting level at $t = \theta + \tau_v$. *Checkpoint:* 0.83 min.

(b) Find how far the level has in fact risen at that time. *Checkpoint:* 0.040 %.

(c) A trend resolves 0.01 % of level. Use the expansion $e^{-x} \simeq 1 - x + x^2/2$ for the time just after θ to estimate when the trend first shows movement. *Checkpoint:* 0.64 min after the step.

5.6 Theory and calculation · 6 min

Scenario 9 starts with the heat load at 130 %, a basin leak of 12 gpm and the makeup header at 40 % of nominal pressure. At 130 % load the evaporation, blowdown and drift total 41.44 gpm. The valve passes 75 gpm wide open at nominal pressure, and the basin holds 22.5 gal per percent.

(a) Find the largest makeup flow the valve can deliver and the total loss. *Checkpoint:* 47.4 and 53.4 gpm.

(b) Find the level rate with the valve wide open and the level at 18 min, starting from 60 %. *Checkpoint:* −0.267 %/min and 55.2 %.

(c) The leak is repaired at 18 min. Find the time at which the level is back within 2 % of the 60 % setpoint. *Checkpoint:* 28.5 min.

State what a higher controller gain would change before the repair.

5.7 App challenge · 8 min

Load scenario 2, Step test (identify the process), which starts in Manual with the makeup valve balanced at 42.5 %. Press Run, wait about one minute, press Step valve +10 %, run for at least 9 min more and export the CSV file. The scenario has no noise, so the default seed plays no part. Take the step time as the time of the last row in which `valve_cmd_pct` still reads 42.5.

(a) Read the true level, column `level_pct`, at 4 and at 8 min after the step. Find the slope of the ramp and the integrating gain. *Checkpoint:* 0.333 %/min and 0.0333 %/min per percent.

(b) Extend the line through those two rows back to 60.00 % and find the apparent dead time. *Checkpoint:* 0.83 min, within 0.02 min.

(c) Repeat part (b) with the transmitter reading, which is the column `level_measured_pct` of the same file. *Checkpoint:* 0.89 min, within 0.02 min.

(d) Under Advanced, enter 0.0333 as the Identified integrating gain and 0.83 as the Identified dead time, and set Tune from my model to SIMC PI. Read the installed tuning and the Model check tile. *Checkpoint:* $K_c = 18.1$ and $\tau_I = 6.6\,\text{min}$. The tile reads excellent, with both errors at 0 %.

5.8 App challenge · 9 min

Load scenario 5, Heat-load disturbance (+30 %). The loop is in Auto with the SIMC tuning, $K_c = 18.1$ and $\tau_I = 6.6\,\text{min}$, and the heat load steps from 100 to 130 % at 5 min. Use the default seed. In each run go a little past 20 min, pause, export the CSV file and read the row at 20.00 min.

(a) Run the scenario as shipped. Read the columns `level_iae` and `level_max_error_pct` and the valve command, `valve_cmd_pct`. *Checkpoint:* 4.42 % min, 0.70 % and 55.5 %.

(b) Set the Integral time to its top, 60 min, which switches integral action off. Press Reset and run again. Read the true level, `level_pct`, and the valve command. *Checkpoint:* 59.30 % and 55.25 %.

(c) Compute the load d in percent of valve and the offset that Equation (5.21) predicts for part (b). *Checkpoint:* 12.75 % and 0.70 %.

Accept an IAE within 0.05 % min, a level within 0.02 % and a valve within 0.1 %. State where each controller stores the extra 12.75 % of valve.

5.9 App challenge · 8 min

Load scenario 10, Three controllers from one model. The setpoint steps from 60 to 65 % at 1 min and the heat load rises by 30 % at 10 min. Use the default seed. In each run go a little past 20 min, pause, read the MV travel tile of the scorecard and export the CSV file. Read `level_iae` in the row at 20.00 min and the highest `level_pct` between 1 and 10 min.

(a) Run the scenario as shipped, with the Controller at MPC and a Move suppression of 0.3. *Checkpoint:* 14.06 % min, 65.03 % and 144 % of travel.

(b) Set the Move suppression to 0 under Advanced, press Reset and run again. *Checkpoint:* 12.95 % min, 65.31 % and about 2700 % of travel.

(c) Set the Controller to PI, press Reset and run again. *Checkpoint:* 18.02 % min, 66.31 % and 150 % of travel.

Accept an IAE within 0.05 % min, a peak within 0.02 %, and a travel within 5 % in parts (a) and (c) and within 200 % in part (b). State which design you would install and what the IAE alone hides.

5.10 Application area · 3 min

The water treatment program proposes to raise the target cycles of concentration from 5 to 7 at the design duty, where the evaporation is 25.50 gpm, the design drift is 0.15 gpm and the makeup conductivity is 400 µS/cm.

(a) Find the blowdown at each target and the saving in total makeup, in gpm, in percent and in gallons per day. *Checkpoint:* 6.225 and 4.10 gpm. The saving is 2.125 gpm, 6.7 % or 3060 gallons per day.

(b) Find the basin conductivity at each target and name what limits the choice. *Checkpoint:* 2000 and 2800 µS/cm.

5.11 Application area · 4 min

A plant historian shows a basin level that stayed within 1 % of its setpoint for a week. Over the same week the makeup valve swung across most of its range about every four minutes.

(a) Name the scorecard measure of this chapter that exposes the problem, and explain why the level error does not. *Checkpoint:* The valve travel.

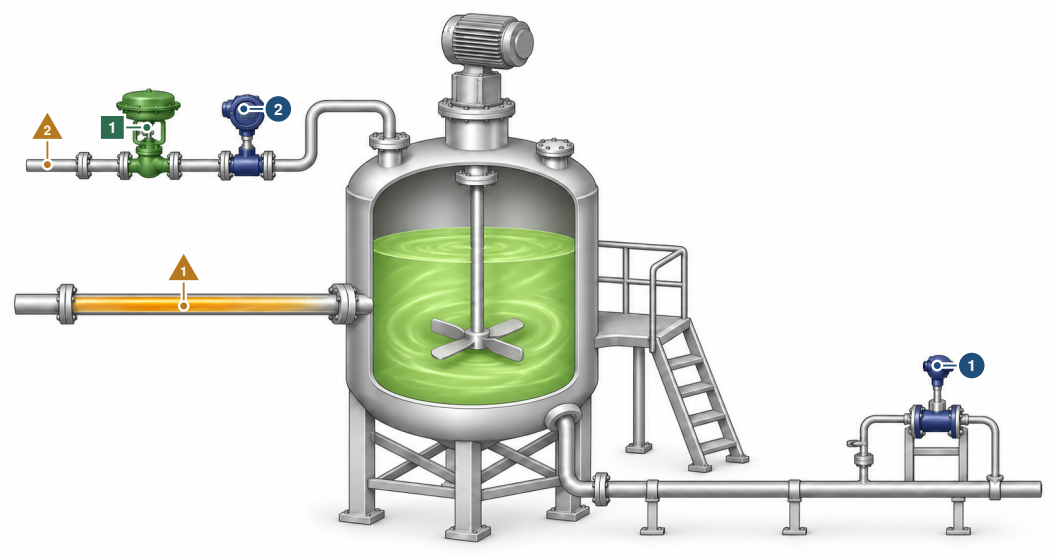
(b) The chapter gives two causes of such a cycle. Name them, and give one test on the plant that separates them. *Checkpoint:* A dead time read too short, or valve stiction.

5.12 Review the agent · 5 min

An AI assistant reviews the scenario 5 run under the SIMC PI and replies as follows. “The level is still 0.05 % low fifteen minutes after the heat-load step, because the integral time of 6.6 min is long. Cut the integral time to 1.0 min and keep $K_c = 18.1$. The proportional gain sets the stability margin against the dead time, so with the gain unchanged the margin is unchanged, and the level will return about six times faster.” Find the flaw in the proposal and say how it could be checked.

CHAPTER 6

pH Neutralization Virtual Laboratory for Gain Scheduling and Measurement Linearization



The controller moves 1 the caustic valve

The controller reads 1 the pH electrode on the discharge 2 the caustic flow meter of the inner loop

What disturbs it 1 acid and buffering in the influent 2 caustic supply pressure and strength

Notice that the caustic is dosed into the top of the tank while the pH that decides the permit is read on the discharge line downstream, after mixing and transport. Illustration, not to scale.

IN PLAIN TERMS **Neutralizing Acid Waste Before It Leaves**

The main idea. Acid waste water from a factory flows through a stirred tank. A valve adds caustic soda, a strong base, to bring the water close to neutral. The water may leave only inside a legal band of acidity, read by a probe in the outlet.

The challenge. Near neutral, a tiny extra opening of the valve swings the water from too acidic to too alkaline. Far from neutral, a wide opening barely moves it. A controller set for one place is too jumpy or too sleepy at the other.

Why it matters. Water outside the band breaks the plant's permit, so the controller must change how hard it pushes as the reading moves.

THREE WAYS TO READ THIS CHAPTER

Explore	11 min	the first page, the fact on the permit and the history of gain scheduling (Section 6.1), the try-it segments of cases 1 and 4 (Section 6.6), the fact on caustic and cost, and the key points
Study	50 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	15 min	the first page, the results (Section 6.6), Technology in Practice (Section 6.7), Limitations (Section 6.8) and the open challenge

The control question A discharge permit of pH 6 to 9 occupies 0.155 % of the reagent valve's travel, and the gain from valve to pH changes by four orders of magnitude across one stroke. When does such a loop need a different structure instead of a better tuning, and which structure repairs which defect?

Learning objectives

- Derive the reagent demand and the process gain at the setpoint from the invariant balances and the charge balance.
- Explain why two correct step tests on this loop return gains that differ by a factor of 691.
- Compare a fixed gain, a gain schedule and control on the inverted titration curve after a load change and after a buffering change.
- State which disturbances an inner reagent flow loop rejects and which it cannot see.
- Judge when valve size, reagent strength or tank volume is the remedy that no control structure supplies.

Prerequisites Sections 1.4.4 and 1.4.5 (step test and tuning), 1.4.13 (scheduling a gain), 1.4.9 (windup) and 1.6.2 (the PID algorithm of the apps). Chapter 4 for tuning from an identified model.

The app https://apmonitor.com/apps/ph-neutralization/

See also Volume 7, Chapter 3, Volume 7, Chapter 2 and Volume 7, Chapter 4 for plants whose gain moves or reverses with the operating point. Volume 3, Chapter 2 for matching a structure to the path of a disturbance and Volume 3, Chapter 6 for a dose computed from an upstream analyzer. Volume 13, Chapter 5 for ratio control with analyzer trim on the plant that makes the caustic.

6.1 Introduction

The discharge permit on the neutralization tank of this chapter is pH 6 to 9, which is three whole units and sounds generous. The reagent valve holds pH 7.00 at 19.899 % of its travel, reaches the bottom of the permit at 19.791 % and passes the top of it at 19.947 %. The entire legal operating range is 0.155 % of one valve, which is finer than the resolution of a typical positioner and far finer than an operator can hold by hand. That single arithmetic fact is what a block diagram of a pH loop cannot deliver, and it is the reason the loop has the industrial reputation it has [1, 2].

> **DID YOU KNOW? The Same Band in a Federal Rule**
>
> The permit of this chapter, pH 6 to 9, is the band that the United States secondary treatment rule sets for publicly owned treatment works. Their effluent pH must stay within 6.0 to 9.0, unless the works adds no inorganic chemicals in treatment and its industrial contributors do not push the pH outside that band [3].

The misconception the simulator corrects is that pH control is hard because the chemistry is hard. It is not. Written in reaction-invariant form, the tank carries two states that blend exactly the way any other conserved quantity blends in a stirred vessel, with one time constant and no coupling [4]. The pH is then recovered from those two states by an algebraic charge balance. The plant is linear and the measurement is not, and a later study by the originators of that form sets out how much of the folklore around pH loops follows from confusing the two [5]. A student who watches the invariant trend travel a clean first-order path while the pH trace falls off a cliff has seen the distinction, and a transfer function from valve position to pH hides it by construction, because the transfer function is only valid at the point where it was linearized.

The general problem is control of a plant whose gain is a strong function of where the plant is sitting rather than a constant. Level in a horizontal drum, temperature through a phase change, composition near a distillation pinch, and flow through a butterfly valve at low lift all have the same shape, and the control literature treats the class rather than the instance. There are two standard answers. The gain can be scheduled on a measured variable so that the product of controller gain and process gain stays fixed, as Section 1.4.13 introduces. The theory and limits of the technique are surveyed for the general case [6]. Alternatively the nonlinearity can be undone by a transformation,

so that the controller works on a variable in which the plant is linear, which is the input-output linearization construction [7] and, applied to this plant, the strong acid equivalent [8]. pH is the textbook case for both because its transformation is the titration curve, which can be computed from equilibrium constants rather than fitted. Chapter 2 is the mild case of the same class, a process gain that falls by a factor of three across the speed range of a car.

> ⧖ HISTORY **Scheduled Gains in Flight Control, 1959**
>
> Aircraft gave gain scheduling its first large field. Wide variations of operating conditions in flight stimulated the development of gain scheduling and of adaptive control, and a high-performance aircraft schedules its controller on Mach number and dynamic pressure, both of them measured. Adaptive control grew from the wish to avoid scheduling in flight control and from the hope of a process controller that adjusts itself, and early work on it was gathered at a symposium on self-adaptive flight control in 1959. Scheduling still dominates flight control, in part because air-data sensors improved so much [9]. In this chapter the measured pH plays the part of Mach number, and the slope of an assumed titration curve sets the gain.

Simulation studies of neutralization are usually built to test an advanced controller, and the model is presented as the setting for that controller [10, 11]. However, a student handed the finished nonlinear controller never performs the step test that returns the wrong number, never sees a correctly tuned loop go to sleep, and never has to decide which of three structural remedies the plant in front of them actually needs. This chapter works with an interactive pH neutralization tank in which the process gain, the reagent demand and the permit band are all measurable from the running plant, so that the case for scheduling, for inversion and for cascade is quantified rather than asserted. The invariant formulation, the titration-curve inverse and the tuning arithmetic are all established methods.

The chapter is organized as follows. Section 6.2 gives the history of the control problem. Section 6.3 states the five-state reaction-invariant model with its instrument chain and its actuator model, in the detail needed to rebuild it. Section 6.4 computes the process gain across one valve stroke and shows why two correct step tests return incompatible answers. Section 6.5 describes the four control architectures, a fixed-gain PI loop, a gain schedule, a loop on the inverted titration curve and a cascade with an inner flow loop. Section 6.6 reports a scenario sequence in which each architecture repairs exactly one defect and the last disturbance defeats all of them. Section 6.7 sets the real instruments and control platform beside the model, and Sections 6.8 and 6.9 state the limitations of the model and weigh the three structural remedies against the equipment decisions. Section 6.10 summarizes the results, and the exercises follow in Section 6.11.

6.2 A History of pH Control

Neutralization became a control problem rather than a batch chemistry problem when effluent limits were written into discharge permits and the treatment had to run continuously alongside production. The first monograph devoted to the subject collected what plants had learned by then, that the reagent valve is almost always oversized for its duty, that a single stirred tank rarely holds a tight permit and two or three in series are the ordinary remedy, that mixing supplies most of the apparent dead time, and that equal-percentage or split-range trim is the cheapest available linearization [1]. Those conclusions were reached from plant experience before there was a dynamic model to support them.

The dynamic model arrived in the early 1970s. A continuous stirred tank receiving a strong acid and a weak acid against a strong base was written down, solved and compared with laboratory data, and it established the shape of the problem, an ordinary mixing lag followed by an equilibrium relation that converts a small change in composition into a large change in pH [12]. That vessel became the benchmark the field re-used, and it was extended to a multivariable version with acid, base and buffer streams that is still the standard test plant [13]. The chemistry in the present application is that benchmark, acetic acid at pK_a 4.76 with a strong mineral acid against sodium hydroxide, scaled from the laboratory rig to a plant-sized vessel.

The decisive modeling step was made in 1983. Rather than integrating concentrations of the ionic species, whose equilibria are microseconds fast against a five-minute tank, two reaction invariants are carried, and neither is changed by any acid-base reaction [4]. The states become linear, the stiffness disappears, and the nonlinearity is isolated in an algebraic output map. The same construction supplies the controller with an exact inverse, and it was later shown that the resulting nonlinear and adaptive schemes are the ones that survive a change in the influent composition [14]. An equivalent transformation, the strong acid equivalent, was derived independently from input-output linearization and demonstrated on an industrial stream [7, 8].

What engineers did before that answer became standard was adapt. Because the titration curve is what the controller needs and nobody measures buffering on line, the curve was estimated on line from the reagent demand the loop turned out to require, and the controller was retuned or re-characterized from the estimate [15]. Adaptive nonlinear schemes that estimate the unmeasured invariants from the pH measurement itself followed and were tested against changes in buffering [10]. Later work identified the curve as the static block of a Wiener model and inverted it inside a feedforward-feedback structure [11]. All of these share one structure, an inverse of an assumed curve placed between the measurement and a linear controller, and they differ in where the assumed curve comes from.

The industrial reputation of the loop did not improve as much as the theory did. Practitioner references written thirty years apart give the same list of causes, an oversized reagent valve, an unmeasured and moving buffering, badly mixed vessels, and a glass electrode that drifts and whose lag grows by an order of magnitude as it fouls [1, 2]. The measurement problem is specific to pH, because the controller is holding a logarithm and a small analog error at the electrode becomes a large error in the quantity the permit is written on. The application of this chapter carries all four of those causes as adjustable features rather than as remarks.

6.3 Process Model

Neutralization tank TK-401 is a 400 L agitated vessel treating 80 L/min of plant effluent that carries 5.0 mmol/L of a strong mineral acid and 0.6 mmol/L of acetic acid at pH 2.301. Reagent is 0.50 mol/L sodium hydroxide through control valve FCV-401, which passes 4.5 L/min at full travel. A small acid trim pump is available for the opposite direction. Electrode AIT-401 measures the discharge, and the permit is pH 6 to 9. Figure 6.1 shows the arrangement, the instrument tags and every symbol that appears in the equations. The residence time at the design point is 4.94 min, and the reagent demand that holds pH 7.00 is 0.8955 L/min, which is 19.899 % of the valve.

Table 6.1 lists what the loop interacts with. The clock is in minutes, concentrations in mol/L, flows in L/min, and valve travel and reagent demand in percent. Five states are integrated and not one of them is the concentration of a reacting species.

6.3.1 Two Reaction Invariants

Writing differential equations for $[H^+]$, $[OH^-]$, [HAc] and $[Ac^-]$ is unnecessary because the equilibria among them complete in microseconds against a 4.94 min tank. The model instead carries the two quantities of Equation (6.1), neither of which any acid-base reaction changes [4].

$$W_a = [H^+] - [OH^-] - [Ac^-], \qquad W_b = [HAc] + [Ac^-] \tag{6.1}$$

where W_a is the excess of strong anion over strong cation in mol/L and W_b is the total weak acid however it is dissociated. Because no reaction term is needed, each invariant blends in the vessel exactly as an inert tracer does, and the balances are Equations (6.2) and (6.3).

$$V \frac{\mathrm{d}W_a}{\mathrm{d}t} = Q_1 c_a - Q_2 C_b + Q_3 C_{\mathrm{HCl}} - Q_t W_a \tag{6.2}$$

$$V \frac{\mathrm{d}W_b}{\mathrm{d}t} = Q_1 c_b - Q_t W_b \tag{6.3}$$

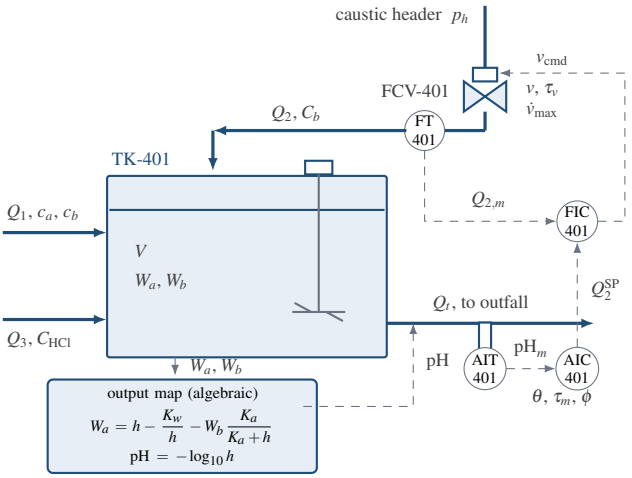


Figure 6.1 Neutralization tank TK-401 with its reagent, effluent and trim streams. Influent Q_1 carries strong acid c_a and weak acid c_b into working volume V, which holds the two reaction invariants W_a and W_b and discharges Q_t. Caustic of concentration C_b enters as Q_2 through valve FCV-401, whose stem v follows the command v_{cmd} with lag τ_v and rate limit $\dot{v}_{\max}$ against header pressure p_h. The pH leaving the tank is not a state and is recovered from W_a and W_b by the charge balance shown. AIT-401 reads that pH through transport delay θ, electrode lag τ_m and fouling fraction ϕ, AIC-401 sets a reagent flow demand, and FIC-401 closes the inner loop on the transmitter FT-401.

Table 6.1 Classification of Variables. Nominal Values Are Those of the Design Operating Point at Which Every Closed-Loop Scenario Starts.

Symbol	Role	Nominal	Range
pH (AIT-401)	CV, measured	7.00	4–11 SP
u demand	MV, outer loop	19.899 %	0–100 %
Q_2^{SP}	MV, cascade only	0.8955 L/min	0–4.5 L/min
v travel	MV, inner loop	19.899 %	0–100 %
c_b weak acid	DV, unmeasured	0.6 mmol/L	0–20 mmol/L
c_a strong acid	DV, unmeasured	5.0 mmol/L	1–15 mmol/L
Q_1 influent	DV, measured	80 L/min	20–170 L/min
p_h header	DV, unmeasured	1.0×	0.4–1.6×
reagent strength	DV, unmeasured	1.0×	0.6–1.4×
Q_3 acid trim	DV, measured	0 L/min	0–0.4 L/min
fouling rate	DV, unmeasured	0 %/h	−30–40 %/h
W_a	state	–0.590 mmol/L	±1000 mmol/L
W_b	state	0.593 mmol/L	0–500 mmol/L
v	state	19.899 %	0–100 %
Q_1^f pump	state	80 L/min	≥ 0.1 L/min
ϕ fouling	state	0	0–1

where V is the working volume in L, Q_1, Q_2 and Q_3 are the influent, reagent and acid trim flows in L/min, $Q_t = Q_1 + Q_2 + Q_3$ is the discharge, c_a and c_b are the influent strong and weak acid concentrations, C_b is the caustic concentration and C_{HCl} is the trim acid concentration. Both sides carry units of mol/min. The caustic appears in (6.2) with a negative sign because a strong base raises the strong cation, and it does not appear in (6.3) at all because it adds no weak acid. Equations (6.2) and (6.3) are two uncoupled first-order lags of time constant V/Q_t, and there is no nonlinearity in either of them.

6.3.2 The Output Map

The pH is recovered from the two invariants by the charge balance of Equation (6.4), solved for the hydrogen ion concentration h.

$$W_a = h - \frac{K_w}{h} - W_b \frac{K_a}{K_a + h}, \qquad \mathrm{pH} = -\log_{10} h \tag{6.4}$$

where $K_a = 10^{-\mathrm{p}K_a}$ and $K_w = 10^{-\mathrm{p}K_w}$. The right side of (6.4) is strictly increasing in h, because its derivative is Equation (6.5) and every term is positive for physical W_b.

$$J = \frac{\partial W_a}{\partial h} = 1 + \frac{K_w}{h^2} + \frac{W_b K_a}{(K_a + h)^2} > 0 \tag{6.5}$$

A bisection on $\log_{10} h$ over $[10^{-15.5}, 10]$ therefore cannot fail. The implementation takes 48 halvings, which places the answer within 6×10^{-14} of a decade, and a check in the application's test suite confirms that inverting (6.4) and evaluating it again reproduces a specified pH to better than 10^{-9} over $1.5 \le \mathrm{pH} \le 12.5$ at weak acid levels from 0 to 20 mmol/L. This one algebraic line is where the entire difficulty of the application lives.

Differentiating (6.4) gives the sensitivity of the measurement to each state, as shown in Equation (6.6).

$$\frac{\partial \mathrm{pH}}{\partial W_a} = \frac{-1}{h \ln 10 \, J}, \qquad \frac{\partial \mathrm{pH}}{\partial W_b} = \frac{-1}{h \ln 10 \, J} \cdot \frac{K_a}{K_a + h} \tag{6.6}$$

The leading h in the denominator moves by ten decades across the pH scale while J moves by two or three, so the sensitivity is dominated by it. The bracket J is the buffer capacity, and it is the term the influent weak acid controls.

6.3.3 Reagent Valve, Influent Pump and Instruments

The reagent valve is a first-order actuator with a rate limit, and its flow follows the ordinary valve law, as shown in Equations (6.7) and (6.8).

$$\tau_v \frac{\mathrm{d}v}{\mathrm{d}t} = v_a - v, \qquad \left| \frac{\mathrm{d}v_a}{\mathrm{d}t} \right| \le \dot{v}_{\max} \tag{6.7}$$

$$Q_2 = Q_{r,\max} \, f(v) \, \sqrt{p_h}, \qquad f(v) = \frac{v}{100} \tag{6.8}$$

where v is the stem position in percent, v_a is the commanded position v_{cmd} after rate limiting, τ_v is the travel time constant, $\dot{v}_{\max}$ is the slew limit in %/min, $Q_{r,\max}$ is the capacity at full travel and p_h is the header pressure as a fraction of nominal. The linear trim of (6.8) is the default and is what is usually installed. Setting the rangeability R_v above unity replaces f with the equal-percentage characteristic $(R_v^{v/100} - 1)/(R_v - 1)$. The influent pump ramps between throughputs through a first-order lag of time constant τ_Q, and the electrode fouling fraction ϕ grows at a rate the operator sets, as shown in Equation (6.9).

$$\tau_Q \frac{\mathrm{d}Q_1}{\mathrm{d}t} = Q_1^{\mathrm{set}} - Q_1, \qquad \frac{\mathrm{d}\phi}{\mathrm{d}t} = \frac{r_f}{6000} \tag{6.9}$$

where r_f is the fouling rate in percent per hour and the constant 6000 converts to a fraction per minute. The state ϕ is clipped to the interval $[0, 1]$ after every step, v to $[0, 100]$, W_b to non-negative values and Q_1 to at least 0.1 L/min.

The electrode is the only pH instrument the controller reads, and it carries a transport delay, a fouling offset, an electrode lag that grows with fouling and colored noise, as shown in Equation (6.10).

$$\tau_m \, (1 + \kappa \phi) \, \frac{\mathrm{d}\mathrm{pH}_m}{\mathrm{d}t} = \mathrm{pH}(t - \theta) + \delta \phi + n(t) - \mathrm{pH}_m \tag{6.10}$$

Table 6.2 Model Parameters. Every Constant in This Application Is Displayed and Adjustable, so Nothing Is Hidden From the Student.

Symbol	Meaning	Value	Unit
V	working volume	400	L
τ_Q	influent pump ramp	0.25	min
$\mathrm{p}K_a$	acetic acid dissociation	4.76	—
$\mathrm{p}K_w$	ion product of water at 25 °C	14.00	—
C_b	caustic concentration	0.50	mol/L
C_{HCl}	trim acid concentration	0.50	mol/L
$Q_{r,\max}$	reagent valve capacity	4.5	L/min
R_v	valve rangeability	1.0	—
τ_v	valve travel time constant	0.08	min
$\dot{v}_{\max}$	valve slew limit	400	%/min
τ_m	clean electrode time constant	0.20	min
θ	transport delay to electrode	0.15	min
κ	electrode lag multiplier, fouled	9	—
δ	electrode offset, fouled	−0.9	pH
τ_F	reagent flow transmitter lag	0.05	min
—	permit lower limit	6.0	pH
—	permit upper limit	9.0	pH
—	caustic price, 100 % basis	0.60	$/kg
—	cost of discharge off permit	12	$/min

where θ is the transport delay from the injection point to the electrode, τ_m is the clean electrode time constant, κ is the lag multiplier at full fouling, δ is the reading offset at full fouling and n is a first-order-filtered Gaussian sequence scaled so that its stationary standard deviation equals the specified value. The offset is added before the electrode lag rather than after it, because the drift is a property of the glass. Equation (6.10) is advanced by the exact zero-order-hold update $\mathrm{pH}_m^k = \alpha\,\mathrm{pH}_m^{k-1} + (1-\alpha)\,x_k$ with $\alpha = \exp(-\Delta t/\tau)$, which is stable at any sample period even while τ is changing. The reagent flow transmitter is a clean first-order lag of time constant τ_F on Q_2 with no delay and no noise, and it is what makes an inner flow loop worth having.

6.3.4 Initialization, Numerics and Scorecard

The constants are listed in Table 6.2. Every scenario opens at a genuine steady state. The valve is placed at the travel that holds the pH setpoint, found by bisection on (6.4) through the steady form of (6.2) and (6.3), or at the operator's manual position when the scenario starts in manual, and the two invariants are set to the values that position produces. The test suite confirms that the resulting derivative vector is zero to within a relative 10^{-8} on every state.

The engine of Section 1.6.1 integrates the model with its fourth-order Runge-Kutta method on a fixed step of 0.01 min (0.6 s). The controller executes every 0.02 min (1.2 s) and the historian logs 20 samples per simulated minute, one every 3 s. The default seed of the noise generator is 20260118. The scorecard accumulates the caustic consumed as kilograms of 100 % sodium hydroxide, the minutes spent outside the permit on the true pH, the integrated absolute error against the setpoint, the total variation of the valve stem, and a cost that adds caustic purchased to a flat charge for every minute off permit. The exported file holds the true pH beside the electrode reading, both invariants, the local process gain, the controller gain in force and the scheduling multiplier at every logged sample, so the gain map behind Figure 6.2 can be rebuilt from the record of a run.

6.3.5 Assumptions

The model is detailed on the chemistry that sets the gain and thin everywhere else, and each simplification fails somewhere.

The acid-base equilibria are instantaneous. Proton transfer completes in microseconds against a 4.94 min residence time, so treating the ionic species as algebraic functions of two invariants is more accurate than writing differential equations for them, not less [4]. The assumption fails the moment the effluent carries something that reacts slowly, such as metal hydroxides coming out of solution, carbonate solids, or lime slurry in place of caustic, because those are minutes-scale processes that buffer the tank and no algebraic charge balance represents them.

One monoprotic weak acid. Real effluent is a mixture with several pK_a values spread across the range, so its curve has several inflections rather than one cliff. Dissolved carbon dioxide is the important omission, because a tank open to atmosphere buffers itself near pH 6 and again near pH 10 on a timescale of minutes, which is the timescale of the control problem.

Fixed temperature, and with it fixed K_a and K_w. At 25 °C neutrality is pH 7.00 and at 40 °C it is near pH 6.8, so a tank that swings fifteen degrees over a day has a titration curve that moves under the controller. The temperature compensation fitted to a real transmitter corrects the Nernst slope of the glass and not the equilibrium constants of the liquid, so this error survives a properly compensated instrument.

Concentrations are used where activities belong. At the 5 mmol/L of this design point the activity correction is worth a few hundredths of a pH unit. In a brine effluent it is worth several tenths and the equilibrium constants shift with it.

One perfectly mixed volume plus a lumped delay. Real neutralization tanks are badly mixed by this standard, because reagent enters at the surface into a stream orders of magnitude more acidic and the local pH there is nothing like the bulk. A defensible model of a real vessel is two or three compartments in series, so the single delay of 0.15 min understates how much of a pH loop's dead time is mixing rather than piping [1].

The valve is ideal apart from its lag, its rate limit and its pressure term. There is no stiction, no deadband and no hysteresis. That is the most consequential omission, because a sticky reagent valve is one of the commonest causes of a real pH limit cycle and it is routinely misdiagnosed as a tuning problem [2]. Chapter 4 includes a case with valve stiction.

Electrode fouling is linear in one state. Real fouling is nonlinear, coats asymmetrically, and often fails toward a fixed reading rather than a fixed offset. The failure the application cannot show is the worst one, in which the trace goes quiet and looks like perfect control.

6.4 The Gain That Is Not a Number

6.4.1 Process Gain Across One Valve Stroke

Combining (6.6) with the steady form of (6.2) and (6.3) gives the gain from valve travel to pH, as shown in Equation (6.11).

$$\frac{\partial \mathrm{pH}}{\partial v} = \frac{1}{Q_t}\frac{\mathrm{d}Q_2}{\mathrm{d}v}\left[\frac{\partial \mathrm{pH}}{\partial W_a}(-C_b - W_a) + \frac{\partial \mathrm{pH}}{\partial W_b}(-W_b)\right] \tag{6.11}$$

where $\mathrm{d}Q_2/\mathrm{d}v = Q_{r,\max}\sqrt{p_h}/100$ for linear trim. Sweeping the valve from shut to wide open at the design composition and evaluating (6.11) at each steady state gives a gain of 5.191×10^{-3} pH per percent of travel with the valve wide open at pH 12.33 and 101.98 pH per percent at pH 7.77. The ratio is 1.96×10^{4}, which is 4.29 orders of magnitude on one valve, one tank and one reagent. The tangent gain at the setpoint itself is 33.78 pH per percent. Figure 6.2 draws the curve and both operating points.

6.4.2 Two Correct Step Tests With Incompatible Answers

The identification test is the step test of Section 1.4.4, and it is performed twice. A step of 1 % on the valve from 12 %, out on the flat part of the curve, moves the tank from pH 2.790 to pH 2.872 by $t = 40$ min. The gain is 0.082 pH per percent of valve, and 63 percent of the change appears in the tank pH 5.32 min after the step. The invariant W_a, which is linear in the mixing, reaches its 63 percent

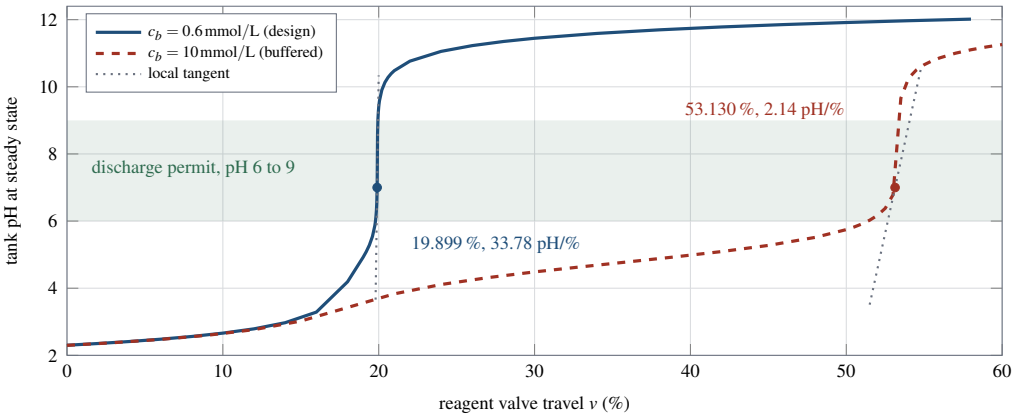


Figure 6.2 Steady-state titration curves of the neutralization tank against reagent valve travel. The design influent (solid curve) reaches pH 7.00 at 19.899 % of travel with a local slope of 33.78 pH per percent, and the whole permit band spans 0.155 % of the same valve. Raising the influent weak acid from 0.6 to 10 mmol/L (dashed curve) moves the same setpoint to 53.130 % and flattens the slope to 2.14 pH per percent, while the influent pH itself moves only from 2.301 to 2.298. The dotted lines are the local tangents at the two operating points, each marked by a filled circle, and the shaded band is the discharge permit.

point at 5.02 min, the 4.96 min residence time at this valve position plus the 0.08 min valve lag, and the curvature of the titration curve, slight on this part, adds the remaining 0.30 min. The electrode sees the response through the 0.15 min transport delay and the 0.20 min lag, and its noise-free reading reaches 63 percent of the change 5.68 min after the step. That is an ordinary first-order-plus-dead-time response with $\tau_p = 4.94$ min (the design residence time) and $\theta_p = 0.35$ min (the transport delay and electrode lag in series). The IMC-PI rule [16] of Equation (6.12), which Section 1.4.5 derives, with a 1.5 min closed-loop target returns $K_c = 33$ percent of valve per pH unit and a reset time equal to τ_p, which the application loads as 5.2 min.

$$K_c = \frac{\tau_p}{K_p\,(\tau_c + \theta_p)}, \qquad \tau_I = \tau_p \tag{6.12}$$

The same test at the setpoint uses a step of 0.03 %, which is thirty-three times smaller, and moves the tank from pH 7.003 to pH 8.701. The secant gain between the two steady states is 56.66 pH per percent, a factor of 691 against the 0.082 the identical procedure measured on the flat part, and a step a thirty-third of the size produced a response twenty times larger. Applying (6.12) to the setpoint tangent of 33.78 with the same 1.5 min closed-loop target returns $K_c = 0.079$, and the application loads 0.092, which corresponds to a 1.24 min target. Both tests were performed correctly. A step test on a pH loop measures the gain at the point where the test was done, and the invariant trend is the honest picture, because W_a traces the same clean first-order response in both tests and moves by the same 0.28 mmol/L per percent of valve (0.279 mmol/L for the 1 % step and 0.0083 mmol/L for the 0.03 % step).

Pause. Try Exercise 6.1 before reading on.

6.4.3 Buffering Moves the Curve Without Moving the Inlet

Acetic acid at pK_a 4.76 is barely dissociated at pH 2.3 and almost completely dissociated at pH 7, so it contributes very little to the influent pH and a great deal to the reagent demand. Raising c_b from 0.6 to 10 mmol/L moves the influent pH from 2.301 to 2.298, which no instrument would report as a change. At the setpoint it divides the tangent gain by 15.8, from 33.78 to 2.14 pH per percent

through the J term of (6.5), raises the reagent demand from 19.899 to 53.130 % of valve, and widens the permit band from 0.155 to 1.973 % of travel. Both curves appear in Figure 6.2. Nobody puts a titrator on an effluent line, so the control system discovers this at the same moment the permit does, which is why the curve has to be estimated from the demand the loop turns out to require [15].

Pause. Try Exercise 6.2 before reading on.

6.5 Controller Design

6.5.1 The Shared PID Algorithm

Both loops use the velocity-form algorithm of the framework, which Section 1.6.2 gives in general. Equations (6.13) and (6.14) restate it in the symbols of this chapter, because the windup analysis and the results cite them.

$$\begin{aligned} e_k &= a\,(\mathrm{SP}_k - \mathrm{PV}_k), \qquad P_k = K_c e_k \\ D_k &= -K_c \tau_D\, a \frac{\mathrm{d}\mathrm{PV}_k^{f}}{\mathrm{d}t} \\ s_k &= s_{k-1} + (P_k - P_{k-1}) + (D_k - D_{k-1}) \end{aligned} \tag{6.13}$$

$$\begin{aligned} c_k &= u_{\mathrm{bias}} + s_k + I_{k-1} \\ u_k &= \mathrm{slew}(\mathrm{clamp}(c_k, u_{\min}, u_{\max}),\, u_{k-1},\, \dot{u}_{\max},\, \Delta t_c) \\ I_k &= I_{k-1} + \left[\frac{K_c}{\tau_I} e_k + \frac{u_k - c_k}{T_t}\right] \Delta t_c \end{aligned} \tag{6.14}$$

The accumulator s_k is the running sum of Equation (1.24) and the demand c_k is the output before limiting of Equation (1.25). The slew operator holds $|u_k - u_{k-1}|$ to at most $\dot{u}_{\max}\Delta t_c$, where Δt_c is the control period. Here $a = +1$ for both loops, which is reverse action (Section 1.4.3), because the process gain from reagent to pH is positive and a rising measurement must be met with a falling output. The accumulator equals the positional $P + D$ for the gains in force, so the demand keeps the identity $c_k = u_{\mathrm{bias}} + P_k + D_k + I_{k-1}$ and a gain change acts on the whole error at the next execution. That property matters on this plant, because the scheduled architecture changes the gain at every execution. Anti-windup is back-calculation with tracking time constant T_t, which defaults to τ_I and is set to 10^9 minutes when the student switches anti-windup off [17]. Sections 1.4.9 and 1.4.10 give the reasoning behind the limit, the correction of the integral and the bumpless transfer between Manual and Auto.

The outer loop is bounded to a reagent demand of 0 to 100 % and carries a derivative filter of 0.12 min. The inner flow loop is bounded to a valve travel of 0 to 100 % and has no derivative term. Tuning constants are bounded to $0 \le K_c \le 40$, $0.2\,\mathrm{min} \le \tau_I \le 120\,\mathrm{min}$ and $0 \le \tau_D \le 10\,\mathrm{min}$ on the outer loop. Both execute every 0.02 min, independent of the 0.01 min physics step. The default derivative time is zero, because derivative action on a pH electrode multiplies its noise by whatever the local slope happens to be and that slope moves by four orders of magnitude.

6.5.2 Four Architectures

The architecture switch selects what the controller measures and what its output means, and each step repairs exactly one defect. In the PI architecture the controller acts on the measured pH directly and its output is the valve position, so K_c has units of percent of valve per pH unit. In the GS architecture the same loop multiplies K_c by the ratio of titration-curve slopes between the measurement and the setpoint, as shown in Equation (6.15).

$$K_c^{\mathrm{eff}} = K_c\,\mathrm{clamp}\left(\frac{u^{*\prime}(\mathrm{pH}_m)}{u^{*\prime}(\mathrm{pH}_{\mathrm{SP}})},\ 0.02,\ 3000\right) \tag{6.15}$$

where $u^*(\mathrm{pH})$ is the reagent demand an assumed influent would need to sit at that pH and the slope is taken as a central difference over ± 0.04 pH. The clamp is necessary because the unclamped ratio reaches 873 at pH 2.5 and 6510 with the valve wide open at pH 12.33. Equation (6.15) is the schedule of Equation (1.22) with the slope of an assumed titration curve in place of a measured gain curve. In the LIN architecture the measured pH and the setpoint are both pushed through u^* before the controller sees them, so the controller works on a reagent demand in percent of valve. The inverse is Equation (6.16).

$$u^*(\mathrm{pH}) = \frac{100}{Q_{r,\max}} \cdot \frac{Q_1 \hat{c}_a + Q_3 C_{\mathrm{HCl}} - \hat{W}_a(\mathrm{pH})\,(Q_1 + Q_3)}{C_b + \hat{W}_a(\mathrm{pH})} \tag{6.16}$$

where $\hat{W}_a(\mathrm{pH})$ is (6.4) evaluated at $h = 10^{-\mathrm{pH}}$ with the assumed weak acid, and the dilution of the weak acid by the reagent stream is resolved by eight fixed-point iterations. At steady state the demand that holds the measured pH is the valve position, so the process gain in this variable is 1 by construction, and (6.12) on a unit-gain 4.94 min lag with 0.35 min of dead time and the same 1.5 min target returns $K_c = 2.7$, dimensionless, which the scenarios round to 3. In the CASCADE architecture the same linearized outer loop writes a reagent flow setpoint $Q_2^{\mathrm{SP}} = Q_{r,\max} u/100$ and FIC-401 drives the valve against the transmitter reading.

Three properties of (6.15) and (6.16) are worth stating plainly. Both are built from an assumed influent composition $(\hat{c}_a, \hat{c}_b)$, which the student sets on the advanced panel, because nobody measures buffering on line. The C_b in the denominator of (6.16) is the nameplate caustic concentration and not the strength of the delivery that is actually in the tank, because the strength of a tanker load is not measured either, so a weaker delivery is unmeasured by the whole control system and not only by the inner flow loop. Neither contains the header pressure term of (6.8), so the demand is a percent of valve at nominal header pressure and a pressure change appears to the outer loop as an unmeasured disturbance. The consequence is that these structures are detuned rather than destabilized when the assumption is wrong, which is measured in Section 6.6.

6.5.3 Windup Against a Rate Limit

Section 1.4.9 treats windup at a position limit, and the rate limit of this valve adds a second case. The controller that writes the valve position is told the stem's slew rate through $\dot{u}_{\max}$ in (6.14), which is the outer loop when there is no cascade and the inner flow loop when there is. Its back-calculation therefore unwinds against what the actuator delivered rather than against what was asked for. This is the case that is easy to miss, because a valve limited to 6 %/min takes 16.7 min to travel its full range and is saturated in rate for minutes at a time while sitting nowhere near either stop. The outer loop in the cascade architecture is given an infinite slew rate instead, since its output is a flow setpoint and not a stem position. No external reset signal passes from the inner loop to the outer one, so while the stem is rate-limited the outer integral is unwound only at its own 0 to 100 % bound on the demand.

6.6 Results

Twelve scenarios are arranged as a laboratory sequence, listed in Table 6.3 with the mechanism each exercises and what a learner must do to complete it. Completion rules and the graded challenge work as Section 1.6.4 describes, and Reset keeps the learner's settings (Section 1.6.1). The challenge is judged on the case as given, so changing a model parameter or a disturbance, or firing an extra disturbance button, voids the attempt. Every figure reported below was reproduced by driving the shipped model headlessly at the fixed step of 0.01 min, exactly as the browser steps it.

6.6.1 Dosing by Hand

Scenario 1 opens in manual and steps the valve from 10 to 21 % in six moves of 5, 4, 0.8, 0.15, 0.15 and 0.9 %, holding each for 8 min. The steady-state answers are the whole lesson. The first move, five percent of the valve, carries the tank from pH 2.662 to pH 3.104, which is 0.088 pH per percent. The fourth move is 0.15 % of travel, from 19.80 to 19.95 %, and it carries the tank from pH 6.034 to

Table 6.3 Scenarios and Completion Requirements. The Letter Beside Each Number Gives the Kind: E Is an Experiment, D a Demonstration and C the Challenge.

#	Scenario	What it exercises	Required to complete
1 E	Dose by hand in six steps	the width of the permit in valve travel	Hold Manual through the six scripted demand steps from 10 to 21 % of valve to the end of the last 8 min hold at $t = 54$ min.
2 E	Step test at 12 % of valve	correct FOPDT identification	In Manual, read the settled pH before the 1 % demand step at $t = 10$ min and after it at $t = 40$ min, for the gain and the time to 63 percent of the change.
3 E	Step test at the setpoint	the same procedure, a different plant	In Manual, follow W_a through the 0.03 % step at $t = 10$ min and run to $t = 40$ min, then take the ratio of the gain to the one from case 2.
4 D	$K_c = 33$ run at pH 7	limit cycling from too much loop gain	Run the full hour in Auto on the fixed-gain PI loop with $K_c = 33$, then read the time in permit and the valve travel.
5 D	$K_c = 0.092$, acid load step	the opposite failure, too little	On the PI loop with $K_c = 0.092$, see the tank fall below pH 6 after the acid step at $t = 25$ min and follow the demand to $t = 70$ min, estimating when it would reach the 34.1 % the new load needs.
6 E	Gain scheduled, same step	scheduling from first principles	On the scheduled loop, bring the demand to 33.5 % after the step at $t = 25$ min and run to $t = 70$ min, comparing the low point and the time off permit with case 5.
7 E	Linearized, same step	inverting the measurement	On the linearized loop with $K_c = 3$, run through the step to $t = 45$ min and on to $t = 70$ min.
8 E	Buffering 0.6 to 10 mmol/L	an unmeasured curve change	Four runs, each structure in place before the batch change at $t = 15$ min and held for 75 min, with Reset between them: PI with $K_c = 0.092$, the scheduled loop, the linearized loop with $K_c = 3$, and the linearized loop with the assumed weak acid set to 10 mmol/L.
9 E	Influent flow 80 to 155 L/min	gain and dynamics moving together	On the linearized loop, run to $t = 45$ min with the second line on (influent above 150 L/min) and on to its shutdown at $t = 65$ min.
10 D	Seven hours of electrode fouling	drift inside the loop	On the linearized loop, run to the end of the shift at $t = 420$ min and through the electrode cleaning at $t = 430$ min to $t = 445$ min.
11 E	Header pressure and reagent strength	the exact scope of a cascade	Run the cascade through the header drop to $t = 30$ min and the weaker caustic to $t = 85$ min, then Reset, switch to the linearized loop before $t = 15$ min and run to $t = 30$ min.
12 C	Graded challenge, four upsets	structure before tuning	At $t = 240$ min, more than 45 % of the run inside the permit and the true pH within ± 1.0 of the setpoint, with the model parameters and disturbances as given.[a]

[a] With the shipped configuration (fixed-gain PI, $K_c = 0.092$, anti-windup off) the run is 8.5 % in permit and ends 4.06 pH from the setpoint, failing both parts. With the suggested configuration (cascade on the linearized signal, $K_c = 3$, $\tau_I = 4$ min, back-calculation) it is 49.7 % in permit, 4.7 points above the threshold, and ends 0.001 pH from the setpoint.

pH 9.035, which is 20.0 pH per percent and crosses the permit from just inside the bottom to just outside the top. The same actuator is worth 226 times more in the fourth move than in the first, with nothing about the tank, the valve or the reagent changed. Eight minutes is only 1.6 residence times, so the trend after the fourth hold reads pH 6.29 rather than the settled 9.04, and the scenario is more convincing when each step is left longer. The manual actuator resolves to 0.001 % of valve for this reason, with arrow keys that move it by 0.05 and 0.005 %, and a helper button places it at the travel that holds any setpoint the student types.

> **TRY IT IN THE APP** **1 · Dose It by Hand, and Find the Cliff**
>
> Load the case and press Run. The valve steps from 10 to 21 % of travel in six moves, each held for 8 min. Watch the pH trend after each move. The first move, five percent of the valve, lifts the tank from pH 2.662 to 3.104, while the fourth, only 0.15 % of travel, carries it across the whole permit.

6.6.2 Two Opposite Failures of a Fixed Gain

Scenario 4 closes the loop at pH 7 with the settings the flat-part step test honestly produced, $K_c = 33$ and $\tau_I = 5.2$ min, and applies no disturbance at all. The loop gain at the setpoint is $33 \times 33.78 = 1115$,

Table 6.4 Measured Performance After an Influent Strong Acid Step From 5 to 9 mmol/L at $t = 25$ min, All Values Read at $t = 70$ min. Every Run Uses $\tau_I = 5.2$ min and $\tau_D = 0$.

	PI	Scheduled	Linearized	Cascade
K_c	0.092	0.092	3.0	3.0
Minimum pH	2.54	4.16	3.51	3.53
Return inside permit (min)	never	15.0	21.3	21.4
Time off permit (min)	45.0	15.0	21.3	21.3
IAE (pH·min)	195.3	36.2	54.0	54.1
Valve travel (%)	3.9	19.9	15.1	15.5
Caustic (kg)	1.34	1.82	1.81	1.81
Cost ($)	540	181	256	257

so the loop is unstable by three orders of magnitude and the only thing that bounds the oscillation is the valve reaching its stops. Over one hour the tank cycles between pH 4.50 and pH 10.97, spends 57.5 of 60 min outside the permit for an in-permit score of 4.1 %, and moves the stem through 5231 % of its travel, which is fifty full strokes in an hour. The mean pH over the cycle is 8.57, inside the permit, so a plant that reported an average would look compliant.

> **TRY IT IN THE APP** **4 · A Gain Tuned on the Flat Part, Run at the Setpoint**
>
> Load the case, which closes the loop at pH 7 with $K_c = 33$ and $\tau_I = 5.2$ min and applies no disturbance, and run the full hour. Watch the valve travel trend swing between its stops while the pH cycles between 4.50 and 10.97. Then read the time in permit on the scorecard.

Scenario 5 uses the settings the setpoint deserves, $K_c = 0.092$, and steps the influent strong acid from 5 to 9 mmol/L at $t = 25$ min. The loop is perfectly stable and it never gets back. The tank falls to pH 2.54 and is still at pH 2.62 at $t = 70$ min, off permit for all 45 min since the step. The reagent demand has to reach 34.12 % and is climbing at 0.079 % per minute, so it would need roughly three hours. The mechanism is on the log-gain trend. At pH 2.5 the plant gain is 0.0384 pH per percent against 33.78 at the setpoint, a factor of 880, so the loop gain has fallen from 3.11 to 0.0035. The loop is not unstable. It is asleep.

6.6.3 Scheduling, Inversion and Cascade on One Load Change

Table 6.4 runs the identical acid load change on all four architectures with each one's proper gain, and reports the numbers the application computes for itself. The integrated absolute error and the off-permit time both accrue from the start of the run, so each is quoted at the same observation point of $t = 70$ min, 45 min after the step.

Scheduling and inversion both convert an unrecoverable excursion into a recovery of a quarter of an hour, and the fixed-gain column is worse on every measure at once. The scheduled loop is the faster of the two on this disturbance, reaching the permit 15.0 min after the step against 21.3 min and bottoming out at pH 4.16 against pH 3.51. That ordering is worth being honest about, because the IMC tuning the linearized plant deserves is more conservative than the effective gain the scheduler happens to apply out on the flat part, where K_c^{eff} reaches 3.14 at a measured pH of 5.20 against the 0.092 it holds at the setpoint. What inversion buys is not speed. It is that the integral in (6.14) accumulates percent-of-valve-minutes rather than pH-minutes, and a percent of valve means the same thing at pH 3 and at pH 7 while a pH unit does not. The valve moves 15.1 % of travel over the disturbance where the scheduler moves 19.9 %, and there is no slope estimate to amplify electrode noise. The cascade column is indistinguishable from the linearized one because this disturbance never touches the reagent flow, which is the result Section 6.6.5 depends on. Figure 6.3 draws the same table as bars.

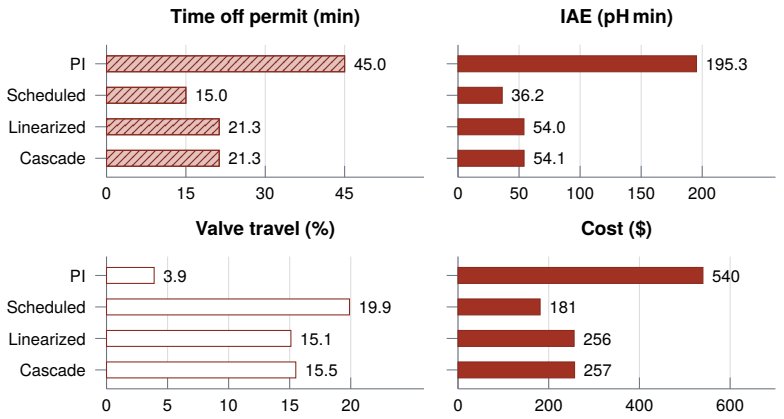


Figure 6.3 The four architectures of Table 6.4 after the influent strong acid step at $t = 25$ min, read at $t = 70$ min: time off permit (hatched bars), integrated absolute error and cost (solid) and valve travel (open). Notice that the fixed-gain PI loop moves the valve least and is worst on every other measure, and that the cascade matches the linearized loop because this disturbance never touches the reagent flow.

Table 6.5 Measured Performance After the Influent Weak Acid Steps From 0.6 to 10 mmol/L at $t = 15$ min, Read at $t = 80$ min. The Demand the New Influent Requires Is 53.130 % of Valve.

	PI	Scheduled	Linearized	Told the truth
K_c	0.092	0.092	3.0	3.0
Assumed $\hat{c}_b$ (mmol/L)	0.6	0.6	0.6	10
Final pH	4.06	6.69	5.71	7.00
Final demand (%)	23.67	52.97	50.36	53.13
Return inside permit (min)	never	31.9	never	10.4
Time off permit (min)	65.0	31.9	65.0	12.6
IAE (pH·min)	198.8	74.0	125.8	35.6
Valve travel (%)	3.8	33.4	30.5	133.5

DID YOU KNOW? The Cheapest Caustic Bill, the Dearest Run

From the simulator: after the acid step, the fixed-gain loop buys the least caustic, 1.34 kg against 1.81 to 1.82 kg for the other three structures, and still costs the most, \$540 against \$181 to \$257, because every minute off permit is charged (Table 6.4).

6.6.4 The Disturbance That Defeats Every Structure

Scenario 8 switches the upstream product at $t = 15$ min. Nothing at the inlet changes and the reagent demand nearly triples. Table 6.5 reports where each configuration has reached 65 min later.

The fixed-gain loop never arrives, reaching 23.67 % of the 53.13 % it needs. The scheduler reaches 52.97 % and the permit 31.9 min after the step. The linearized loop reaches 50.36 % and is still climbing at pH 5.71, which is better than the fixed gain and slower than the schedule, because its inverse is built from the clean curve. On that curve the tank at its low point of pH 4.31 appears to be 1.7 % of valve short of the setpoint when the demand is really 26 % short. Setting the assumed weak acid to 10 mmol/L, which is the same structure told the truth, returns the discharge to the permit 10.4 min after the step and holds pH 7.00 at exactly 53.13 % of valve. The last column is the argument for estimating the curve on line rather than fixing it [10, 15].

Of the 12.6 min off permit in the last column, 2.2 min accrue before the step. A controller told that the influent is buffered when it is not applies about sixteen times the intended loop gain on the clean stream's cliff, so it limit-cycles between pH 6.31 and 9.21 and crosses the caustic side of the permit. That is the same defect in the other direction, and it is worth showing.

The return of the clean stream at $t = 80$ min defeats every architecture equally, because 53 % of valve of caustic is entering a stream that now needs 20 %. Running both events in manual with a perfect feedforward, which changes the demand at the same instant as the composition and which no plant can build, still carries the tank down to pH 5.37 on the buffering step and up to pH 10.16 on the return, and leaves 13.9 min outside the permit over the run. That excursion belongs to the tank and not to the controller. No feedback structure removes an unmeasured composition change, and designing for graceful degradation under one is a legitimate goal in its own right [14].

6.6.5 What the Inner Loop Does and Does Not Reject

Scenario 11 drops the caustic header pressure to 0.6 of nominal at $t = 15$ min, which by (6.8) reduces the reagent flow by 23 % without the stem moving at all. With the inner flow loop closed the transmitter reports the change within seconds and the valve opens to compensate, and the peak pH excursion is 0.91 pH. With the same outer loop and no inner loop the pH controller has to discover the disturbance through five minutes of tank, the 0.15 min transport delay and the 0.20 min electrode, and the peak excursion is 2.25 pH. Over the 90 min run the single loop is off permit for 50.9 min with an IAE of 121.7 pH·min against 15.3 min and 39.1 pH·min for the cascade. A caustic delivery arriving 15 % weaker at $t = 70$ min is invisible to the inner loop, because the flow is exactly what was asked for, and it is equally invisible to the inverse of (6.16), which carries the nameplate concentration. The demand setpoint does not move, the tank falls 1.92 pH below the setpoint at $t = 72.6$ min, and only the outer loop recovers it through the pH. The general rule is that an inner loop rejects everything appearing in its own measurement and nothing else. Volume 3, Chapter 2 develops that rule on a level loop with two disturbance paths.

🕮 **Pause.** Try Exercise 6.3 before reading on.

6.6.6 Fouling and the Graded Challenge

Scenario 10 runs seven hours with nothing wrong except a film building on the glass at 14 % per hour. The controller holds the reading at 7.00 for the whole shift while the true discharge drifts to pH 7.886, because an electrode reading low makes the controller overdose, and the reagent demand moves by 0.014 % of valve over the whole shift. The run never leaves the permit and no permit alarm fires. The only warning, from $t = 235.7$ min when the coating passes 55 %, reports that the electrode needs cleaning and not where the discharge has gone. The second effect is less visible and more dangerous, because at 98 % fouling the electrode lag of (6.10) has grown from 0.20 to 1.96 min, a factor of ten of pure phase lag inside the loop. Cleaning the electrode at $t = 430$ min then steps the measurement by 0.88 pH, which the controller cannot distinguish from the tank moving. The reading jumps to pH 7.76 and the controller cuts the caustic, so a maintenance action moves the valve and the tank as a process upset would, here carrying the true pH from 7.88 to 7.10 within 2.2 min.

The challenge applies four unannounced composition changes over four hours, a buffering step at $t = 20$ min, an acid load rise at $t = 60$ min, an acid load fall at $t = 100$ min and the return of the clean stream at $t = 130$ min, with the valve slowed to 6 %/min and anti-windup switched off in the starting configuration. The case is passed with more than 45 % of the run inside the permit and a final pH within ± 1.0 of the setpoint. Table 6.6 reports three configurations over the full 240 min.

The answer is the whole stack and the scorecard says what each layer is worth. Linearization is required because two of the four upsets change the shape of the curve. Cascade is required because a slew-limited valve is the actuator an inner flow loop was invented for. Anti-windup is not a refinement here, since switching back-calculation off takes the same cascade from 49.7 to 10.2 % in permit, ends the run at pH 2.49 rather than pH 7.00, and multiplies the valve travel by 3.9 while raising the cost

Table 6.6 Graded Challenge Over 240 min With a 6 %/min Valve, Four Unannounced Composition Changes, and Every Figure Read at the End of the Run.

	As loaded	Suggested	Suggested, no anti-windup
Architecture	PI	cascade	cascade
K_c, τ_I (min)	0.092, 5.2	3.0, 4.0	3.0, 4.0
Anti-windup	off	back-calc.	off
In permit (%)	8.5	49.7	10.2
Final pH	11.06	7.00	2.49
IAE (pH·min)	836	291	729
Valve travel (%)	15	134	524
Caustic (kg)	4.9	6.8	7.2
Cost ($)	2639	1453	2591

from $1453 to $2591. Half the afternoon is spent outside the permit even when everything is right, because the four upsets move the demand the setpoint requires through 19.90, 53.13, 67.35, 47.80 and 14.57 % of valve, a total of 100 % of travel on a stem that moves at 6 % a minute into a tank of 400 L. The honest engineering answer to that plant is a bigger tank, a smaller valve and a feedforward from an influent analyzer, and knowing that no control structure buys them is part of the lesson.

Figure 6.4 shows the application after that same buffering scenario has run for three hours on the fixed-gain architecture.

6.7 Technology in Practice

The loops of Section 6.5 read AIT-401 through a delay, a lag and a fouling term, and move FCV-401 through a lag, a rate limit and a header pressure. Figure 6.5 draws them with the devices a neutralizer would carry in place of the schematic's symbols.

What Measures It

AIT-401 is a glass electrode with a reference electrode, wired to a transmitter that converts its millivolts to pH. Dipping the probe into an open vessel of the liquid is a common installation in water treatment [18]. The transport delay θ of Equation (6.10) stands for the pipe between the injection point and the probe, the lag τ_m for the response of the glass, and the fouling terms $\kappa\phi$ and $\delta\phi$ for a coating that slows the electrode and shifts its reading. Calibration uses buffers that bracket the expected range, most often pH 4, 7 and 10 [18]. No accuracy or response time is given here, because none was found in an open source that names no maker.

> **TECHNOLOGY** **Sensor: The Glass pH Electrode**
>
> A glass electrode ends in a thin bulb that passes hydrogen ions and is filled with a buffer at pH 7. A difference between the process liquid and that buffer sets up a voltage across the glass in proportion to the pH difference, and a reference electrode completes the circuit. The slope is 59.17 millivolts per pH unit at 25 °C and 68.11 at 70 °C, so the probe carries a temperature sensor for compensation. The glass must stay wet, and the same hydration slowly wears it away, so every electrode has a limited life. At high pH, sodium ions add a voltage of their own. In dirty service such as wastewater the bulb needs regular cleaning, with no solid tool touching the fragile glass [18].

What Moves It

In a plant, FCV-401 is a caustic valve whose actuator is driven through a positioner, a closed-loop controller in its own right that holds the stem at the commanded position [18]. The lag τ_v and the slew limit $\dot{v}_{\max}$ of Equation (6.7) stand for that actuator, and the term $\sqrt{p_h}$ of Equation (6.8) for the

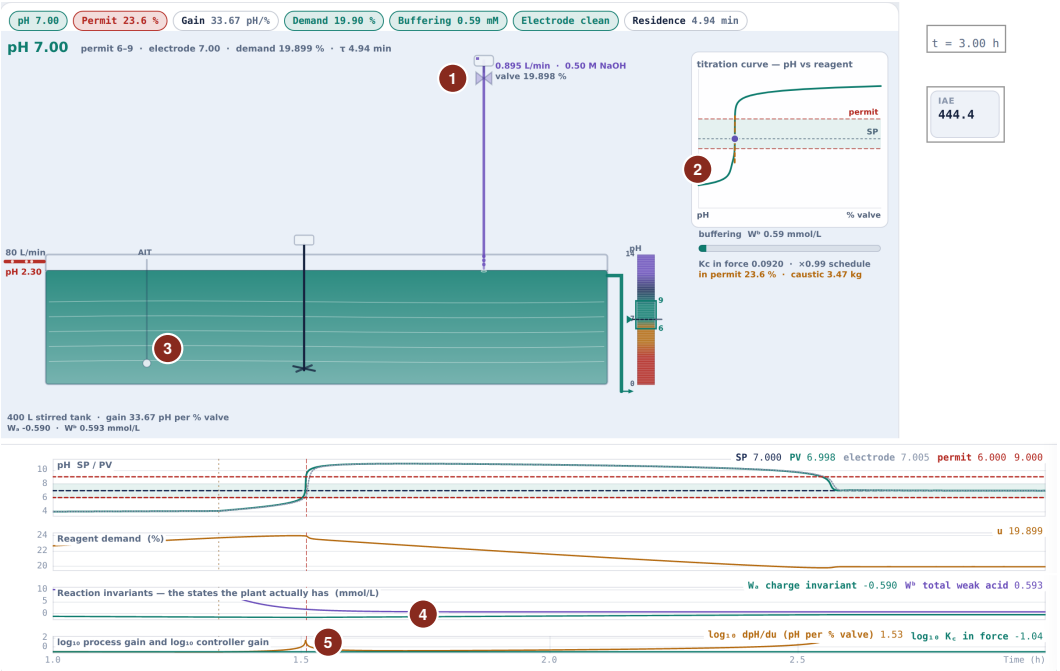


Figure 6.4 Scenario 8 at $t = 3.00$ h (framed readout), after the buffering step at $t = 15$ min and the return of the clean stream at $t = 80$ min, with the fixed-gain PI architecture. The figure is a composite: the plant view above four of the six trend strips, with the elapsed time and the IAE tile moved beside the plant view. The tank has settled back to pH 6.998 at a reagent demand of 19.899 %, and the scorecard records 23.6 % of the run inside the permit, 3.47 kg of caustic and an IAE of 444.4 pH·min. The titration curve beside the tank carries the operating point and the shaded permit band, the invariant strip shows W_b dropping from 10 (Table 6.5) back to 0.593 mmol/L when the clean stream returns, and the log-gain strip shows the process gain swinging while the controller gain holds a straight line. Callouts: (1) the caustic valve FCV-401 and its flow, (2) the operating point on the titration curve, (3) the electrode AIT-401, (4) the invariant strip, (5) the log-gain strip.

supply header behind it. FT-401 measures the caustic flow for the inner loop of the cascade, which is why a header pressure change is rejected there before it reaches the tank.

> **TECHNOLOGY** **Final Element: The Caustic Valve and the Acid Valve**
>
> Where a stream can stray to either side of neutral, one pH controller can drive two reagent valves in exclusive sequence. Both are shut at half output, the acid valve opens as the output rises toward full scale and the caustic valve as it falls toward zero [18]. The chapter's acid trim pump plays the acid side. An air-to-open valve fails closed when its air supply is lost, and the risks of the process, not habit, should decide which way each valve fails [18]. On this tank a caustic valve that fails shut sends the influent to the outfall at the acidity it arrived with.

Where the Controller Runs

A distributed control system runs production at one site in water and wastewater treatment and in chemical manufacturing, and its loops run with cycle times from milliseconds to minutes [19]. AIC-401 and FIC-401 execute every 0.02 min (1.2 s), well inside that range. The inverse titration curve of the linearized architecture fits a multi-segment characterizer, a function block of such a

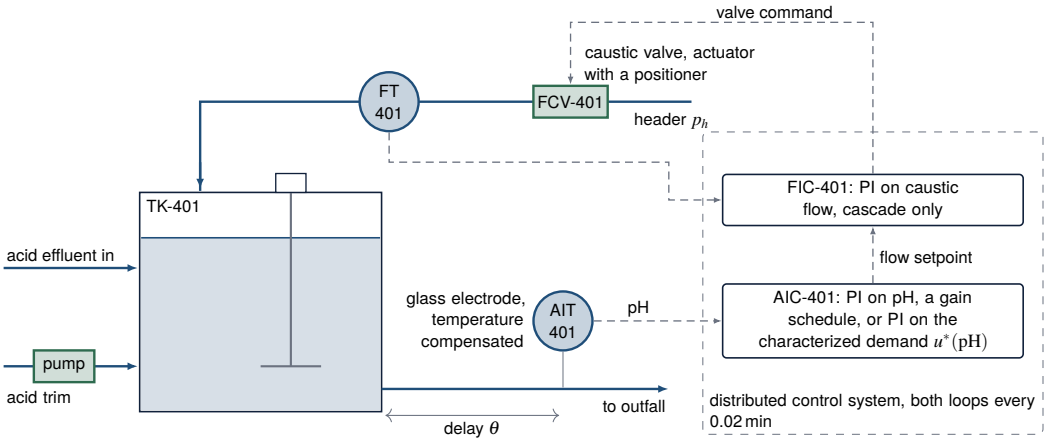


Figure 6.5 The loops of the chapter with plant devices in place of the schematic's symbols. AIT-401, a glass electrode on the discharge, reads the pH after the transport delay θ. AIC-401 on the control platform acts on the pH directly, through a gain schedule, or through the characterized demand, and in the cascade it sets the flow that FIC-401 holds by moving FCV-401 against FT-401. Without the cascade, AIC-401 writes the valve command itself. Notice that the caustic enters at the top of the tank while the pH that the permit judges is read downstream, after mixing and transport.

system that can be programmed to follow almost any continuous curve [18]. The gain schedule is the same PID block with its gain set by another variable, here the slope of the assumed curve at the measured pH.

> **IN PRACTICE** **Mapping the Gain Before Tuning**
>
> An instrument handbook advises exploring a process before tuning it. With the controller in manual, step the output several times across the safe range and compare the size of each response. If the gain grows from one end of the output range to the other, a valve with a different characteristic can counteract it. If the gain follows the measured value instead, the remedy is an adaptive gain controller, whose proportional setting varies as the process changes [18]. A pH loop is the second kind, and its manual steps have to shrink as the tank nears neutral.

What Is Modernizing It

Three developments touch this loop. The first is fieldbus, which lets the PID function block run inside the field devices, in the transmitter or in the valve positioner [18]. The second is nonlinear predictive control. Its industrial models are usually linear dynamic subsystems with an added nonlinearity [20], which is the shape of this plant, two linear invariants and one algebraic map to pH. The third is self-tuning PID, which has done well in the market [20], although the outer loop here would need one that follows a gain moving by four orders of magnitude.

6.8 Limitations

The model is not a design model for a neutralization plant. One perfectly mixed tank cannot support a claim about how many stages a real effluent needs, and plants that hold a tight permit use two or three in series precisely because one cannot [1]. There is no ratio control and no influent analyzer, so the first structural improvement a real plant would make is absent, and scenario 9 would largely disappear with it. Volume 3, Chapter 6 computes a dose from an upstream analyzer and keeps the downstream measurement as a slow trim. Carbonate, temperature and activity effects are all outside the chemistry, and each of them moves the titration curve on a timescale comparable with the control

problem. The valve has no stiction, which matters because a stiction limit cycle looks like the one in scenario 4 and has an entirely different cause. Electrode fouling is one linear state and cannot produce the failure that matters most in service, which is a probe that stops responding and reports a flat and plausible trace. Economics are a caustic price and a flat charge per minute off permit, whereas a real consent order is triggered by a grab sample, so the cost tile compares two runs rather than predicting compliance. Repeatability is itself unrealistic, because freezing the integration step and the controller period reproduces one trajectory on every machine while a distributed control system does no such thing. Where the individual assumptions break is set out in Section 6.3.5.

Two elements of the scenarios are stylized on purpose. A buffering change that arrives as an instantaneous step is not what an upstream product change does, but a step is the disturbance whose response can be read off a trend. A valve slowed to 6 %/min in the challenge is slower than a healthy actuator, although it is the only way to make rate saturation visible on a plant whose dominant time constant is five minutes.

> ⚠ **OPEN CHALLENGE** **A Controller That Learns Its Own Curve**
>
> Both the schedule and the inverse rest on an assumed titration curve, and buffering is not measured on line. A controller that found the curve by itself would remove it, and a simple universal controller that adjusts itself has long been a goal of process engineers, with the relay autotuner of PID control only a partial answer [9]. A survey of industrial predictive control found that only two adaptive versions had reached the market despite a strong incentive, and it did not expect that to change without a theoretical breakthrough. Self-tuning PID controllers, by contrast, have sold well [20]. Progress would be an estimator that tracks the curve from the dosing record and the pH alone, and that an operator could trust.

6.9 Discussion

The three structural remedies in this application are not alternatives of equal standing. Scheduling leaves the controller working in pH, where an error of one unit is a few micromoles per liter near the setpoint and several millimoles per liter out on the flat part, so the proportional term is corrected while the integral accumulates a quantity with no consistent meaning. Inversion corrects the whole error signal and reduces to one characterizer block on the transmitter in a modern control system, which is the same function block that linearizes a valve pointed at a measurement instead. Cascade corrects neither nonlinearity and removes the actuator from the outer loop, which is why Table 6.4 shows it as indistinguishable from inversion on a composition change and Section 6.6.5 shows it worth a factor of 2.5 in peak excursion on a header pressure change. The order in which they are introduced is the order in which a plant learns them, and each is measured against the same scorecard. Volume 7, Chapter 3 sets a gain schedule against feedback linearization on a mechanical plant.

The equipment decisions dominate all three. The reagent valve passes 4.5 L/min against a duty of 0.8955 L/min, which is five times oversized and is normal, because the valve is specified for the worst influent anyone imagined. Halving the caustic concentration from 0.50 to 0.25 mol/L moves the design demand from 19.899 to 39.798 % of travel and widens the permit band from 0.155 to 0.312 %, at the cost of twice the reagent volume. Fitting equal-percentage trim makes the installed gain far more nearly constant across the range without any control system change at all. The application exposes the capacity, the rangeability, the caustic strength and the tank volume as adjustable parameters so that the trade between capital and control structure can be computed rather than argued, and the outcome of that computation is often that the money is better spent on the valve than on the algorithm.

The model also supports methods beyond the implemented loops. The two linear invariant states with an algebraic output map are the natural form for a nonlinear model predictive controller, which enforces the valve rate limit of the challenge explicitly instead of defending against it with back-calculation. Moving horizon estimation on the same equations can infer the unmeasured c_a and c_b from the pH measurement and the reagent demand, which is the on-line titration curve estimate that Table 6.5 shows to be worth 52 min of permit time [10, 15]. Identifying the static map from data

rather than from equilibrium constants gives the Wiener structure that has been applied to the same plant [11]. Nothing in the model has to change for any of them, because the difficult part was already separated out into one algebraic line.

6.10 Conclusion

This chapter studied an effluent neutralization tank as an interactive laboratory for plants whose gain is a function of the operating point rather than a constant. The simulator integrates five states, the two reaction invariants, the valve stem, the influent pump and the electrode fouling fraction, and recovers the pH each step by bisecting a charge balance, so the plant is linear and the measurement is not. The reagent valve holds pH 7.00 at 19.899 % of travel and the whole permit band of pH 6 to 9 occupies 0.155 % of it, while the gain from valve to pH runs from 5.191×10^{-3} to 101.98 pH per percent across one stroke, a factor of 1.96×10^{4}.

Settings obtained correctly from a step test at 12 % of valve produce a limit cycle between pH 4.50 and pH 10.97 at the setpoint with 5231 % of valve travel in an hour, and settings obtained correctly at the setpoint leave the tank at pH 2.62 three quarters of an hour after a load change. Scheduling the gain returns the discharge to the permit 15.0 min after that load change and inversion of the titration curve does so in 21.3 min, against never for the fixed gain. An inner reagent flow loop reduces the excursion from a header pressure change from 2.25 to 0.91 pH and does nothing about a weaker caustic delivery.

A fuller model would add a second tank in series, ratio control from the influent flow meter, valve stiction as an independent source of cycling, and a carbonate species so that the curve has more than one inflection. The pH neutralization tank is an accessible browser platform for learning when a loop needs a different structure rather than a better tuning.

KEY POINTS

- The tank mixes caustic smoothly, and only the pH scale turns that smooth change into a cliff.
- The whole permit of pH 6 to 9 is 0.155 % of the reagent valve's travel, and across one stroke the gain changes by a factor of 1.96×10^{4}.
- A step test measures the gain where it is made: the same procedure returns gains that differ by a factor of 691.
- After the acid step, scheduling returns the discharge to the permit in 15.0 min and inversion in 21.3 min. The fixed gain never does.
- An inner flow loop rejects a header pressure change but not a weaker caustic delivery.

Where else this applies. The gain of Volume 7, Chapter 3 also moves with its operating point, and there a schedule meets feedback linearization.

References

[1] F. G. Shinskey, *pH and pIon Control in Process and Waste Streams*. New York, NY: John Wiley & Sons, 1973.

[2] G. K. McMillan and R. A. Cameron, *Advanced pH Measurement and Control*, 3rd ed. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2004.

[3] U.S. Environmental Protection Agency, "Secondary treatment, 40 CFR 133.102," Code of Federal Regulations, Title 40, Volume 24, U.S. Government Publishing Office, govinfo.gov, 2024.

[4] T. K. Gustafsson and K. V. Waller, "Dynamic modeling and reaction invariant control of pH," *Chemical Engineering Science*, vol. 38, no. 3, pp. 389–398, 1983.

[5] ——, "Myths about pH and pH control," *AIChE Journal*, vol. 32, no. 2, pp. 335–337, 1986.

[6] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[7] C. Kravaris and C.-B. Chung, "Nonlinear state feedback synthesis by global input/output linearization," *AIChE Journal*, vol. 33, no. 4, pp. 592–603, 1987.

[8] R. A. Wright and C. Kravaris, "Nonlinear control of pH processes using the strong acid equivalent," *Industrial & Engineering Chemistry Research*, vol. 30, no. 7, pp. 1561–1572, 1991.

[9] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed. Princeton, NJ: Princeton University Press, 2021.

[10] M. A. Henson and D. E. Seborg, "Adaptive nonlinear control of a pH neutralization process," *IEEE Transactions on Control Systems Technology*, vol. 2, no. 3, pp. 169–182, 1994.

[11] A. D. Kalafatis, L. Wang, and W. R. Cluett, "Linearizing feedforward–feedback control of pH processes based on the Wiener model," *Journal of Process Control*, vol. 15, no. 1, pp. 103–112, 2005.

[12] T. J. McAvoy, E. Hsu, and S. Lowenthal, "Dynamics of pH in controlled stirred tank reactor," *Industrial & Engineering Chemistry Process Design and Development*, vol. 11, no. 1, pp. 68–70, 1972.

[13] R. C. Hall and D. E. Seborg, "Modelling and self-tuning control of a multivariable pH neutralization process part I: Modelling and multiloop control," in *1989 American Control Conference*. IEEE, 1989, pp. 1822–1827.

[14] T. K. Gustafsson and K. V. Waller, "Nonlinear and adaptive control of pH," *Industrial & Engineering Chemistry Research*, vol. 31, no. 12, pp. 2681–2693, 1992.

[15] M. Gulaian and J. Lane, "Titration curve estimation for adaptive pH control," in *1990 American Control Conference*. IEEE, 1990, pp. 1414–1419.

[16] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: PID controller design," *Industrial & Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252–265, 1986.

[17] K. J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2006.

[18] T. R. Kuphaldt, *Lessons in Industrial Instrumentation*, version 2.33 ed. Open textbook, Creative Commons Attribution 4.0 license, ibiblio.org/kuphaldt/socratic/sinst, 2024.

[19] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, "Guide to operational technology (OT) security," National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-82r3, 2023.

[20] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

6.11 Exercises

6.1 Concept check · 2 min

The step tests at 12 % of valve and at the setpoint return gains of 0.082 and 56.66 pH per percent, while the invariant W_a moves by the same 0.28 mmol/L per percent of valve in both. Where does the difference between the two gains arise?

(a) In the charge balance, an algebraic map from the invariants to pH that is steepest near neutrality

(b) In the mixing of the tank, whose time constant shortens as the contents approach the neutral point

(c) In the reagent valve, whose linear trim passes more caustic per percent near the design travel

(d) In the electrode, whose lag and transport delay stretch a small change into a large reading

6.2 Concept check · 1 min

The influent weak acid rises from 0.6 to 10 mmol/L while the strong acid stays at 5 mmol/L. What does the model predict for the inlet pH, the reagent demand at pH 7 and the process gain there?

(a) The inlet pH falls by about one unit, and the demand and the gain both stay nearly as they were

(b) The inlet pH barely moves, the demand nearly triples and the gain falls by a factor of about 16

(c) The inlet pH barely moves, the demand stays nearly fixed and the gain falls by a factor of about 16

(d) The inlet pH barely moves, the demand nearly triples and the gain rises by a factor of about 16

6.3 Concept check · 2 min

In scenario 11 the cascade meets a drop of the caustic header pressure to 0.6 of nominal and, later, a caustic delivery that is 15 % weaker. Which of the two does the inner reagent flow loop reject?

(a) Both, because each one changes the amount of base that reaches the tank per percent of valve travel

(b) Neither, because the lag of the flow transmitter hides both from the inner loop until the pH has moved

(c) The pressure drop only, because it changes the measured flow and the weaker caustic leaves that flow as requested

(d) The weaker caustic only, because the inverse of the titration curve carries the true strength of the delivery

6.4 Theory and calculation · 5 min

At the design point the influent is $Q_1 = 80\,\mathrm{L/min}$ with $c_a = 5.0\,\mathrm{mmol/L}$ and $c_b = 0.6\,\mathrm{mmol/L}$, the caustic is $C_b = 0.50\,\mathrm{mol/L}$, the valve passes 4.5 L/min at full travel and the acid trim is off. Take $\mathrm{p}K_a = 4.76$ and $\mathrm{p}K_w = 14$.

(a) Show from Equation (6.4) that $W_a = -f\,W_b$ at pH 7, and find $f = K_a/(K_a + h)$. *Checkpoint:* $f = 0.9943$.

(b) Use the steady forms of Equations (6.2) and (6.3) to find the reagent flow and the valve travel that hold pH 7. *Checkpoint:* 0.8955 L/min and 19.90 %.

(c) Repeat part (b) for $c_b = 10\,\mathrm{mmol/L}$. *Checkpoint:* 2.391 L/min and 53.13 %.

6.5 Theory and calculation · 8 min

The process gain at the setpoint follows from the buffer term. Take pH 7, $K_a = 1.738 \times 10^{-5}$, $K_w = 10^{-14}$, $C_b = 0.50\,\mathrm{mol/L}$ and a valve of 4.5 L/min with linear trim at nominal header pressure.

(a) At the design point the tank holds $W_b = 0.593\,\mathrm{mmol/L}$. Find J from Equation (6.5). *Checkpoint:* $J = 35.7$.

(b) Find $\partial\mathrm{pH}/\partial W_a$ from Equation (6.6). *Checkpoint:* -1.215×10^5 pH per mol/L.

(c) Find the gain from valve travel to pH with Equation (6.11) and $Q_t = 80.90\,\mathrm{L/min}$. Use $W_a = -f\,W_b$ with $f = K_a/(K_a + h)$ to simplify the bracket. *Checkpoint:* 33.8 pH per percent.

(d) With $c_b = 10\,\mathrm{mmol/L}$ the tank holds $W_b = 9.710\,\mathrm{mmol/L}$ and $Q_t = 82.39\,\mathrm{L/min}$. Find J and the gain. *Checkpoint:* $J = 554$ and 2.14 pH per percent.

6.6 Theory and calculation · 5 min

The step test at 12 % of valve gives a process gain of 0.082 pH per percent, $\tau_p = 4.94\,\mathrm{min}$ and $\theta_p = 0.35\,\mathrm{min}$.

(a) Find K_c from Equation (6.12) with a closed-loop target of 1.5 min, and the loop gain $K_c K_p$ that the rule designs for. *Checkpoint:* 32.6 percent of valve per pH unit, and 2.67.

(b) The loop is closed at the setpoint with $K_c = 33$, where the tangent gain is 33.78 pH per percent. Find the loop gain. *Checkpoint:* 1115.

(c) With $K_c = 0.092$ the tank sits at pH 2.5 after a load step, where the gain is 0.0384 pH per percent. Find the loop gain there and at the setpoint. *Checkpoint:* 0.0035 and 3.11.

6.7 App challenge · 8 min

Load scenario 2, A step test on the flat part, and change no setting. The reagent demand steps from 12 to 13 % at 10 min. Run until just past 40 min with the default seed and export the CSV file. The column `ph` holds the true tank pH.

(a) Read `ph` in the rows at 10 and 40 min and compute the gain. *Checkpoint:* 2.790 and 2.872, within 0.002, and 0.082 pH per percent.

(b) Find the time after the step at which `ph` has covered 63.2 % of that change. *Checkpoint:* 5.3 min, within 0.1 min.

(c) Load scenario 3, The same step test at the setpoint, whose step is 0.03 % of valve at 10 min. Run until just past 40 min, read the same two rows and compute the gain and its ratio to the gain of part (a). *Checkpoint:* 7.003 and 8.701, 56.6 pH per percent, a ratio near 690.

Compare the change of the column `Wa_mmol` per percent of valve in the two files.

6.8 App challenge · 8 min

Load scenario 5, A gain tuned at the setpoint, and what it costs elsewhere. The influent strong acid steps from 5 to 9 mmol/L at 25 min. Change no setting, run until just past 70 min with the default seed and export the CSV file.

(a) Find the lowest `ph` after the step. In the row at 70 min read `ph`, the reagent demand `demand_pct` and the time off permit `metric_offspec_min`. *Checkpoint:* 2.54. Then 2.62, 23.73 % and 45.0 min.

(b) From `demand_pct` at 40 and 60 min compute the climb rate. Estimate how long after 70 min the demand would need to reach the 34.12 % that the new load requires. *Checkpoint:* 0.078 % per minute and about 130 min.

(c) Load scenario 6, Gain scheduling, and repeat the run. Find the lowest `ph`, the time off permit at 70 min and the largest value of the gain in force, `kc_eff`. *Checkpoint:* 4.16, 15.0 min and 4.33.

Accept a pH within 0.02, a demand within 0.05 % and a time off permit within 0.2 min.

6.9 App challenge · 9 min

Load scenario 8, The buffering changes without warning, and set the Control panel to Advanced. The influent weak acid steps from 0.6 to 10 mmol/L at 15 min. Use the default seed, run each case until just past 80 min and export the CSV file.

(a) Set Control architecture to Linearized and Controller gain to 3, then run. In the row at 80 min read `ph`, `demand_pct` and `metric_offspec_min`. *Checkpoint:* 5.71, 50.36 % and 65.0 min.

(b) In the group for the controller's titration curve, set the weak acid of the titration model to 10 mmol/L. Press Reset, run again and read the same row. *Checkpoint:* 7.00, 53.13 % and 12.6 min.

(c) In the file of run (b) read `metric_offspec_min` in the row at 15 min, before the batch changes, and explain it. *Checkpoint:* 2.2 min.

Accept a pH within 0.02, a demand within 0.05 % and a time within 0.2 min.

6.10 Application area · 3 min

The reagent valve passes 4.5 L/min at full travel, and the design influent needs 0.8955 L/min of 0.50 mol/L caustic at pH 7. The permit of pH 6 to 9 spans 0.155 % of valve travel.

(a) Find how many times the valve is oversized and the width of the permit as a reagent flow. *Checkpoint:* 5.0 times and 7.0 mL/min.

(b) The plant changes to 0.25 mol/L caustic. Estimate the new demand and the new width of the permit in valve travel, and name the cost. *Checkpoint:* 39.8 % and about 0.31 %.

6.11 Application area · 4 min

Scenario 10 fouls the electrode at 14 % per hour for a shift of 420 min. A fully fouled electrode reads 0.9 pH low, and its lag is the clean 0.20 min multiplied by $1 + 9\phi$.

(a) Find the time at which the fouling fraction ϕ passes 0.55, where the app raises its electrode warning, and ϕ at the end of the shift. *Checkpoint:* 235.7 min and 0.98.

(b) Find the electrode lag and the reading offset at the end of the shift, and the true pH while the controller holds the reading at 7.00. *Checkpoint:* 1.96 min, -0.88 pH and pH 7.88.

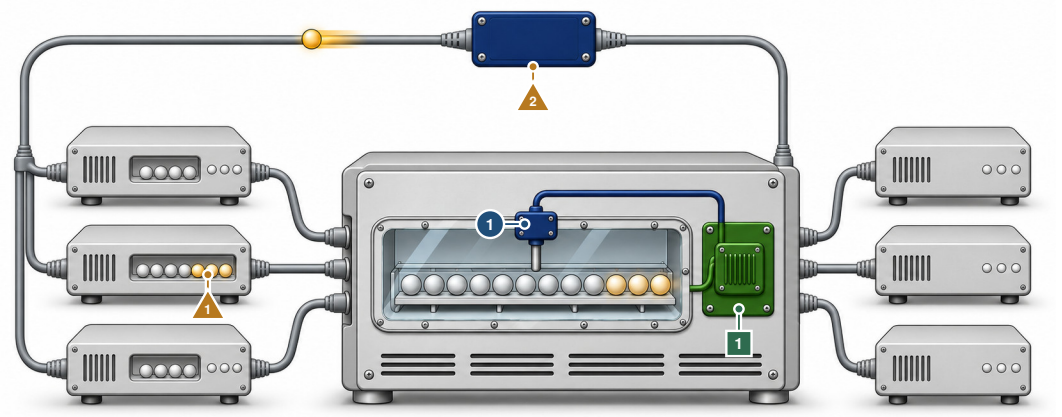
(c) Explain why no permit alarm fires and why the pH trend that the controller sees shows nothing wrong.

6.12 Review the agent · 5 min

An AI assistant is asked to move the fixed-gain loop of scenario 5 to the linearized architecture and replies as follows. "The IMC rule at the setpoint gave $K_c = 0.092$ with $\tau_I = 5.2$ min, and the loop is stable there. Linearization changes only the signal that the controller reads. The tank, the valve and the electrode are the same, so the time constant and the dead time are unchanged. Keep $K_c = 0.092$ and $\tau_I = 5.2$ min, select Linearized, and the acid load step will be rejected about as fast as the gain schedule rejects it." Find the flaw in the proposal and say how it could be checked.

CHAPTER 7

Delay Differential Equations and the Stability Boundary in a TCP Congestion Simulator



The controller moves 1 the share of packets the router marks

The controller reads 1 the length of the line of packets in the router

What disturbs it 1 more senders sharing the link 2 a longer route, which lengthens the round trip

Notice that a mark made at the router travels back over the upper cable, past the dark box that stands for the round-trip delay, before any sender can slow down. Illustration, not to scale.

IN PLAIN TERMS Holding a Router Queue Through a Delay

The main idea. Packets from many computers wait in a line at a router before one busy link. The router keeps that line near a target length by marking a few packets, and each sender that learns of a mark slows down.

The challenge. A mark reaches its sender only after a trip out and back, so the router acts on old news. On a longer trip each correction of a fixed setting comes later and hits harder, until the line swings.

Why it matters. A line that is too long delays every packet, even on a link that looks healthy.

THREE WAYS TO READ THIS CHAPTER

Explore	12 min	the first page, the history of congestion collapse and of the PI queue controller (Section 7.2), the fact on the queue setpoint, the try-it segments of scenarios 4 and 10 and the fact on the challenge (Section 7.8), and the key points
Study	50 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	15 min	the first page, the results (Section 7.8), Technology in Practice (Section 7.9), Limitations (Section 7.10) and the open challenge

The control question A router that marks packets to hold its queue acts on senders that learn of each mark one round trip later, and that round trip grows with the queue being held. At what round-trip time does a fixed PI tuning lose its phase margin, and does the simulated loop stop settling where the characteristic equation says it should?

Learning objectives

- Compute the operating point of the window and queue model and test whether it exists.
- Explain why the process gain rises as the cube of the round-trip time and falls as the square of the session count.
- Compute the phase margin of the PI loop behind a dead time and locate the critical round trip.
- Tell an actuator with no feasible operating point from a badly tuned loop.
- Explain how the queue setpoint sets the dead time of its own loop.

Prerequisites Sections 1.4.1 (dead time), 1.4.12 (phase margin), 1.4.3 (controller action), 1.6.2 and 1.4.9 (velocity form and windup). Chapter 6 is useful background on a gain that moves with the operating point.

The app https://apmonitor.com/apps/tcp-congestion/

See also Volume 5, Chapter 2 and Volume 5, Chapter 3 for control through a transport delay and a slow analyzer, and Volume 5, Chapter 8 for a loop closed over a delayed channel. Volume 7, Chapter 13 for a second delay differential equation with a stability boundary and Volume 7, Chapter 3 for a plant that the setpoint chooses.



7.1 Introduction

The internet moves data by running a feedback loop across every congested path. Senders raise their transmission rate until a router signals congestion, then cut back, and the equilibrium reached by that adjustment is what allocates a shared link among the flows using it [1, 2]. The congestion signal reaches a sender only after a full traverse of the path. A router that decides now to mark a packet cannot change any sender's rate until the marked packet has reached its destination and the acknowledgment has come back, and the effect of the changed rate does not appear at the router for a further round trip. The loop therefore carries a dead time equal to the round-trip time of the path, and no protocol design removes it.

Dead time appears in many process control problems, usually as a transport lag appended to a lumped model of the process, as in the first-order-plus-dead-time model of Section 1.4.1. This system is different in three ways that are worth separating. The delay is inside the state equations rather than in series with them, because the window adaptation at time t is driven by the marking probability the sender learned about one round trip earlier. The length of that delay is a function of the queue, which is the very state the controller regulates, so the delay is state dependent and the setpoint moves the plant. The steady-state process gain is proportional to the cube of the same round-trip time, so lengthening the delay subtracts phase and adds gain at the same instant. A loop with a fixed tuning can therefore be pushed across its stability boundary by a route change that touches nothing else.

The physical content of the model is a mass balance and a rate law. The router queue is an inventory whose derivative is the difference between the packet arrival rate and the link drain rate, which is the accumulation that governs any surge vessel, and the congestion window obeys a rate law whose growth is proportional to the fraction of acknowledgments that were not marked and whose decay is proportional to the fraction that were. A separation of time scales between a window that adapts over several round trips and a queue that fills in a fraction of one is what allows an entire congested path to be represented by two ordinary differential equations [3, 4]. Experiments built on those two states cover process order, transport delay, the phase a dead time removes at the crossover frequency, actuator saturation and anti-windup, feedforward from a computed operating point, and a process gain that varies with the load rather than staying constant [3, 5]. The same combination appears in the process industries wherever the measurement is remote from the actuator, and a queue managed badly costs interactive latency on a link that every throughput measurement reports as healthy.

Fluid-flow models of TCP make all three effects computable. The window and queue equations of Misra, Gong and Towsley reduce a congested path to two states and one delay [6], and the control-theoretic linearization of Hollot and colleagues turns those two states into a second order transfer function behind a pure dead time [7–9]. The same model supports the classical feedback account of congestion control given by Low, Paganini and Doyle [10] and connects to the wider literature on time-delay systems, where stability is decided by the roots of a transcendental characteristic equation rather than a polynomial [11]. Volume 7, Chapter 13 treats a second delay differential equation, the regenerative chatter of a milling cutter, whose stability boundary moves with the spindle speed. However, the congestion material is normally presented either as networking research, in which the control content is a few paragraphs of Nyquist argument, or as delay-systems theory, in which the plant is an abstract example. Neither arrangement lets the reader adjust the delay and measure the phase margin that results.

This chapter works with a browser-based fluid-flow simulator of TCP and active queue management built around one adjustable quantity, the round-trip time, so that the collapse of a stability margin under a lengthening dead time becomes a measurement instead of an assertion.

The chapter is organized as follows. Section 7.2 recalls how congestion control and active queue management became feedback problems. Section 7.3 states the two-state delay differential model, its state-dependent delay, the operating point and every constant, completely enough for an independent reimplementation. Section 7.4 linearizes the model and solves the characteristic equation of the loop for a critical round trip of 297 ms. That prediction is then bracketed against the round trip at which the simulator itself stops settling. Section 7.5 describes the PI queue controller, which is the established design of the AQM literature and not a new one, and Section 7.6 gives the numerical machinery that advances a delay equation. Section 7.7 lists the ten scenarios and the graded challenge, arranged so that one mechanism is isolated at a time, and Section 7.8 reports the measured runs. Section 7.9 describes the devices that measure and mark a real queue. Sections 7.10 and 7.11 state the limitations of the model and its place in practice, Section 7.12 summarizes the results, and the exercises follow in Section 7.13.

7.2 Background

Congestion collapse was observed on the internet before any congestion controller existed, and the history box that follows tells how the first controller answered it [1]. The remedy was a control law. A sender increases its window by one packet per round trip while nothing is lost and halves it when a loss is detected, which is the additive-increase and multiplicative-decrease rule whose convergence to a fair and efficient allocation was analyzed by Chiu and Jain [2].

> ⧖ HISTORY **Congestion Collapse, 1984 to 1988**
>
> A technical note of January 1984 described congestion collapse. A sudden load can raise the round trip faster than the hosts can update their estimates of it, the hosts then send several copies of the same packets, the buffers of the switching nodes fill, and the condition is stable [12]. In October 1986 the internet had the first of a series of such collapses: throughput between two sites 400 yards apart fell from 32 kbit/s to 40 bit/s. The repair, published in 1988, made the sender's window a control law. The window is halved on congestion and grows by a small constant step otherwise, additive increase and multiplicative decrease. Its authors argued that a network is, to a very good approximation, a linear system built of integrators, delays and gain stages [1].

That first generation of congestion control placed the whole loop in the end hosts and left the router with a single signal, buffer overflow. A drop-tail router gives no indication of congestion until its buffer is full, by which time every packet in it is delayed by the full buffer drain time and every flow sharing the link receives the same synchronized loss. Random early detection was proposed to move the decision into the router, dropping or marking arriving packets with a probability computed

from an averaged queue length [13]. The Internet Architecture Board recommended active queue management for deployment on that basis in 1998, and named the determination of good parameter settings as an open problem [14]. It remained one, because RED was specified as a static mapping from average queue to drop probability rather than as a controller, and its parameters are not the quantities a control designer would choose.

The analysis that reframed the problem came from two directions at once. The optimization view of Kelly, Maulloo and Tan interpreted the congestion signal as a shadow price and the sender's rate law as a gradient step, which established what a network of such loops converges to and made the stability of the dynamics a separate question [15]. The fluid-flow view of Misra, Gong and Towsley wrote the coupled stochastic differential equations for a network of AQM routers and their TCP flows and reduced them to a small set of ordinary equations with delay [6]. Hollot and colleagues linearized that model, showed that RED is a low-pass filter followed by a proportional gain, and demonstrated that its published parameter guidance gives a loop whose stability depends on load and round-trip time in ways the guidance does not mention [7]. The same authors then designed a proportional-integral queue controller for the linearized plant and showed it holds the queue at a target with faster response than RED [8, 9]. The controller of this chapter is that one.

> ⧖ HISTORY **A PI Queue Controller in the Cable Modem, 2013**
>
> The PI queue controller left the research papers for the edge of the network. A design named Proportional Integral controller Enhanced (PIE) applies the classical PI controller, studied earlier for queue length, to the queueing delay, and its parameters are chosen by a stability analysis of the feedback loop [16]. Version 3.1 of the cable data service specification, first published in October 2013, requires cable modems to run a variant of it on their upstream queues. That technology gives broadband access to over one hundred million users [17]. Both designs were published as Requests for Comments in February 2017.

The consequences reached the public as latency rather than as loss. Cheap memory allowed edge devices to be built with buffers holding seconds of traffic, and a full buffer converts spare capacity into delay without reducing throughput, so every conventional measurement of a link reports that it is working while interactive use degrades. Gettys and Nichols named the condition bufferbloat and traced it through home routers, cable modems and operating system stacks [18]. The deployed answers are queue managers that target delay directly rather than occupancy [19]. Every one of them is a feedback controller operating around one round trip of dead time, so the phase margin argument of this chapter applies to each of them regardless of the marking rule.

7.3 Network Model

A single bottleneck router carries N long-lived TCP sessions from a group of senders to a group of receivers. The router transmits C packets per second, holds up to B packets while waiting, and marks or drops a fraction p of arriving packets to signal congestion. A marked packet is noticed by the receiver, reported in an acknowledgment and acted on by the sender, so a marking decision reaches the traffic one round trip after it is made. The sending hosts, the bottleneck, the marking block, the return path and every symbol of the model appear in Figure 7.1. The instantaneous queue length q is the controlled variable and the marking probability p is the manipulated variable.

7.3.1 Variables and Signal Classification

Table 7.1 classifies the quantities the operator and the controller see. Only two of them are integrated states. The round-trip time, the arrival rate, the equilibrium window and the loop gain are all algebraic functions of those two states and the parameters, and all of them are reported as outputs.

The queue setpoint is a latency setting as well as a control target. A queue of q packets on a link of C packets per second delays every packet that joins it by q/C seconds, so the setpoint is a latency

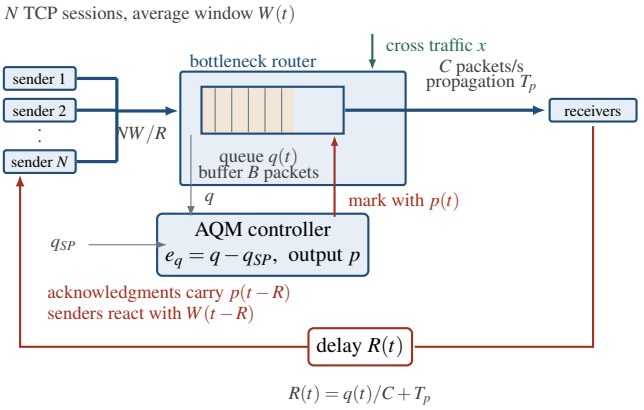


Figure 7.1 Congested path modeled by the application, with the marking decision and the traffic it controls separated by one round trip. The senders offer NW/R packets per second to a router that drains at C, and the AQM computes a marking probability from the queue error. The mark reaches a sender only after the return path, so the queue responds to $p(t-R)$ and the length of that delay depends on the queue itself. The error printed in the controller block is the convention of Equation (7.10) with the action flag already applied.

Table 7.1 Signal Roles on the Queue Loop. Each Nominal Entry Belongs to the Rest State That Opens Every Scenario.

Symbol	Role	Nominal	Range
q	CV, state	100 packets	10–400 packets SP
p	MV	1.556 %	0–100 %
W	state	11.25 packets	≥ 1 packet
R	output, delay	180 ms	T_p to $B/C + T_p$
N	DV, declared	20 sessions	1–200
ΔN	DV, operator	0 sessions	-15 to $+500$
x	DV, operator	0 packets/s	0–900 packets/s
C	DV, parameter	1250 packets/s	300–4000 packets/s
T_p	DV, parameter	100 ms	20–1000 ms
B	parameter	400 packets	60–1200 packets

budget written in a unit of storage. At the default settings the 100-packet target is 80 ms of queueing delay added to every packet of every session.

> **DID YOU KNOW? Five Milliseconds of Standing Queue**
>
> A delay-based queue manager published in 2018 sets its target to 5 ms of persistent queueing delay for normal internet traffic, the low end of a best range of 5 to 10 % of the round trip, and configures itself from a round-trip metric of 100 ms for the terrestrial internet [20]. The 100-packet target of this chapter adds 80 ms.

7.3.2 Window and Queue Equations

The average congestion window carried by the N sessions obeys the additive-increase and multiplicative-decrease rule of TCP Reno written as a rate, as shown in Equation (7.1).

$$\dot{W}(t) = \frac{1 - p(t-R)}{R(t)} - \frac{W(t)}{2}\,\frac{W(t-R)}{R(t-R)}\,p(t-R) \tag{7.1}$$

where W is the average window in packets, R is the round-trip time in seconds and p is the marking probability. The first term is additive increase, one packet per round trip, applied only for acknowledgments that were not marked. The second term is multiplicative decrease, in which marked acknowledgments arrive at $W(t-R)p(t-R)/R(t-R)$ per second and each one halves the window. Because W is a count of packets and is therefore dimensionless, both terms carry units of 1/s and the equation is dimensionally consistent. Retaining the $(1-p)$ factor is a small departure from the original fluid model [6], and its only visible consequence is stated in Section 7.3.4.

Every term carrying the marking probability in (7.1) is evaluated at $t-R$. That is the whole of the dead time in this application, and it is where the delay differential equation comes from. The packets whose acknowledgments are arriving now passed the router one round trip ago, so they were marked or not marked according to a decision the router has since revised hundreds of times.

The router queue is a mass balance on packets, as shown in Equation (7.2).

$$\dot{q}(t) = -C + \frac{NW(t)}{R(t)} + x, \qquad 0 \le q \le B \tag{7.2}$$

where C is the link capacity in packets per second, x is unresponsive cross traffic in packets per second and B is the buffer in packets. Each session holds W packets in flight over R seconds, so N of them deliver NW/R packets per second and the link removes C of them. Both sides carry units of packets per second. The two bounds on q are hard events and are applied after each integration step rather than inside the right-hand side, which is discussed in Section 7.6.

7.3.3 State-Dependent Round-Trip Time

The round-trip time is the propagation delay of the path plus the time a packet spends waiting in the bottleneck queue, as shown in Equation (7.3).

$$R(t) = \frac{q(t)}{C} + T_p \tag{7.3}$$

where T_p is the propagation delay in seconds. Equation (7.3) is the single line that sets this plant apart from a process with a fixed transport lag. The delay that limits the loop is computed from the state the loop is regulating. Raising the queue setpoint from 100 to 250 packets on the default link raises the round trip from 180 to 300 ms, which is enough on its own to destabilize the default tuning. A control problem in which the setpoint changes the plant is unusual, and it is the reason the modern answer to bufferbloat is to specify the target in units of time rather than in units of storage [19]. Volume 7, Chapter 3 treats a mechanical plant that its setpoint chooses in the same sense.

The propagation delay is floored at 20 ms. That is a numerical constraint rather than a physical one and its reason is given in Section 7.6. The window is floored at one packet, which is physical, because TCP always keeps one segment in flight.

Table 7.2 Model and Controller Parameters. All Are Visible and All Are Adjustable While a Run Is in Progress.

Symbol	Meaning	Value	Unit
Plant			
C	bottleneck link capacity	1250	packets/s
N	declared TCP sessions	20	sessions
T_p	propagation delay	0.100	s
B	router buffer	400	packets
L	packet size	1500	bytes
$W_{\min}$	minimum window	1	packet
Controller, default tuning			
K_c	proportional gain	3.5×10^{-4}	1/packet
τ_I	integral time	5.0	s
τ_D	derivative time	0	s
τ_f	derivative filter	0	s
p_0	bias, from (7.4)	0.015557	—
Numerics			
Δt	integration step	0.005	s
Δt_c	controller period	0.005	s
—	historian rate	20	samples/s
—	delay-history capacity	2048	entries
—	seeded history span	12	s

7.3.4 Operating Point

Requiring both derivatives to vanish pins down the rest state of the network, as given by Equation (7.4).

$$R_0 = \frac{q_{SP}}{C} + T_p, \quad W_0 = \frac{R_0 (C - x)}{N}, \quad p_0 = \frac{2}{2 + W_0^2} \tag{7.4}$$

where q_{SP} is the queue setpoint. The queue balance fixes the window the link can support and the window equation then fixes the loss rate that sustains it. At the default settings this is $R_0 = 180\,\text{ms}$, $W_0 = 11.25$ packets and $p_0 = 1.5557\,\%$. Dropping the $(1 - p)$ factor from (7.1) would give $p_0 = 2/W_0^2 = 1.5802\,\%$ instead, a difference of $1.6\,\%$, and the exact form is the one the application's first verification test checks by evaluating both derivatives at (7.4) and requiring them to vanish to within 10^{-9}.

Equation (7.4) is also a feasibility test. If W_0 comes out below one packet the operating point does not exist, because no marking probability can make a TCP session send less than one segment per round trip. Five hundred sessions on the default link need $W_0 = 0.45$ packets, and the application reports the requested operating point as infeasible.

7.3.5 Parameters

Table 7.2 lists every constant. Nothing is hidden from the user in this application, because the lesson is structural rather than one of identification. The link speed is the one quantity that is not entered directly: $C = 1250$ packets per second at $L = 1500$ bytes per packet is $15.0\,\text{Mbit/s}$.

The proportional gain is entered as $3.50\,\%$ per 100 packets and stored as 3.5×10^{-4} per packet. The exported column `ctrl_Kc_per_packet` carries the stored value, so the two representations differ by a factor of 10^4 and must not be interchanged.

7.3.6 Assumptions

The two-state description rests on several assumptions, and each one fails in a specific place.

Fluid approximation. W and q are continuous averages over many packets and many sessions. The model has no notion of an individual packet, of burstiness inside a round trip or of the discrete nature

of a drop. It applies for N of order ten or more and does not apply to a single connection, where the sawtooth of one window dominates the average.

One congestion control algorithm. Windows grow by one packet per unmarked round trip and halve on a mark, which is TCP Reno with no slow start and no retransmission timeouts. CUBIC, BBR and Vegas reach different equilibrium loss rates, so (7.4) would change. The argument about dead time and phase margin would not.

One round trip for every session. Every session is assumed to share one bottleneck and to have the same propagation delay and the same average window. Real paths carry a spread of round-trip times, which acts as an additional low-pass filter and is mildly stabilizing. Real paths also carry multiple congested hops and reverse-path congestion, neither of which is represented and neither of which is stabilizing.

An ideal measurement. The AQM reads the instantaneous queue exactly, with no noise, no averaging and no sampling interval of its own, and applies its marking probability to arrivals immediately. Deployed queue managers average the queue and update on a timer, and both of those add lag to a loop that has none to spare. The delay in this application is entirely in the actuation path and none of it is in the instrument.

Window dilution when sessions join. W is an average over N sessions, so changing N changes what the state means. When the session count rises from N_{old} to N_{new} the average window is rescaled to $(N_{\text{old}}W + (N_{\text{new}} - N_{\text{old}}))/N_{\text{new}}$, which conserves the packets in flight apart from the one segment each newcomer brings. Without that correction a flash crowd would arrive with every new flow already at the incumbent window, which is an artifact of averaging rather than a property of TCP. Departing flows take their windows with them, so a falling N leaves the average alone.

No stochastic element anywhere. The physics contains no noise source, so two runs of the same scenario produce identical files. The only use of a random number generator in the application is in the display of individual packets, which does not feed back into the model. The seed of Section 1.6.1 therefore plays no part in the runs of this chapter.

7.4 Linearization and the Characteristic Equation

7.4.1 Linearized Plant

A first-order expansion of (7.1) and (7.2) around the rest state (7.4), taken with $x = 0$ and with the delay frozen at its rest value R_0, reproduces the two coupled equations the AQM literature works from [9]. These are Equations (7.5) and (7.6).

$$\delta\dot{W} = -\frac{2N}{R_0^2 C}\,\delta W - \frac{R_0 C^2}{2N^2}\,\delta p(t - R_0) \tag{7.5}$$

$$\delta\dot{q} = \frac{N}{R_0}\,\delta W - \frac{1}{R_0}\,\delta q \tag{7.6}$$

where δ denotes a deviation from the operating point. The window coefficient in (7.5) collects the sensitivity of the decrease term to both $W(t)$ and $W(t-R)$, which the linearization treats as one perturbation. The queue coefficient in (7.6) is exact, because $NW_0 = CR_0$ makes $\partial\dot{q}/\partial q$ equal to $-1/R_0$ with no approximation. Transforming and eliminating δW gives the plant the queue controller faces, as shown in Equation (7.7).

$$P(s) = \frac{\delta q(s)}{\delta p(s)} = -\frac{C^2}{2N}\,\frac{\mathrm{e}^{-sR_0}}{(s+a)(s+b)} \tag{7.7}$$

$$a = \frac{2N}{R_0^2 C}, \qquad b = \frac{1}{R_0}, \qquad |P(0)| = \frac{R_0^3 C^3}{4N^2} \tag{7.8}$$

where the two real poles are the TCP window dynamics at a and the queue at b, as shown in Equation (7.8). At the default operating point they are 0.988/s and 5.556/s, the numerator gain $C^2/2N$

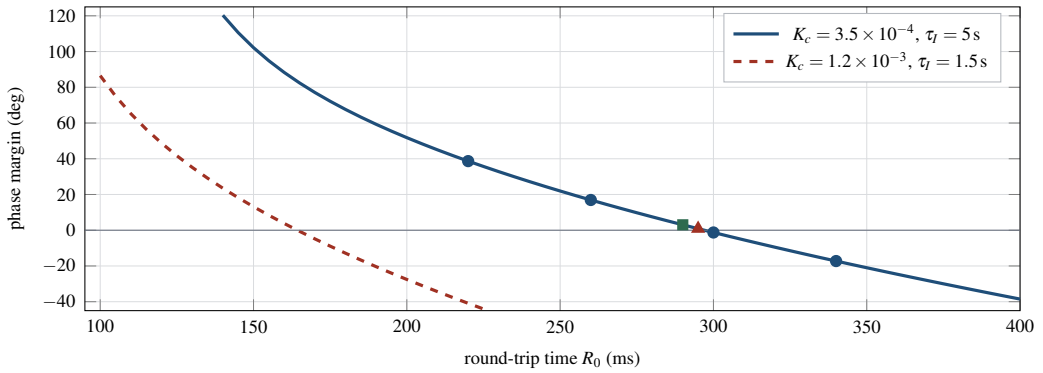


Figure 7.2 Phase margin against round-trip time computed from the characteristic equation (7.9) at the two tunings the application ships. The solid line is the default tuning and the dashed line is the faster one. Filled circles are the four operating points scenario 4 visits. On the 5 ms grid of the propagation-delay setting, the square at 290 ms is the longest round trip at which the simulator settles and the triangle at 295 ms is the shortest at which it sustains an oscillation, against a predicted crossing of the zero line at 297 ms.

is 3.906×10^4 packets per second squared per unit marking probability, and the DC gain is 7119 packets per unit marking probability. Retaining the $(1 - p)$ factor of (7.1) in the linearization moves the DC gain to 7346 packets, a difference of 3.2 %, and the application logs the simpler expression of (7.7) in the column `loop_dc_gain_packets`.

Three features of (7.7) and (7.8) matter more than the pole locations. The gain is negative, so a queue above target must be met with more marking rather than less. The gain varies as the cube of R_0, so a round trip twice as long multiplies it by eight. The gain varies as the inverse square of N, so the loop gain is highest on a lightly loaded path rather than on a busy one.

7.4.2 Phase Margin Against Round-Trip Time

Closing (7.7) with the PI controller of Section 7.5 gives a transcendental characteristic equation, as shown in Equation (7.9).

$$\tau_I s\,(s+a)(s+b) + \frac{K_c C^2}{2N}\,(\tau_I s + 1)\,\mathrm{e}^{-sR_0} = 0 \tag{7.9}$$

where the sign inversion of the process gain has been absorbed into the controller. Equation (7.9) has infinitely many roots because of the exponential, which is the defining property of a delay differential equation [11] and the reason a root-locus construction on a polynomial cannot be used here. What can be used is the frequency response, with the phase margin as Section 1.4.12 defines it. Dead time contributes a phase lag of ωR_0 radians and attenuates nothing at all, so it removes margin at the frequencies where the loop gain is near unity.

At the default tuning and the default operating point the loop gain crosses unity at $\omega_c = 2.09\,\mathrm{rad/s}$ with a phase margin of 67.6°, of which the dead time has already taken 21.6°. Holding the tuning fixed and lengthening R_0 does two things at once. The dead time takes more phase, and the cubic DC gain pushes crossover to a higher frequency where it takes more still. The phase margin against round-trip time computed from (7.9) is shown in Figure 7.2. Between $R_0 = 180\,\mathrm{ms}$ and $R_0 = 300\,\mathrm{ms}$ the margin falls from 67.6° to −1.3°, a loss of 68.9° with no gain touched, while crossover moves from 2.09 to 3.02 rad/s.

🕮 **Pause.** Try Exercise 7.1 before reading on.

7.4.3 Critical Round-Trip Time and Its Measurement

Solving (7.9) for the round trip at which the phase margin reaches zero gives $R_{0,\mathrm{crit}} = 297\,\mathrm{ms}$ at the default tuning, with a crossover of 3.01 rad/s and therefore an expected oscillation period of 2.09 s. Retaining the $(1-p)$ factor moves the prediction to 296 ms.

That prediction is testable inside the application. Starting from the nominal network at rest, stepping the propagation delay at $t = 5\,\mathrm{s}$ and running to $t = 100\,\mathrm{s}$ gives a queue transient whose peak-to-peak excursion either decays or does not. At $R_0 = 285\,\mathrm{ms}$ the excursion falls from 15.2 packets in the window $t = 25$–$40\,\mathrm{s}$ to 0.37 packets in $t = 85$–$100\,\mathrm{s}$, and at 290 ms it still falls, from 29.1 to 8.8 packets over the same two windows. At $R_0 = 295\,\mathrm{ms}$ it holds between 37.4 and 37.6 packets in every 15 s window from $t = 25\,\mathrm{s}$ onward and does not decay at all. The propagation delay is set in steps of 5 ms, so the finest bracket the application allows places the measured boundary between 290 and 295 ms, against a predicted 297 ms, and the measured period of the sustained oscillation at 295 ms is 2.07 s against a predicted 2.09 s. The remaining disagreement is in the direction the approximations predict, because the linearization holds the delay fixed at R_0 while the simulator lets it move with the queue.

The same calculation gives the ultimate gain of Section 1.4.7. Holding $R_0 = 180\,\mathrm{ms}$ and $\tau_I = 5\,\mathrm{s}$ and raising the proportional gain drives the phase margin to zero at $K_c = 1.06 \times 10^{-3}$ per packet, which is 10.6 % per 100 packets in the entered units. The default gain of 3.5×10^{-4} is a factor of 3.0 below that.

7.5 Controller Design

7.5.1 Discrete Algorithm

One loop is closed, and it runs the velocity-form PID algorithm of Section 1.6.2, with the derivative taken on the measurement [21]. Equations (7.10) to (7.14) restate that algorithm in the symbols of the queue loop, because the results of Section 7.8 cite them. They are Equations (1.23) to (1.26) with $s_k = S_k$, $c_k = v_k$ and $\tau_t = T_t$.

$$e_k = a\,(q_{SP} - q_k), \qquad a = -1 \tag{7.10}$$

$$\Delta p_k^P = K_c\,(e_k - e_{k-1}) \tag{7.11}$$

$$\Delta p_k^I = \frac{K_c}{\tau_I}\,e_k\,\Delta t_c \tag{7.12}$$

$$\Delta p_k^D = -\,K_c\,\tau_D\,a\,\frac{q_k^f - 2q_{k-1}^f + q_{k-2}^f}{\Delta t_c} \tag{7.13}$$

$$\begin{aligned} s_k &= s_{k-1} + \Delta p_k^P + \Delta p_k^D \\ c_k &= p_0 + s_k + I_{k-1} \\ p_k &= \min(1,\ \max(0,\ c_k)) \\ I_k &= I_{k-1} + \Delta p_k^I + \frac{\Delta t_c}{\tau_t}\,(p_k - c_k) \end{aligned} \tag{7.14}$$

where a is the controller action flag, p_0 is the output bias of (7.4), s_k accumulates the proportional and derivative increments without limit, c_k is the marking probability the algorithm demands, p_k is that demand clipped to the physical range of a probability, I_k carries the integral contribution and $\tau_t = \tau_I$ is the tracking time of the back-calculation. Differentiation uses a backward difference on a filtered queue, $q_k^f = \alpha q_{k-1}^f + (1-\alpha) q_k$ with $\alpha = \exp(-\Delta t_c/\tau_f)$. That coefficient comes from an exponential and cannot go negative, so the filter is stable at whatever period it is run.

The action flag is the entire expression of the negative process gain in (7.7). Setting $a = -1$ makes the error $q_k - q_{SP}$, so a queue above target produces more marking. The controller output therefore rises as the measurement rises, which is a direct-acting controller in the convention of Section 1.4.3 [3, 21]. The sign is checked by one of the application's own verification tests.

🕮 **Pause.** Try Exercise 7.2 before reading on.

Anti-windup is the back-calculation term of Equation (7.14), which corrects the integral alone for the reasons Section 1.4.9 gives [21]. At the default $\tau_I = 5\,\mathrm{s}$ each execution removes $1/1000$ of the clipped excess, and with $\tau_t = \tau_I$ a loop held against a limit settles where $c_k - p_k = K_c e_k$. Away from the limits the recursion is algebraically identical to the positional law with the same three gains, so the tuning constants and the linear analysis of Section 7.4 carry over. Against a limit it is not, and the saturated runs of Section 7.8 are those of the back-calculation as coded.

Windup protection is not optional on this loop. The marking probability is bounded to $[0, 1]$ by its own definition, and both bounds are reached in ordinary operation, $p = 0$ whenever the queue sits well below target and $p = 1$ whenever more sessions are present than the link can serve at one packet per round trip each. No slew limit is applied, because a router can change its marking probability between one packet and the next. The controller executes every 5 ms, the same period as the physics step, which is 2.8 % of the default round trip.

7.5.2 Output Bias and Manual Operation

The output bias is p_0 from (7.4), evaluated at the declared session count and the declared capacity and recomputed on every controller execution. It is a feedforward term computed from the model rather than a constant. It places the marking probability at the value the target operating point requires on the first execution, and it moves with the queue setpoint, so a setpoint change of 100 to 160 packets steps the marking probability toward its new equilibrium immediately rather than waiting for the integral. The disturbances are excluded from it deliberately. Extra sessions and cross traffic do not enter the bias calculation, so the mismatch between the declared network and the real one is left entirely to the integral term.

Manual operation tracks the operator-set marking probability, and automatic operation copies the applied marking probability to the manual setting after every controller execution, so transfer is bumpless in both directions as Section 1.6.2 describes. The manual actuator is the marking probability itself, entered over 0 to 100 % at a resolution of 0.05 %. An equilibrium helper applies p_0 computed for the traffic actually present, and it reports the operating point as unreachable when $W_0 < 1$.

7.5.3 Derivative Action and Drop-Tail Operation

Derivative time is adjustable from 0 to 2 s and defaults to zero. Derivative action on a process whose dead time dominates its lag differentiates a queue slope that the senders committed to one round trip ago and can no longer change, so the prediction it forms is of an interval that is already determined. Raising τ_D is an instructive experiment rather than a recommended tuning. The derivative filter τ_f is relevant only when τ_D is nonzero.

The active queue manager can be switched off. The router then marks at the fixed nominal rate p_0 and the queue finds its own level, which is the drop-tail behavior that motivated active queue management in the first place [13]. The idle PI block keeps tracking the applied value while it is off, so switching it back on during a run produces no step.

7.6 Numerical Solution of the Delay Equations

A delay differential equation needs two things an ordinary initial value problem does not: a history of the solution reaching back at least one delay, and an initial function rather than an initial point.

The history is a ring buffer of 2048 entries, each holding the four numbers $[W, q, R, p]$ at one completed physics step. At 5 ms per step that is 10.24 s of stored past, which exceeds the longest reachable round trip of $B/C + T_p \le 5\,\mathrm{s}$. Reading the history at an arbitrary time performs a binary search for the bracketing pair of entries and interpolates linearly between them, and clamps to the oldest or newest stored value outside the range. The buffer is extended by exactly one entry per completed step, stamped at the new simulation time and holding the values after the hard constraints have been applied, so the value the senders react to is the value the router applied.

The initial function is a constant. At every reset the history is seeded with two identical entries, one stamped at $t = -12\,\mathrm{s}$ and one at $t = 0$, both carrying the initial state and the equilibrium marking probability. Linear interpolation between two equal values is a constant over the whole span, and anything older clamps to the same value, so the run opens from a network that has held that condition over the whole seeded span. This is the correct initial condition for (7.1) and it is also what makes a rerun with the same settings produce an identical file, because the controller object and its history are rebuilt from scratch at every reset.

Integration uses the classical fourth-order Runge-Kutta method at a fixed step of 5 ms, the framework default of Section 1.6.1. That choice requires justification on a delay equation, because a Runge-Kutta stage evaluates the right-hand side at $t + \Delta t$ and therefore asks the history for the state at $t + \Delta t - R$, which would be an extrapolation if R were smaller than Δt. The propagation delay floor of 20 ms removes that possibility. It guarantees $R \geq 4\Delta t$ even with an empty queue, so the latest query any stage can make is $t - 3\Delta t$, which lands strictly inside the stored history and is served by ordinary interpolation. Nothing is extrapolated. An explicit Euler step would need no such floor, because it queries the history only at the start of the step, and the shared framework selects Euler for applications whose right-hand side is discontinuous. This application keeps the fourth-order method by moving its discontinuities out of the derivative function. The queue bounds of (7.2) and the one-packet window floor are applied between steps rather than inside them, which leaves the right-hand side smooth across every stage.

Two approximations remain, and both are documented in the application. The delay is resolved to within one step, which is 2.8 % of the round trip at the 180 ms default, about 5 % in the short round-trip scenario and at worst 25 % in the corner where T_p is set to its 20 ms minimum and the queue is held empty. Applying the bounds at step boundaries makes the first step after a boundary contact first-order accurate rather than fourth-order, which matters only at an empty queue and at the buffer limit.

Of the three clocks of Section 1.6.1, the physics and the controller both advance at 5 ms, an unusually fast controller relative to the process and one chosen so that no sampling artifact confuses the delay argument, while the log takes 20 rows per simulated second. Six checks are built into the application itself. They cover the rest condition of (7.4), the sign of the queue response to marking, the sign of the queue response to added sessions, the sign of the controller, the interpolation and clamping of the delay buffer against a known ramp, and the assertion that a step in marking moves the queue one round trip later rather than sooner.

7.7 Scenarios

Every scenario is built on the same nominal network of 20 sessions, 1250 packets per second, 100 ms of propagation delay and a 100-packet target, and is arranged so that each run isolates a single mechanism. Table 7.3 lists them. Two tunings recur. The default $K_c = 3.5 \times 10^{-4}$ with $\tau_I = 5\,\mathrm{s}$ was designed at the nominal 180 ms round trip. The alternative $K_c = 1.2 \times 10^{-3}$ with $\tau_I = 1.5\,\mathrm{s}$ was designed by the same procedure at a 100 ms round trip, where the plant is far less sensitive and the gain can be three times larger. Scenario 8 keeps the default gain and shortens the integral time to 1.5 s. Table 7.4 gives each scenario under the name the application shows, with the requirement for completing it under the rules of Section 1.6.4. The challenge is judged on the case as given, so changing a model parameter or a disturbance voids the attempt, and the footnote of the table gives its outcome under the shipped and the suggested tuning.

Two operator-set disturbances and four timed events are available in every scenario. The disturbances are extra TCP sessions, from -15 to $+500$ in steps of 5, and unresponsive cross traffic in packets per second, which consumes capacity and ignores marking entirely. The events are a flash crowd of 40 sessions, the departure of every added session, a halving of the link capacity and a route change adding 120 ms of propagation delay.

Table 7.3 Scenarios and the Mechanism Each One Exercises

#	Change applied	Mechanism exposed
1	none, network at rest	an equilibrium that needs no controller
2	N 20 to 40 at $t = 6$ s	operating point moves, CV does not
3	C 1250 to 600 packets/s at $t = 6$ s	same queue, longer latency
4	T_p to 140, 180, 220, 260 ms at $t = 8, 30, 52, 74$ s	the stability boundary at fixed tuning
5	+40 sessions at $t = 6$ s	a peak affected by dead time and tuning
6	q_{SP} 100 to 160 to 40 packets	the setpoint is the plant and the latency
7	$T_p = 20$ ms, fast tuning, route change to 160 ms at $t = 16$ s	a correct design commissioned elsewhere
8	+480 sessions at $t = 5$ s, $\tau_I = 1.5$ s	no feasible operating point, windup
9	$B = 140$, $q_{SP} = 60$, +30 sessions	overshoot larger than headroom
10	AQM off, +40 sessions at $t = 6$ s	bufferbloat with the loop open
C	$q_{SP} = 120$, aggressive tuning, crowd arrives and leaves	gain scheduling against session count

Two of the scenarios are deliberately extreme. Five hundred sessions arriving at once on a 15 Mbit/s link would not occur in this form on a physical network, and the scenario is there to show how an operating point that does not exist appears from inside the loop. A propagation delay stepped four times in 100 s is equally artificial, and it is the only arrangement that holds everything else constant while the delay moves. The scripted steps of scenario 4 overwrite any earlier setting of the propagation delay. The boundary is therefore explored by hand either after the last scripted step at $t = 74$ s or on the undisturbed network of scenario 1, as in Section 7.4.

7.8 Results

Table 7.5 collects the scorecard values at the stated simulated time in each case. The trajectories behind those numbers were taken from the 40-column comma-separated export at 20 samples per simulated second. The integrated measures accumulate over the whole run, while the last two columns are instantaneous readings, so for the three runs that end in a limit cycle (4, 7 and the shipped challenge tuning) those two columns depend on the phase of the cycle at the reading instant. The saturated fraction of scenario 4 is exactly 347 of its 2000 logged samples, so it is given to two decimals rather than rounded either way. Figure 7.3 draws three of its columns.

7.8.1 Dead Time Under a Flash Crowd

Scenario 5 makes the dead time visible in a single trace. Forty sessions open at $t = 6.00$ s and the controller responds from its next execution at 6.005 s, raising the marking probability from 1.56 to 2.99 % within 0.2 s. The column `mark_probability_delayed`, which is $p(t - R)$ and therefore the value the senders are reacting to, is still 1.556 % at $t = 6.20$ s and first moves between 6.20 and 6.25 s, one round trip after the event. No marking decision made in that interval had reached a sender. The queue peaks at 226.8 packets at $t = 7.18$ s at the 5 ms step, and the initial buildup is forced by the disturbance and the feedback delay. Tuning still changes the eventual peak and decides whether the recovery is monotone. The delay does not make that peak independent of the gain.

The steady state that follows is the one (7.4) predicts. Sixty sessions sharing the same link settle at $W_0 = 3.75$ packets, which is sustainable only at a marking probability of 12.45 %, and the integral term raises the output from 1.56 toward that value while the queue returns to target.

7.8.2 Stability Boundary Under Stepped Propagation Delay

Scenario 4 holds the tuning, the traffic, the capacity and the buffer fixed and raises the propagation delay in four steps. The round trip at the 100-packet target is $R_0 = q_{SP}/C + T_p$, which gives 220, 260,

300 and 340 ms. The behavior at each is summarized in Table 7.6 and is shown in Figure 7.4. The first two steps ring and recover, the third does not, and the transition falls where (7.9) places it.

The decay of the ringing tracks the predicted margin. At $R_0 = 220$ ms the successive queue peaks above target are 19.4, 5.3, 1.9 and 1.0 packets, a first decay ratio of 0.27. At $R_0 = 260$ ms they are 23.0, 13.0, 7.4, 4.2, 2.4 and 1.4 packets, a ratio of 0.57 held on every cycle and a longer settling. At $R_0 = 300$ ms the ratio reaches unity and the queue holds a 40-packet swing with the marking probability against its lower limit 22 % of the time. Nothing about the controller changed at any of those steps. The oscillation period is 2.0 s at 260 ms and 2.11 s at 300 ms, against the 2.09 s that ω_c at the boundary predicts.

Table 7.4 Scenarios and Completion Requirements

#	Scenario	Kind	Required to complete
1	Nominal network	Experiment	Hold Auto for 10 s with no added sessions, then switch to Manual, add 40 sessions and hold the queue within ±25 packets of 100 for 5 s by setting the marking probability by hand.
2	Twice the TCP sessions	Experiment	Run in Auto through the step from 20 to 40 sessions at $t = 6$ s until the queue has held within 3 packets of target for 5 s.
3	Link capacity cut in half	Experiment	Run in Auto through the cut to 600 packets/s at $t = 6$ s until the queue has held within 3 packets of 100 for 5 s.
4	Propagation delay, step by step	Experiment	Run the full 100 s in Auto through the steps of T_p to 140, 180, 220 and 260 ms at $t = 8$, 30, 52 and 74 s.
5	Sudden traffic arrival (flash crowd)	Experiment	Run in Auto past the 40-session arrival at $t = 6$ s to at least $t = 10$ s, then Reset, switch to Manual before the arrival, move the marking probability and stay in Manual for 12 s ending by $t = 18$ s.
6	Moving the queue target	Experiment	Run in Auto through the target steps to 160 and 40 packets at $t = 8$ and 35 s to $t = 50$ s, then Reset and, after the step to 160 packets, hold a 250-packet target for 20 s.
7	Tuned for a short RTT, run at a long one	Demonstration	Run in Auto with the fast tuning through the route change at $t = 16$ s until link utilization falls below 95 %, and on to $t = 60$ s, then Reset and hold $K_c = 3.5$ % per 100 packets and $\tau_I = 5$ s from before the route change for 45 s.
8	Packet marking saturated at 100 %	Demonstration	Run in Auto past the 480-session arrival at $t = 5$ s until the marking probability has held at 100 % for 5 s, then remove the extra sessions and run 5 s with the marking probability off its upper limit.
9	Small router buffer	Demonstration	Run in Auto to $t = 40$ s with the shipped tuning and more than 0.05 s of buffer overflow, then after Reset retune K_c or τ_I before $t = 6$ s and run to 40 s with no overflow, then after a second Reset hold a 30-packet target from before $t = 6$ s to 40 s.
10	Controller off (tail drop only)	Demonstration	With the AQM off, run in Auto until the queue has held at the 400-packet buffer limit for 3 s, then switch the PI controller on and run until the queue has held within 5 packets of 100 for 3 s.
C	Challenge, ride out a flash crowd	Challenge	With the case as given and q_{SP} left at 120 packets, run 75 s with the queue within ±25 packets of target from $t = 49.5$ s, mean link utilization of at least 99 % and no more than 0.05 s of buffer overflow.[a]

[a] The shipped tuning, $K_c = 15$ % per 100 packets and $\tau_I = 1.2$ s, fails with a worst queue error of 46 packets over the judged window, 21 packets outside the band, at 100.0 % mean utilization and no overflow. The suggested tuning, $K_c = 3.5$ % per 100 packets and $\tau_I = 5$ s, passes with a worst error of 9 packets, a margin of 16 packets, at 100.0 % mean utilization and no overflow, whether it is entered before Run or carried through Reset.

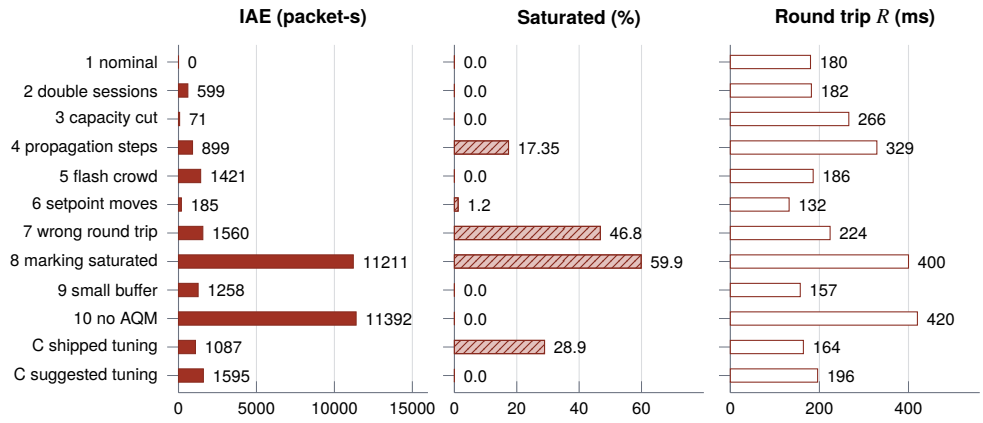


Figure 7.3 Three columns of Table 7.5 at the end of each run: integrated absolute error (solid bars), the share of the run spent against a marking limit (hatched) and the round trip at the reading instant (open). Notice that scenarios 8 and 10 carry almost the same error from opposite causes, one with the marking probability saturated for most of the run and the other with no marking change at all, and that the drop-tail router of scenario 10 holds the longest round trip on the chart.

Table 7.5 Scorecard Values Read at the End of Each Run. IAE Is in Packet-Seconds and Marking Travel Is $\sum_k |p_k - p_{k-1}|$.

Scenario	t (s)	IAE	travel	sat. (%)	$\lvert e_q \rvert$	R (ms)
1 nominal	25	0	0.00	0.0	0	180
2 double sessions	40	599	0.05	0.0	2	182
3 capacity cut	40	71	0.05	0.0	0	266
4 propagation steps	100	899	0.64	17.35	13	329
5 flash crowd	45	1421	0.10	0.0	8	186
6 setpoint moves	60	185	0.16	1.2	0	132
7 wrong round trip	60	1560	1.81	46.8	50	224
8 marking saturated	45	11211	0.98	59.9	275	400
9 small buffer	40	1258	0.09	0.0	12	157
10 no AQM	45	11392	0.00	0.0	300	420
C shipped tuning	75	1087	3.60	28.9	39	164
C suggested tuning	75	1595	0.13	0.0	0	196

Table 7.6 Scenario 4. Predicted Margin and Measured Queue Swing at Each Propagation-Delay Step. The Swing and the Saturated Fraction Are Both Taken Over the Second Half of Each Segment, After the Transient.

T_p (ms)	R_0 (ms)	$\lvert P(0) \rvert$	PM (deg)	swing (pkt)	sat. (%)
140	220	12 998	38.7	0.3	0
180	260	21 455	16.9	2.0	0
220	300	32 959	−1.3	40.4	22
260	340	47 979	−17.3	54.4	42

TRY IT IN THE APP **4 · Propagation Delay, Step by Step**

Load the case and press Run. Watch the router queue trend through the four steps of the propagation delay. After the first two steps the queue rings and settles. After the third, at a round trip of 300 ms, it holds a swing of about 40 packets, although nothing about the controller has changed.

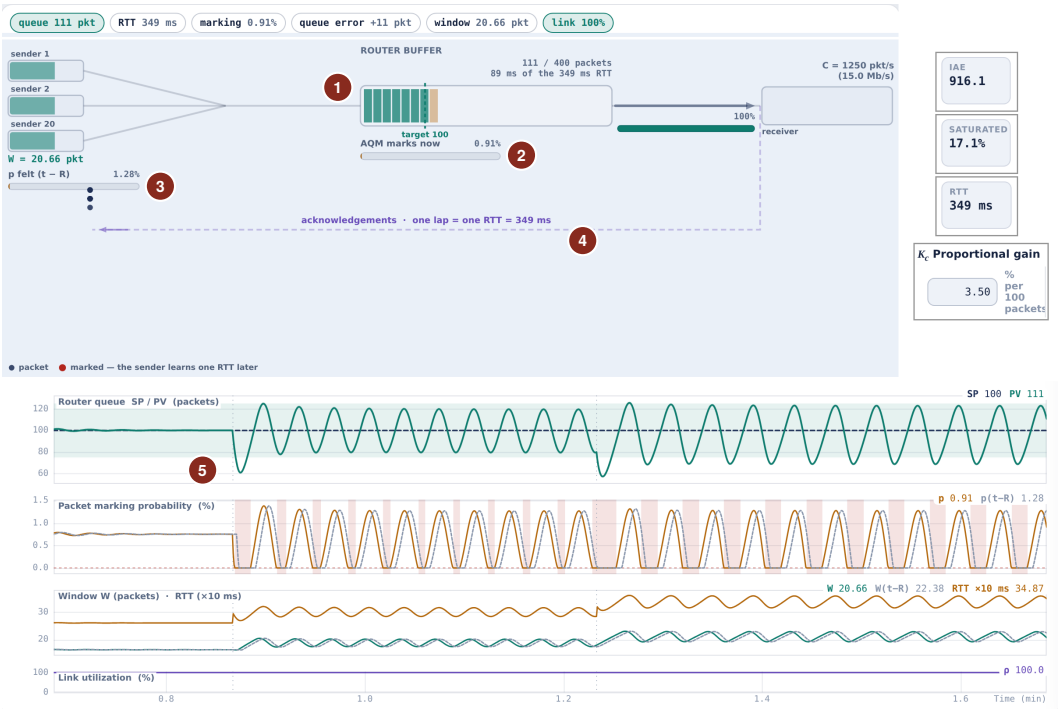


Figure 7.4 Crossing a stability boundary with nothing but the propagation delay. The screenshot was taken at $t = 101.2$ s of scenario 4, so its 60-second trend window opens on the settled tail of the 260 ms round trip, contains the step to 300 ms at $t = 52$ s and the step to 340 ms at $t = 74$ s. The queue strip changes from flat to a sustained swing at the first of those steps and to a larger one at the second. The marking strip shows the shaded bands where the probability sits at zero, which is 17.1 % of the run, and the window strip carries W, the same window one round trip earlier and the round-trip time itself, so the dead time appears as a horizontal displacement between two curves of the same shape. The scorecard reads IAE 916.1 packet-seconds and a round trip of 349 ms at this instant, and the proportional-gain field still holds the $K_c = 3.50$ % per 100 packets it started with. The figure is a composite: the readout chips and the plant view above the trend strips, with three scorecard tiles and the proportional-gain field moved into gray frames beside the plant view. Callouts: (1) the router queue against its 100-packet target, (2) the marking probability the queue manager applies now, (3) the marking the senders feel, $p(t - R)$, (4) the acknowledgment path, one lap of which is one round trip, and (5) the queue strip at the step to 300 ms.

7.8.3 A Tuning Commissioned at a Shorter Round Trip

Scenario 7 starts on a 20 ms path with the faster tuning, which has a phase margin of 86.5° at the 100 ms round trip it was designed for, and handles a setpoint move from 100 to 130 packets with a 48.9-packet excursion and no sustained motion. At $t = 16$ s the route changes and the propagation delay becomes 160 ms. The critical round trip for that tuning is 164 ms by (7.9), and the new operating point is 264 ms, so the predicted margin is −67°. The queue swings from empty to 159.4 packets in the first cycles after the route change and then settles into a limit cycle between 60.9 and 156.1 packets with a period of 2.23 s. The marking probability sits against a limit for 46.8 % of the run. The queue empties only once, from $t = 16.26$ to 16.64 s at the bottom of the first swing, and the link idles there to 76.2 % utilization at the 5 ms step, a dip the 20-sample-per-second log records as 79.0 %. Once the cycle is established the link stays fully used. The tuning is correct for the path it was designed on and incorrect for this one, and nothing about it was changed.

7.8.4 Marking Saturation and Drop-Tail Operation

Scenarios 8 and 10 produce almost the same integrated error from opposite causes. In scenario 8 five hundred sessions arrive on a link that can carry 1250 packets per second. The target operating point needs $W_0 = 0.45$ packets per session, TCP cannot go below one, and the marking probability runs to 100 % and stays there for 59.9 % of the run to $t = 45$ s. The queue then finds its own level. With every window pinned at one packet the offered load is N/R, which falls as the queue grows because the queue is inside the round trip, and the system settles where $N/R = C$, at $q = N - CT_p = 375$ packets and $R = 400$ ms. The controller has no authority left and a permanent error of 275 packets. The evidence that anti-windup is working is the pre-clamp command in `mark_probability_cmd`, which holds at 1.096, the value $1 + K_c e_k$ at which the back-calculation balances the integral increment, rather than ramping without limit. When the extra sessions are removed at $t = 45$ s the pre-clamp command falls at the first logged sample at $t = 45.05$ s, while the applied probability remains at its bound, and the marking probability leaves 100 % at the execution at $t = 45.235$ s, as soon as the draining queue crosses its target.

In scenario 10 the AQM is switched off and 40 sessions arrive. With p frozen at 1.556 % the window equation has no idea the link is now shared sixty ways, so each session returns to the same 11.25-packet window it had before. Sixty of them hold 675 packets in flight, the queue would have to reach $NW - CT_p = 550$ packets to accommodate them, and the 400-packet buffer stops it first. The queue parks against the limit for 36.3 s, the link runs at 100 % utilization throughout, and the round-trip time goes from 180 to 420 ms. With no active queue management the standing round trip is $B/C + T_p$, so the latency is set by the buffer size the operator specified and by nothing the loop does, which is the bufferbloat argument stated as an equation. Throughput and loss both report a healthy link throughout. Switching the PI controller back on mid-run returns the queue to within 25 packets of the 100-packet target after 26.4 s and within 5 packets after 53.0 s, from above and with the link at 100 % utilization throughout, whatever the time of the switch once the queue has parked. That recovery with no loss of throughput is the entire case for active queue management [14, 18].

> **TRY IT IN THE APP** **10 · Controller Off (Tail Drop Only)**
>
> Load the case and run it. The queue manager is off when 40 sessions arrive. Watch the round-trip tile climb from 180 to 420 ms while the link stays fully used. Then switch the PI controller on and watch the queue come back to its 100-packet target with no loss of throughput.

7.8.5 Load-Dependent Plant Gain in the Graded Challenge

The challenge ships with $K_c = 1.5 \times 10^{-3}$ and $\tau_I = 1.2$ s, and asks the user to hold a 120-packet target within ±25 packets through the arrival of 30 sessions at $t = 8$ s and their departure at $t = 32$ s. At the 120-packet target $R_0 = 196$ ms, and (7.9) gives that tuning a phase margin of 36.8° with 50 sessions present and −40.3° with 20, because the plant gain varies as N^{-2}. The measurement matches. With the crowd present the queue peaks at 172.5 packets, swings 60.5 packets peak to peak in the first transient and holds within 0.3 packets of target from $t = 16$ s until the crowd leaves. After the crowd leaves the swing is 70.8 packets over the last 25 seconds, with 28.9 % of the run saturated and a marking travel of 3.60. The suggested tuning gives up the peak, letting the queue reach 214.8 packets, and returns a swing of 8.4 packets over the same last 25 seconds with a travel of 0.13 and no saturation at all, although the queue empties briefly when the crowd leaves and the link dips to 93.5 % utilization at $t = 32.3$ s. The footnote of Table 7.4 gives the verdict of the completion rule on both tunings. The loop was validated under load and commissioned onto an empty network, which is the general case whenever the plant gain falls as the disturbance grows.

DID YOU KNOW? The Failing Tuning Has the Smaller Error

From the simulator: in the challenge, the shipped tuning ends with an IAE of 1087 packet-seconds and fails the completion rule, while the suggested tuning ends at 1595 and passes. The shipped tuning moves the marking probability by 3.60 in all, the suggested one by 0.13 (Tables 7.4 and 7.5).

A controller gain scheduled on the session count, in the sense of Section 1.4.13, would cancel the N^{-2} factor of the plant gain. The session count is declared and not measured in this application, so such a schedule would need an estimate of it. Chapter 6 carries a gain schedule through on a plant whose gain moves far more.

Pause. Try Exercise 7.3 before reading on.

7.8.6 Effect of the Queue Setpoint on the Dead Time

Scenario 6 raises the target to 160 packets and then drops it to 40. The queue reaches 181.6 packets and settles at 160.0 with the round trip at 228 ms, then falls to a minimum of 26.6 packets and settles near 40 with the round trip at 132 ms. Sixty extra packets of queue on this link is 48 ms of extra delay on every packet of every session, bought for no throughput at all, because the link was already fully used at 100 packets. Going the other way costs headroom instead of latency. Raising the target to 250 packets, set one step after the scripted move to 160 at $t = 8\,\mathrm{s}$ and held until the scripted move to 40 at $t = 35\,\mathrm{s}$, puts the round trip at 300 ms. Over the second half of that interval the nominal network holds a sustained 40.5-packet swing with the marking probability on its lower limit 19 % of the time, close to the 40.4-packet swing scenario 4 produces at the same round trip. Nothing about the traffic or the tuning was touched. The setpoint of this loop sets the dead time of the loop.

7.8.7 Application Access

The application is used as Section 1.6.4 describes for every framework app. The scorecard reports integral absolute error, marking travel, the fraction of the run spent against a marking limit, the queue error, the mean link utilization, the seconds of buffer overflow and the current round-trip time, and those are the quantities reported in Table 7.5. The applied marking probability is displayed together with $p(t - R)$ and the window together with $W(t - R)$, so the dead time appears as a horizontal displacement between two curves of the same shape, and the 40-column export at 20 samples per simulated second carries both pairs so that a cross-correlation recovers the round-trip time offline.

7.9 Technology in Practice

The loop of Section 7.5 reads the queue q exactly and acts through the marking probability p with no lag of its own. Figure 7.5 draws it with the parts that a congested path carries in place of the schematic's symbols.

What Measures It

The measurement sits in the router or modem that owns the bottleneck queue, and both designs of this section measure a delay rather than a count. One estimates the queueing delay from the bytes waiting and the departure rate at each update, or from timestamps [16]. Another tracks the sojourn time, the time each packet spent in the queue, because it observes the delay every packet experiences and does not depend on the link rate [20]. The model of this chapter reads the instantaneous queue in packets at every execution, which Section 7.3.6 calls an ideal measurement. Its queueing delay is q/C in Equation (7.3).

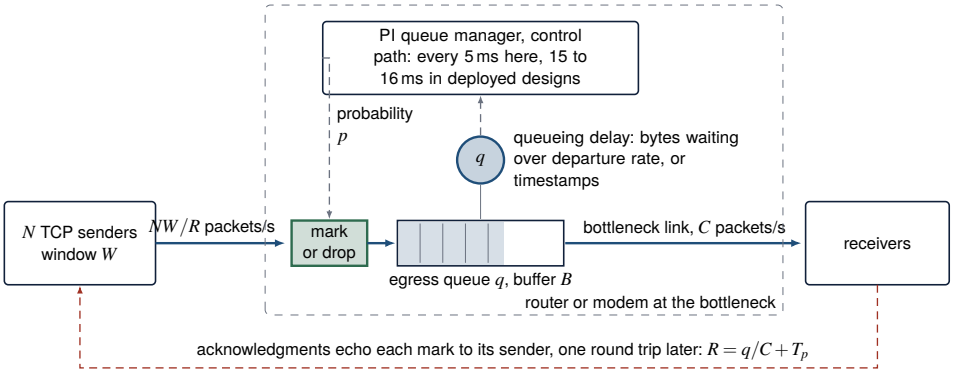


Figure 7.5 The queue loop with the parts of a real path in place of the schematic's symbols. The device that owns the bottleneck estimates its queueing delay, runs the PI law and marks or drops arriving packets, and each mark reaches its sender in an acknowledgment one round trip later. Notice that sensor, controller and final element sit in one device, so the whole dead time lies in the actuation path and none of it in the instrument.

> **TECHNOLOGY** **Sensor: Queueing Delay From a Rate or a Timestamp**
>
> A router measures its queue in one of two ways. It can divide the bytes waiting by an estimate of the departure rate, by Little's law, or it can stamp each packet with the time it joined the queue and read the delay as the packet leaves [16]. A timestamp yields a sample only when a packet reaches the head of the queue. The departure rate varies when other queues share the link or the link rate changes, which is common on wireless links, and short bursts that empty the queue make the rate estimate less accurate [16]. A queue length alone misleads: a drop in link rate shows on its trend as a spike that is not lasting congestion [20].

What Moves It

The final element is a decision taken for each arriving packet: mark it, drop it or let it pass. The marking probability p of Equation (7.1) stands for the share of arrivals that carry either signal, applied at once and with no slew limit, and its hard bounds of 0 and 1 are the actuator limits that scenarios 4 and 8 reach.

> **TECHNOLOGY** **Final Element: The Congestion Mark or the Drop**
>
> The router acts through two bits of the packet header. A sender that supports explicit congestion notification (ECN) sets a capable codepoint, and a congested router sets the congestion experienced codepoint instead of dropping the packet. The receiver echoes the mark in a flag of the TCP header, and the sender must respond as to a single dropped packet, halving its window once for any window of data that held a mark or a drop. A full queue leaves the router no choice but to drop [22]. A mark can be lost on the way: a broken receiver, or a device on the path, can hide it, so a router must still drop excessive traffic even from senders that claim to support marks [23].

Where the Controller Runs

The queue manager runs inside the network device. The cable modem variant of the PI design splits the algorithm into a control path, which updates the drop probability every 16 ms, and a data path, which makes the drop decision as each packet is queued. The data path may be built in hardware, and the control path is best kept in software so that it can be updated [17]. The general design updates every 15 ms by default against a delay target of 15 ms [16], and the controller of this chapter executes every 5 ms. That design also writes its law in increments: each update adds a term on the delay error and a term on the change of the delay since the last update to the drop probability, which is the

velocity form of Equation (7.14). It divides that increment by up to 2048 when the drop probability is small, a gain schedule on the level of congestion [16].

> **IN PRACTICE A Queue Manager Should Need No Tuning**
>
> The best current practice for queue management, RFC 7567 (BCP 197), asks that a queue manager need no tuning of its initial or configuration parameters in common use. Its default must auto-tune to reasonable performance, any manual parameters should be visible to the operator, and its response may have to be set for the largest expected path round trip, which cannot be known at initialization. Logging and alarms may compare input, output and mark or drop rates over time [23]. Scenario 7 of this chapter is the case that the round-trip clause warns about.

What Is Modernizing It

Three developments touch this loop. The first is marking in place of dropping: the Internet Engineering Task Force recommends that deployed queue managers support ECN as well as loss [23]. The second is the move from a target in packets to a target in time, because a queue length controls latency only indirectly [16]. The third is a dual queue that keeps traffic from scalable congestion controls apart from classic traffic and runs a PI controller whose output, squared, sets the drop probability of the classic traffic. In residential tests it held the average queueing delay of the low-latency queue below 1 ms [24].

7.10 Limitations

The simulator is not a network planning tool. It represents one bottleneck, one congestion control algorithm and one round-trip time shared by every session, so it carries no information about multiple congested hops, about the interaction of flows with different delays or about short flows, which are most of the flows on a real link. It has no slow start and no retransmission timeout, so it cannot reproduce the behavior of a path under heavy loss. The fluid approximation removes packets, and with them burstiness, jitter and every question about queue occupancy on a scale shorter than a round trip.

The AQM measurement is idealized in a way that favors the controller. Reading the instantaneous queue exactly, with no averaging and no sampling interval, gives the loop a measurement no deployed queue manager has, and the lag that a real averaging filter adds falls on a loop that has no margin to spare. The delay itself is resolved only to one integration step, which is 2.8 % of the round trip at the default and up to 25 % in the shortest-delay corner. The linear analysis of Section 7.4 holds the delay fixed at R_0 while the simulator lets it move with the queue, so the predicted critical round trip of 297 ms is an approximation to a state-dependent problem and sits 0.7 % above the measured upper bracket of 295 ms.

The controller is a fixed-gain PI loop with no gain scheduling, no delay compensation and no adaptation, which is deliberate but is also the reason scenario 7 fails. A Smith predictor removes the delay from the characteristic equation only to the accuracy of the delay it carries internally [3, 25]. Here that delay is $q/C + T_p$, so a predictor built around a constant loses accuracy exactly when the queue grows and the loop has least margin left. Its internal delay would have to be recomputed from the measured queue. Volume 5, Chapter 5 has a gauge loop that a process model turns into a Smith predictor around a transport delay that is known. A predictive controller of the kind Section 1.6.3 describes would carry the delay inside its model, with the $[0, 1]$ bound on the marking probability and the buffer limit as constraints, and this application does not offer one.

Four complications are retained rather than simplified away, the state-dependent round trip, the hard bounds on the marking probability, the window floor and the buffer limit, and each is retained because the controller has to be designed around it.

OPEN CHALLENGE **A Queue Manager That Tunes Itself**

The recommendations of 2015 call for research on ECN marking that needs no operational tuning, on the right use of delay-based methods, and on flash crowds of short flows that the network signals as if they were long ones [23]. In this chapter the plant gain moves with the session count and the dead time with the path. Progress would be a marking law with no site parameters that keeps its margin across both.

7.11 Discussion

Congestion control decides how a shared link is divided, and it decides the delay every user of that link experiences. The second consequence is the one that took longest to be recognized. A router with a large buffer and no queue manager will fill that buffer, and a full buffer converts unused capacity into latency at no cost in throughput, so a link can be simultaneously 100 % utilized, loss-free by the usual measurement and unusable for anything interactive [18]. Scenario 10 reduces that condition to two numbers, a round trip that goes from 180 to 420 ms while utilization stays at 100 %.

The remedy has been a control problem rather than a hardware problem for twenty-five years. The recommendation to deploy active queue management was made in 1998 and the obstacle named at the time was parameter selection [14]. The fluid model and its linearization turned that obstacle into a design calculation [6, 9], and the algorithms that followed target delay rather than occupancy precisely because a queue expressed in packets means different things on different links [19]. The same argument governs any loop closed over a network, including supervisory control hosted away from the plant and demand-response schemes that report to a central scheduler. Volume 5, Chapter 8 follows a force-reflecting loop closed through such a channel. What matters in those cases is not the average latency of the link but the phase it costs the loop at crossover.

7.12 Conclusion

This chapter studied a congested internet path as a plant whose dead time is set by one of its own two states. The simulator advances the fluid-flow window and queue equations with every marking term evaluated at $t - R$, where $R = q/C + T_p$ is computed from the state being regulated, using a 2048-entry interpolated history seeded with a constant initial function and a fourth-order step made legitimate by a 20 ms floor on the propagation delay.

Linearizing about the default operating point gives two real poles at 0.988 and 5.556/s behind one round trip of dead time with a DC gain of 7119 packets per unit marking probability, and the resulting characteristic equation places the phase margin at 67.6° at a 180 ms round trip and at zero at 297 ms. The simulator settles at 290 ms and sustains an oscillation of 37.4 packets peak to peak at 295 ms, with a measured period of 2.07 s against a predicted 2.09 s.

Around that comparison the application arranges ten scenarios and a graded challenge, including an operating point that does not exist, a correct tuning commissioned onto the wrong path and a drop-tail router that reaches 420 ms of round trip at full utilization. A fuller model would add a delay-compensating controller whose internal delay is recomputed from the measured queue, a spread of round-trip times across the session population and a second bottleneck. The exercises that follow measure what a lengthening delay costs a feedback loop.

KEY POINTS

- A mark made at the router reaches the senders only a full round trip later, so the queue loop carries that trip as dead time.
- At the default PI tuning the phase margin falls from 67.6° at a 180 ms round trip to zero at 297 ms, and the simulator stops settling between 290 and 295 ms.
- The setpoint of this loop sets the dead time of the loop.

Where else this applies. Volume 5, Chapter 8 closes a loop over a delayed channel.

References

[1] V. Jacobson, "Congestion avoidance and control," in *Symposium Proceedings on Communications Architectures and Protocols (SIGCOMM '88)*. ACM, 1988, pp. 314–329.

[2] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1–14, 1989.

[3] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, III, *Process Dynamics and Control*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.

[4] B. W. Bequette, *Process Control: Modeling, Design, and Simulation*. Upper Saddle River, NJ: Prentice Hall, 2003.

[5] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed. Princeton, NJ: Princeton University Press, 2021.

[6] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '00)*. ACM, 2000, pp. 151–160.

[7] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A control theoretic analysis of RED," in *Proceedings IEEE INFOCOM 2001*, vol. 3. IEEE, 2001, pp. 1510–1519.

[8] ——, "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings IEEE INFOCOM 2001*, vol. 3. IEEE, 2001, pp. 1726–1734.

[9] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transactions on Automatic Control*, vol. 47, no. 6, pp. 945–959, 2002.

[10] S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 28–43, 2002.

[11] J.-P. Richard, "Time-delay systems: an overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, 2003.

[12] J. Nagle, "Congestion control in IP/TCP internetworks," RFC Editor, RFC 896, 1984.

[13] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[14] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on queue management and congestion avoidance in the internet," RFC Editor, RFC 2309, 1998.

[15] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

[16] R. Pan, P. Natarajan, F. Baker, and G. White, "Proportional integral controller enhanced (PIE): A lightweight control scheme to address the bufferbloat problem," RFC Editor, RFC 8033, 2017.

[17] G. White and R. Pan, "Active queue management (AQM) based on proportional integral controller enhanced (PIE) for data-over-cable service interface specifications (DOCSIS) cable modems," RFC Editor, RFC 8034, 2017.

[18] J. Gettys and K. Nichols, "Bufferbloat: Dark buffers in the internet," *Queue*, vol. 9, no. 11, pp. 40–54, 2011.

[19] K. Nichols and V. Jacobson, "Controlling queue delay," *Queue*, vol. 10, no. 5, pp. 20–34, 2012.

[20] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, "Controlled delay active queue management," RFC Editor, RFC 8289, 2018.

[21] K. J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2006.

[22] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP," RFC Editor, RFC 3168, 2001.

[23] F. Baker and G. Fairhurst, "IETF recommendations regarding active queue management," RFC Editor, RFC 7567 (BCP 197), 2015.

[24] K. De Schepper, B. Briscoe, and G. White, "Dual-queue coupled active queue management (AQM) for low latency, low loss, and scalable throughput (L4S)," RFC Editor, RFC 9332, 2023.

[25] O. J. M. Smith, "Closer control of loops with dead time," *Chemical Engineering Progress*, vol. 53, no. 5, pp. 217–219, 1957.

7.13 Exercises

7.1 Concept check · 2 min

The propagation delay of the default network lengthens until the round trip at the 100-packet target goes from 180 to 300 ms. The tuning, the traffic and the capacity stay as they were. Why does the phase margin fall from 67.6° to −1.3°?

(a) The dead time alone takes the phase, at a crossover frequency that stays at 2.09 rad/s

(b) The process gain alone rises, as the cube of the round trip, and the dead time takes the same phase as before

(c) The process gain falls with the longer round trip, so the integral term adds its lag nearer to crossover

(d) The dead time takes more phase, and the gain rising as the cube of the round trip moves crossover up, where it takes more still

7.2 Concept check · 1 min

The gain from the marking probability to the queue length is negative. The queue rises above its setpoint. What must the controller do, and what is that action called in the convention of Section 1.4.3?

(a) Raise the marking probability, which is direct action because the output rises with the measurement

(b) Raise the marking probability, which is reverse action because the process gain is negative

(c) Lower the marking probability, which is reverse action because the output falls as the measurement rises

(d) Lower the marking probability, which is direct action because the error $q_{SP} - q$ is negative

7.3 Concept check · 2 min

The tuning shipped with the challenge, $K_c = 15\,\%$ per 100 packets and $\tau_I = 1.2\,\text{s}$, has a phase margin of 36.8° with 50 sessions on the link and −40.3° with 20. What makes the lighter load the harder case?

(a) Fewer sessions lengthen the round trip at the same queue target, so the dead time takes more phase

(b) The process gain varies as N^{-2}, so it is 6.25 times higher with 20 sessions than with 50

(c) The bias p_0 is computed for 50 sessions, so the integral term winds up when the crowd leaves

(d) Each window is larger with 20 sessions, so the one-packet window floor limits the marking probability

7.4 Theory and calculation · 5 min

In scenario 3 the capacity of the link falls from 1250 to 600 packets per second. The 20 sessions, the propagation delay of 100 ms and the 100-packet target stay.

(a) Find the round trip, the window and the marking probability of the new operating point. *Checkpoint:* 267 ms, 8.00 packets and 3.03 %.

(b) Find the queueing delay of the 100-packet target before and after the cut. *Checkpoint:* 80 and 167 ms.

(c) Find the largest session count for which an operating point exists on the slower link at this target. *Checkpoint:* 160 sessions.

7.5 Theory and calculation · 8 min

The third step of scenario 4 puts the round trip at $R_0 = 300\,\text{ms}$, with $C = 1250$ packets per second, $N = 20$ and the default tuning, $K_c = 3.5 \times 10^{-4}$ per packet and $\tau_I = 5\,\text{s}$.

(a) Find the two poles and the steady gain $|P(0)|$ of the linearized plant, and the factor by which the gain has risen from the 7119 packets of the 180 ms round trip. *Checkpoint:* 0.356 and 3.33/s, 32 959 packets, a factor of 4.63.

(b) The loop gain crosses unity near 3.02 rad/s. Find the phase lag of the dead time at that frequency. *Checkpoint:* 51.9°.

(c) Add the phase of the PI controller and of the two poles at the same frequency and find the phase margin. *Checkpoint:* −1.2°, within 0.2°.

7.6 Theory and calculation · 5 min

With the marking probability at 100 % every window rests on its one-packet floor, so N sessions offer N/R packets per second.

(a) For the 500 sessions of scenario 8 on the default link, find the queue and the round trip at which the offered load equals the capacity, and the error that remains from the 100-packet target. *Checkpoint:* 375 packets, 400 ms and 275 packets.

(b) Back-calculation with $\tau_t = \tau_I$ holds the demand where $c_k - p_k = K_c e_k$. Find the pre-clamp command for $K_c = 3.5 \times 10^{-4}$ per packet. *Checkpoint:* 1.096.

(c) Find the session count at which this resting queue reaches the 400-packet buffer. *Checkpoint:* 525 sessions.

7.7 App challenge · 8 min

Load scenario 5, Sudden traffic arrival (flash crowd). Forty sessions join the 20 at 6 s. Run the scenario as shipped for 45 s at the default seed and export the file. The probability columns hold fractions, so 0.0299 is 2.99 %.

(a) Read `mark_probability` in the row at 6.20 s. Then find the first row after 6.00 s in which `mark_probability_delayed` has left 1.556 %. *Checkpoint:* 2.99 %, and the row at 6.25 s.

(b) Find the largest value of `queue_packets` and its time. *Checkpoint:* 226.8 packets at 7.20 s, within 0.5 packets.

(c) Read `mark_probability` and `window_packets` in the row at 45 s, and compare them with the operating point for 60 sessions. *Checkpoint:* 11.8 % and 3.88 packets, against 12.45 % and 3.75 packets.

Explain the delay found in part (a) from the queue in the same rows.

7.8 App challenge · 9 min

Load scenario 4, Propagation delay, step by step. Run it as shipped for 100 s at the default seed and export the file.

(a) Find the peak-to-peak swing of `queue_packets` over 41 to 52 s, 63 to 74 s and 87 to 100 s. *Checkpoint:* 2.0, 40.4 and 54.4 packets, each within 0.5 packets.

(b) Find the share of the rows in which `mark_saturated` is 1 over the last two of those windows. *Checkpoint:* 22 and 42 %, each within one point.

(c) Measure the period between 63 and 74 s from successive peaks of the queue. *Checkpoint:* 2.10 s, within 0.05 s.

Name the step at which the loop stops settling, and compare it with the margins of Table 7.6 and the critical round trip.

7.9 App challenge · 8 min

Load the Challenge, ride out a flash crowd. Thirty sessions join at 8 s and leave at 32 s, and the target is 120 packets. Run each case for 75 s at the default seed and export the file.

(a) Run the case as shipped, with the Proportional gain at 15 and the Integral time at 1.2 s. Find the highest queue between 8 and 32 s, and the lowest and the highest over the last 25 s. *Checkpoint:* 172.5 packets, then 74.1 and 144.9 packets.

(b) Set the Proportional gain to 3.5 and the Integral time to 5.0 s, press Reset and run again. Find the same three values. *Checkpoint:* 214.7 packets, then 111.5 and 119.9 packets.

(c) In the file of part (a), read `loop_dc_gain_packets` in the rows at 5 and 30 s. *Checkpoint:* 9191 and 1471 packets, a ratio of 6.25.

Accept each queue value within 0.5 packets. Say which run holds the ±25-packet band over the last third of the run, and why the shipped tuning fails only after the crowd has left.

7.10 Application area · 3 min

A queue target is a latency budget written in packets.

(a) Find the queueing delay of a 100-packet target on the default link of 1250 packets per second and on a link of 600 packets per second. *Checkpoint:* 80 and 167 ms.

(b) An operator specifies 5 ms of queueing delay instead. Find that target in packets on each link, and say what it does to the dead time of the loop when the capacity changes. *Checkpoint:* 6.25 and 3 packets.

7.11 Application area · 4 min

The model hands the controller an exact and instantaneous queue. Suppose a deployed queue manager averages the queue with a first-order filter of time constant 0.1 s and updates the marking probability every 0.1 s, under the default tuning.

(a) Estimate the phase that the filter and the update period take at the default crossover of 2.09 rad/s. Treat the update period as half a period of added dead time (Section 1.4.11). Find what is left of the 67.6° margin. *Checkpoint:* 11.8° and 6.0°, which leave about 50°.

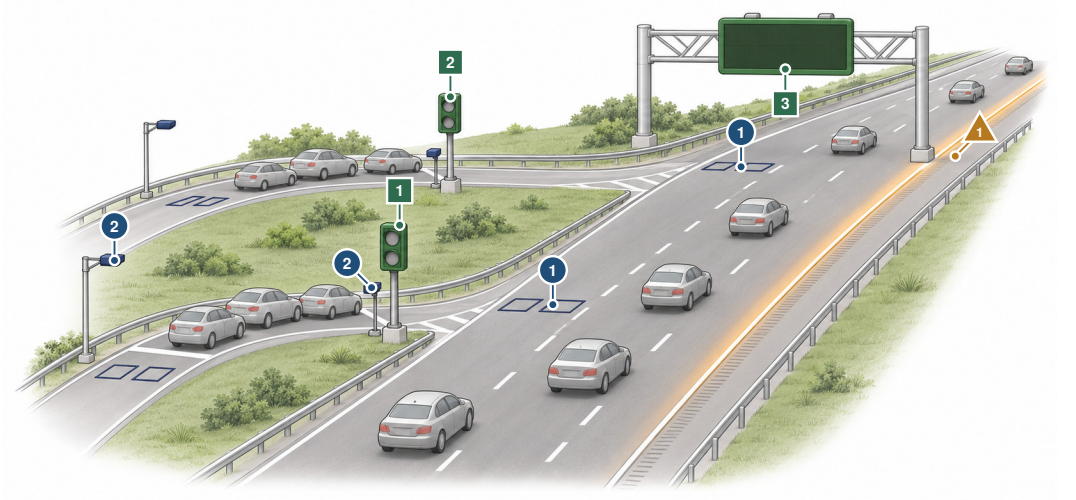
(b) Decide whether the same filter and period can be used at the 260 ms round trip, where the margin is 16.9°. *Checkpoint:* No. They take at least 17.8° there.

7.12 Review the agent · 5 min

An engineer asks an AI assistant how to give the default router more room for a flash crowd. The assistant answers: “Raise the queue setpoint from 100 to 250 packets. The buffer holds 400 packets, so 150 packets of headroom remain. The setpoint is the target of the loop and not part of the plant, so the default tuning keeps its phase margin of 67.6°, and the only cost is a fuller buffer.” Find the flaw. Say how a calculation and one run of the app would each expose it.

CHAPTER 8

Ramp Metering Simulator for Holding a Freeway Merge Short of Breakdown



The controller moves 1 the signal on the upstream ramp 2 the signal on the merge ramp 3 the speed limit sign over the freeway

The controller reads 1 loops in the road at each merge, which sense how much of the time a car is over them

2 counters where cars join each ramp and where they leave its signal

What disturbs it 1 a crash past the merge that closes part of the road

Notice the cars held at both signals: the ramps store the wait that keeps the freeway flowing past the merges. Illustration, not to scale.

IN PLAIN TERMS **Holding a Freeway Merge Short of a Jam**

The main idea. Where an entrance ramp joins a freeway, the road can carry only so many cars. A signal on the ramp lets cars in at a pace that a controller sets, so that the merge stays just below its limit.

The challenge. If the merge gets too full it jams, and a jammed merge passes about a tenth fewer cars until the jam is gone. Every car held back waits on the ramp, and a ramp has room for only so many.

Why it matters. A few minutes of too much traffic can cost more than an hour of jam.

THREE WAYS TO READ THIS CHAPTER

Explore	12 min	the first page, the fact on the count of ramp meters (Section 8.1), the two history segments (Section 8.2), the try-it segments of cases 1 and 4 and the fact on the fixed plan (Section 8.5), and the key points
Study	50 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	20 min	the first page, the results (Section 8.5), Technology in Practice (Section 8.6), Limitations (Section 8.7) and the open challenges

The control question A freeway merge passes the most traffic just before it breaks down, and the queue it then forms discharges a tenth less until it is gone. How far below that fold should an integral controller hold the merge, and what happens when the ramp storage that pays for the margin runs out?

Learning objectives

- Compute the critical occupancy, the dropped capacity and the recovery density of a merge from its fundamental diagram.
- Explain how a capacity drop with hysteresis turns minutes of overload into more than an hour of breakdown.
- Show that ALINEA is integral control on occupancy, and predict its decay per report from the static gain of the merge.
- Choose an occupancy setpoint with a margin below the fold, and weigh the time saved on the freeway against the ramp queue.
- Judge what a queue override, coordination and a detector watchdog each protect, and what each costs.

Prerequisites Sections 1.4.2 and 1.6.2 (velocity form), 1.4.9 (windup and back-calculation) and 1.4.11 (a loop that acts once per report). Section 1.6.3 is needed only for the predictive comparison.

The app https://apmonitor.com/apps/ramp-metering/

See also Volume 7, Chapter 4 for a setpoint at the peak of the plant's map and Volume 7, Chapter 2 for folds with ignition and extinction. Volume 4, Chapter 2 for a protective override beside a folded map and Volume 4, Chapter 14 for a finite store spent to defend a target.

8.1 Introduction

A lane of freeway carries more vehicles as its density rises, up to a critical density, and fewer as the density rises further, so the steady relation between flow and density folds over at a peak. The kinematic wave theory built on that relation treats traffic as a fluid whose changes travel as waves and whose abrupt changes are shock waves [1, 2], and the cell transmission model represents highway traffic in cells consistently with that hydrodynamic theory [3]. Ramp metering puts a traffic signal on the on-ramp that releases its vehicles at a rate the controller chooses, and an overview of the field explains why it can lead to a dramatic amelioration of traffic conditions on freeways [4]. ALINEA, a local feedback strategy, reported easy application, flexibility and high efficiency in its field applications in Paris and Amsterdam [5].

The mechanism is not obvious, because every vehicle held on the ramp still has to pass the same merge. On the average Tuesday of the simulator the demand at the merge exceeds its 6000 veh/h for 17 min in all, from 7:45 to 8:05, yet with the signal dark the merge breaks down at 7:48 and stays in the capacity drop for 104.9 min, and the morning costs 1840.7 veh·h. ALINEA holding an occupancy of 15.5 % keeps the merge flowing to 10:00 for 1529.4 veh·h, 16.9 % less, while the ramp queue reaches 163.9 vehicles, 103.9 of them past the ramp's storage. The simulator answers the objection that a meter only moves delay from the freeway to the ramp. It moves delay, but the merge it protects keeps a tenth of its capacity that a breakdown would throw away, and the storage that holds the moved delay is finite.

> ✦ DID YOU KNOW? **About 2,160 Ramp Meters**
>
> In 2002 the United States had about 2,160 ramp meters, and about 2,370 were expected by the end of 2005. Metering systems ran in 26 metropolitan areas. Each meter is a signal on an entrance ramp with its controller in a cabinet beside the ramp [6].

The same shape recurs outside traffic. When a computer metered the traffic entering the Lincoln Tunnel in 1967, congestion was already known to reduce the tunnel's throughput appreciably [7]. In the first congestion collapse of the early Internet, in October 1986, the data throughput between two

sites fell from 32 kbps to 40 bps [8], and Chapter 7 holds the queue of such a link with a PI controller. An axial compressor at its stall line falls into surge, or into rotating stall at a substantially reduced flow and pressure ratio [9], and Volume 4, Chapter 2 keeps a centrifugal compressor short of surge with a protective override. In each the operating point should sit near the peak of a map that folds, a step past the peak costs a lower branch, and whatever is held back must wait somewhere. None of the three survives a linearization of the merge around free flow, where the release simply adds to the occupancy.

However, the evaluations of metering cited here were made in the field, at ramps in Paris and Amsterdam and during a metering holiday in the Twin Cities [5, 10], where no morning can be run twice, and a student cannot replay one morning with one setting changed and read the minutes in the drop, the ramp queue and the total time spent side by side. This chapter takes the problem apart into the margin below the fold, the storage behind the meter, the measurement it trusts and the actuator it uses, each with its own setting on a simulated merge whose morning replays identically. The cell transmission model, the capacity drop, ALINEA and its extensions are established methods, and the chapter applies them as published.

Section 8.2 traces freeway control from the kinematic wave to coordinated metering. Section 8.3 gives the model in full, enough to rebuild it without the source: a capacity drop held by a per-cell latch with hysteresis on the cell transmission model, detectors and counters that report once a minute, and a score integrated from the model's own state. Section 8.4 gives the meters as implemented. They read only the minute reports of occupancy and vehicle counts, and ALINEA runs on the velocity-form PID of Section 1.6.2. Section 8.5 reports the measured mornings, ten cases and a graded morning, each isolating one part of the problem and each credited by its own completion rule. Section 8.6 describes the devices of a real meter, Section 8.7 states the limitations, Section 8.8 gathers the results, and the exercises follow in Section 8.9.

8.2 From Kinematic Waves to Ramp Meters

The theory came first. In 1955 a functional relation between flow and concentration on crowded roads, which by then had experimental backing, was used to deduce how changes in the distribution of traffic propagate, to show that vehicles entering a region of higher concentration must reduce speed rather suddenly at a shock wave, and to estimate the extent and duration of the hold-up at a bottleneck too narrow to admit the increased flow [1]. A year later the same picture was reached by replacing individual vehicles with a continuous density and an empirical relation between speed and density, and a traffic signal was found to cause a threshold effect, minor disturbances in light traffic that suddenly build to large values once a critical density is exceeded [2].

Control followed within a decade. A 1965 volume based on papers presented at an interdisciplinary clinic on instrumentation requirements for traffic control systems, held in New York City in December 1963, included experimentation with manual and automatic ramp control [11]. The history box below follows the first ramp meters and a tunnel metered by a computer [7]. By 1973 automatic freeway surveillance and control systems had been introduced in several cities of the United States over the preceding decade, and mathematical models of freeway traffic that allow on-ramp metering were being developed to evaluate and optimize them [12].

> ⧖ HISTORY **From a Police Officer to Feedback, 1963 to 1972**
>
> The first ramp meters went up in 1963 on an expressway in Chicago, and a traffic enforcement officer in the field controlled them by hand. Experiments followed in Detroit and Los Angeles [6]. During most of 1967 an experimental system counted the vehicles in three sections of one lane of the Lincoln Tunnel from detectors at four points, on a computer forty miles away, and reduced the traffic let in whenever congestion threatened the tunnel's throughput. It gave longer periods of sustained high flow [7]. In 1970 the first two meters near St. Paul ran on fixed time, and two years later they were upgraded to respond to traffic [6], the step from case 2 to case 3 of this chapter.

The strategies then grew from fixed-time plans, through traffic-responsive regulators, to nonlinear optimal control [4]. ALINEA, Asservissement Linéaire d'Entrée Autoroutière, a local feedback strategy, had by 1997 been applied at single and multiple ramps of the Boulevard Périphérique in Paris and on the A10 West freeway in Amsterdam [5], and it uses occupancy measured in the merging area, at most a few hundred meters downstream of the ramp [13]. Metering an on-ramp was studied as a way to increase the capacity of an isolated merge [14], and the capacity drop has been observed at freeway merges under ramp control and integrated into a first-order model [15]. The cell transmission model [3] was given an asymmetric form for the optimal metering of freeway ramps [16].

Successors extended the local law. A linked strategy, responsive to traffic, coordinates the metering of two or more consecutive on-ramps with ALINEA at the local level, uses only real-time measurements, and in simulation reached the efficiency of proactive optimal control [17]. When the bottleneck lies further downstream than the merge, ALINEA may give a poorly damped loop, and its proportional-integral extension, PI-ALINEA, restores satisfactory performance [13]. Variable speed limits act on aggregate flow in a way similar to ramp metering at potentially active bottlenecks and can be integrated with coordinated metering [18]. Ramp meters have also been put on trial, with evidence from a metering holiday in the Twin Cities [10].

> ⧖ HISTORY **A Metering Holiday, 2000**
>
> In 2000 a small but vocal group of citizens opposed to ramp metering prompted the Minnesota legislature to have the meters of the Twin Cities switched off for six weeks and to fund an independent evaluation. Traffic was measured for six weeks with the meters and six weeks without. With the meters off, the volumes on the freeway mainline fell by nine percent, freeway speeds fell by 14 percent, travel times became nearly twice as unpredictable, and crashes on the freeways and ramps rose by 26 percent. The system's benefits were put at about 40 million dollars, 15 times its costs [6]. The agency then moved from metering every rebuilt freeway by default to a more performance-based approach [6].

8.3 The Freeway and Its Model

The road is six kilometers of three-lane freeway, traffic from left to right, divided into $N = 12$ cells of $L = 0.5\,\text{km}$, as shown in Figure 8.1. The merge ramp joins cell $m = 8$ at 4.0 km and the upstream ramp, closed unless a case opens it, joins cell $m_1 = 4$ at 2.0 km. Speed limits can hold cells 6 and 7, over the kilometer before the merge, and an incident halves cell 10, a kilometer past it. The signals are classified in Table 8.1. The integrated state holds 39 numbers, 24 density slots of which the 12 cells use 12, the three queues q_0, q_1 and q, and 12 registers that the detectors, counters and score accumulate. Everything discrete, the capacity-drop latches, the incident's end, the queue accounting and the end of the morning at 10:00, is updated after each integration step.

8.3.1 Cells and the Fundamental Diagram

Each cell balances the vehicles that enter and leave it, with time in hours, as shown in Equation (8.1).

$$\lambda L \frac{\mathrm{d}k_i}{\mathrm{d}t} = f_{i-1} - f_i + r\,\delta_{i,m} + r_1\,\delta_{i,m_1} \tag{8.1}$$

where k_i is the density per lane in veh/km, $\lambda = 3$ the lanes, f_i the flow in veh/h from cell i to cell $i+1$, and $\delta_{i,j}$ the Kronecker delta. The flow between two cells is the smaller of what the upstream cell can send and what the downstream one can receive, as shown in Equation (8.2).

$$\begin{aligned} S_i &= \lambda \,\min\left(v_i k_i,\ (1-\delta\,\ell_i)\,Q_i\right) \\ R_i &= \lambda \,\max\left(0,\ \min(Q_i,\ w(k_{jam}-k_i))\right) \\ f_i &= \min(S_i,\ R_{i+1}), \qquad f_{N-1} = S_{N-1} \\ f_{-1} &= \min\left(\lambda Q,\ q_0/\Delta t + D_m,\ R_0\right) \end{aligned} \tag{8.2}$$

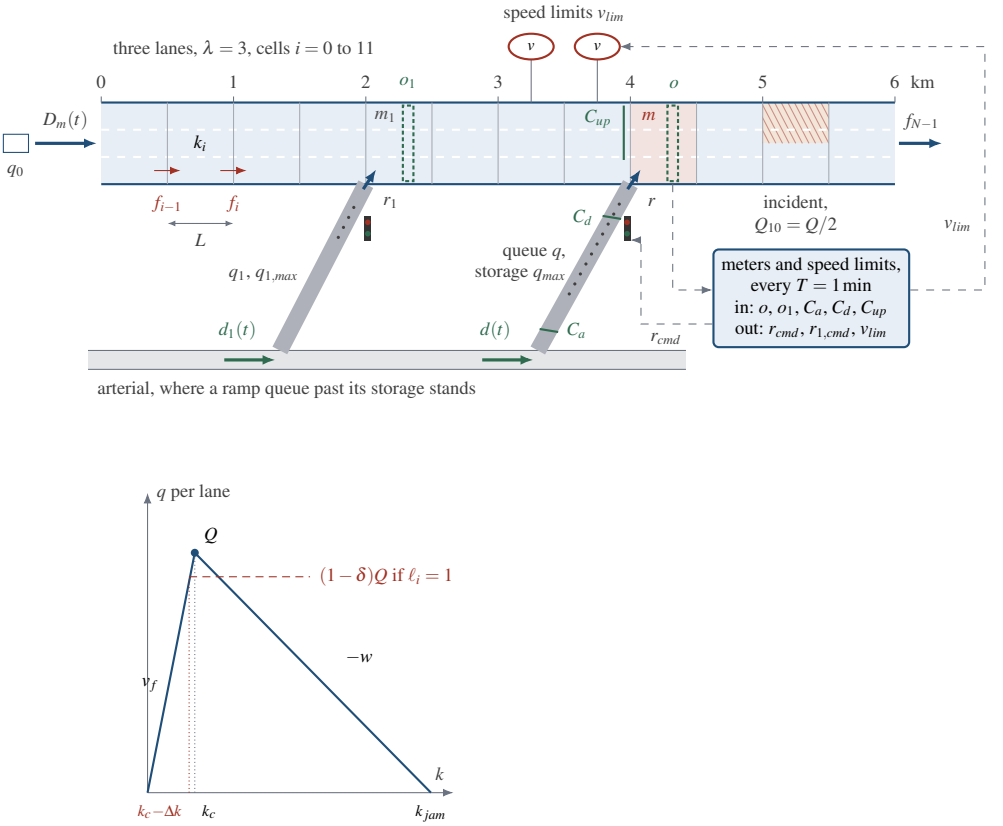


Figure 8.1 The six kilometers in plan, to scale along the road and not across it. Cell i of length L and λ lanes holds the density k_i and passes the flow f_i to cell $i+1$, the mainline demand $D_m(t)$ enters cell 0 through the queue before the map q_0, and f_{N-1} leaves at 6 km. The merge ramp joins the merge cell m with the release r from a queue q of storage q_{max} that the arrivals $d(t)$ feed, and a queue past the storage stands on the arterial. The upstream ramp is the same with r_1, q_1, $q_{1,max}$ and $d_1(t)$ at cell m_1. Loop detectors in cells m and m_1 report the occupancies o and o_1, counters at each ramp's entrance and stop line give C_a and C_d, and a counter at the upstream face of the merge cell gives the mainline count C_{up}. Every T the controller sets r_{cmd}, $r_{1,cmd}$ and the speed limit v_{lim} over cells 6 and 7. The incident halves the capacity of cell 10. The inset is the fundamental diagram per lane, with free speed v_f, wave speed w, capacity Q at the critical density k_c, jam density k_{jam}, the dropped capacity $(1-\delta)Q$ that a cell sends while its latch ℓ_i is set, and the density $k_c - \Delta k$ below which the latch releases.

where S_i and R_i are the sending and receiving flows, v_i and Q_i the free speed and capacity of cell i now, $\ell_i \in \{0,1\}$ its capacity-drop latch, δ the drop, w the congestion wave speed, k_{jam} the jam density, f_{-1} the inflow to cell 0 from the mainline demand D_m of Equation (8.8) and the queue before the map q_0, and Δt the integration step, so that $q_0/\Delta t$ releases that queue in one step. The nominal diagram is a triangle whose three declared numbers close it, as shown in Equation (8.3).

$$Q = v_f k_c = w\,(k_{jam} - k_c) \tag{8.3}$$

so that $k_c = Q/v_f$ and $k_{jam} = k_c + Q/w$. With $Q = 2000$ veh/h per lane, $v_f = 80$ km/h and $w = 16$ km/h the critical density is 25 veh/km and the jam density 150 veh/km per lane. Holding the capacity, a critical density of 25 and a jam density of 150, the triangle closes only at these two speeds, and a free speed of 100 km/h would need a capacity of 2500 veh/h per lane. Rain at a fraction ρ

Table 8.1 Signals of the Freeway by Role, With Their Defaults

Symbol	Role	Default	Range
o	CV, merge occupancy (reported)	SP 15.5 %	SP 12–20
o_1	CV, upstream occupancy	SP 15.5 %	SP 12–20
r	MV, merge ramp release	ALINEA	240–1800 veh/h
r_1	MV, upstream ramp release	ALINEA	240–1800
v_{lim}	MV, speed limit, cells 6, 7	off	20–80 km/h
k_i	states, 12 cell densities	free flow	0–150 veh/km
q, q_1	states, ramp queues	0	storage 60, 150
q_0	state, queue before the map	0	—
—	12 registers, counts and score	0	—
D_m, d, d_1	DV, demand by weekday	Tuesday	5 days
—	DV, scaling, surge	0	±10, 0–20 %
—	DV, incident at 5 km	off	20 min event
—	DV, an event lets out	off	+300 veh/h, 30 min
—	DV, rain, diverted traffic	0	20 %, 600 veh/h
—	DV, detector noise, freeze	2 %, off	0–10 %
C_a, C_d	measured, ramp counts	exact	—
C_{up}	measured, mainline count	exact	—
TTS	score, total time spent	—	veh·h

lowers v_i and Q_i by the same share, so the critical density stays where it was. The incident halves Q_{10} for 20 min. A speed limit v_{lim} below the free speed keeps the congested branch and moves the peak, $v_i = v_{lim}$ and $Q_i = \min\big(Q_i,\ v_{lim} w k_{jam}/(v_{lim}+w)\big)$, which is 1565 veh/h per lane at 30 km/h. Each cell's critical density is $k_{c,i} = Q_i/v_i$.

8.3.2 The Capacity Drop

After every step each cell's latch follows the rule of Equation (8.4).

$$\ell_i \leftarrow \begin{cases} 1, & \ell_i = 0 \text{ and } k_i > k_{c,i} \\ 0, & \ell_i = 1 \text{ and } k_i < k_{c,i} - \Delta k \\ 0, & \text{a speed limit holds cell } i \\ \ell_i, & \text{otherwise} \end{cases} \tag{8.4}$$

with $\delta = 10\,\%$ and $\Delta k = 3$ veh/km per lane, declared teaching values. The gap between the two densities is the hysteresis of the drop. A latched merge sends at most $(1-\delta)\lambda Q = 5400$ veh/h instead of 6000, and it recovers only once its density falls below 22 veh/km per lane, where free flow carries 5280 veh/h. In free flow no cell reaches its critical density, so the latch acts at the head of a queue, and the merge's latch is the one the morning is scored on. The incident cell latches too, so while it is blocked it discharges 2700 veh/h, not the 3000 its halved capacity would allow. The exemption under a speed limit is the premise of mainstream control, that vehicles held to a limit accelerate to it rather than out of a jam.

8.3.3 Ramps, Queues and the Merge

Each ramp's queue gains its arrivals and loses its release, and the release is limited four ways, as shown in Equation (8.5).

$$\begin{aligned} \frac{\mathrm{d}q}{\mathrm{d}t} &= d - r, \qquad \frac{\mathrm{d}q_0}{\mathrm{d}t} = D_m - f_{-1} \\ r &= \max\Big(0,\ \min\big(r_{cmd},\ r_{max},\ q/\Delta t + d,\ \tfrac{1}{3}R_m\big)\Big) \end{aligned} \tag{8.5}$$

where r_{cmd} is the meter's command and $r_{max} = 1800\,\text{veh/h}$ a two-lane signal's highest rate, and q_1 and r_1 obey the same equations with d_1 and R_{m_1}. The ramp enters on top of the mainline inflow, up to a third of the merge cell's receiving flow, which the application describes as the merge of the asymmetric cell transmission model [16]. In a congested merge whose ramp has vehicles waiting that share is a quarter of what the merge passes, and when mainline and ramp together bring more than the merge can pass, the merge cell fills past k_c, which is the breakdown. A queue longer than its storage is not blocked. The extra vehicles stand on the arterial, are counted as spillback, and still count in the total time spent.

8.3.4 Detectors, Counts and the Score

A loop detector reports the share of time a vehicle is over it, the occupancy, which in steady traffic is the density times the effective vehicle length. Every reporting period T the merge station reports the period's mean, as shown in Equation (8.6).

$$\begin{aligned} o(k) &= \max\left(0,\ \frac{L_{eff}}{1000}\,\bar{k}_m(k)\,\left(1+\sigma_n \nu_k\right)\right) \\ L_{eff} &= L_{veh} + g_{det} \end{aligned} \tag{8.6}$$

where $\bar{k}_m(k)$ is the mean density of the merge cell over report k, $L_{veh} = 4.5\,\text{m}$ the mean vehicle length, $g_{det} = 2.0\,\text{m}$ the detection zone, $\sigma_n = 2\,\%$ the noise and ν_k a standard normal deviate from the seeded generator. The critical occupancy is $L_{eff}k_c/1000 = 16.25\,\%$, one percentage point of merge occupancy is 369 veh/h of free-flow discharge, and the default setpoint of 15.5 % stands for 23.85 veh/km per lane and 5723 veh/h. The upstream station reports o_1 the same way. A frozen detector repeats its last reading. The counters at each ramp's entrance and stop line give the arrivals $\hat{d} = 60\,\Delta C_a/T$, the release $\hat{r} = 60\,\Delta C_d/T$ and the queue $\hat{q}_k = \max(0,\ \hat{q}_{k-1} + \Delta C_a - \Delta C_d)$, and the mainline counter gives the flow into the merge cell $\hat{q}_{up} = 60\,\Delta C_{up}/T$, with T in minutes. The meters read nothing else from the road.

The score is the total time spent of Equation (8.7).

$$\text{TTS}(t) = \int_{6:00}^{t} \left(\sum_{i=0}^{N-1} \lambda L k_i + q_0 + q_1 + q\right) dt \tag{8.7}$$

It is integrated as a state, from the model's own vehicles rather than from any detector, and every vehicle-hour on the freeway, the ramps and the arterial counts the same.

8.3.5 The Morning's Demand

The demand is declared, not routed. Each stream s, the mainline, the merge ramp and the share of the mainline that uses the upstream ramp when it is open, follows Equation (8.8).

$$D_s(t) = P_s(t - \tau_d)\,\kappa_d\left(1 + \frac{a}{100}\right)\left(1 + \frac{u}{100}\right)\left(1 + \varepsilon_{s,d}(t)\right) \tag{8.8}$$

where P_s is the average weekday of Table 8.2, interpolated linearly in time, κ_d and τ_d the scale and shift of weekday d, a the demand scaling and u the surge, both in percent, and $\varepsilon_{s,d}$ a first-order autoregressive fluctuation of 2 % standard deviation and 6 min correlation time, drawn once a minute from a generator seeded by the weekday and interpolated between minutes. Tuesday is the average day. Monday is 1.5 % lighter and 6 min later, Wednesday 1.5 % heavier and 4 min earlier, Thursday 3 % heavier and 8 min earlier, and Friday 4.5 % lighter and 12 min later. When the upstream ramp is open its stream is taken out of the mainline rather than added, so the merge sees the same morning. Diverted traffic and an event letting out add to the merge ramp. On Tuesday the mainline and the merge ramp bring 4396 veh/h at 6:00, peak at 6147 veh/h at 7:51 and are above 6000 veh/h for 17 min.

Table 8.2 The Average Weekday, veh/h, at the Knots of $P_s(t)$

Clock	6:00	6:20	6:50	7:20	7:45	8:05	8:30	8:55	9:25	10:00
Mainline	3975	4020	4415	4770	4900	4900	4725	4370	4020	3800
Merge ramp	520	540	720	920	1050	1050	915	720	595	520
Upstream	380	390	470	560	600	600	560	480	420	380

The upstream row is part of the mainline, used only while that ramp is open.

Table 8.3 Model Constants. Class V Is a Slider in the Show Model Dialog, H a Hidden Parameter and C Fixed in the Code

Symbol	Meaning	Value	Class
Fundamental diagram and breakdown			
v_f	free-flow speed	80 km/h	V
Q	capacity per lane	2000 veh/h	V
w	congestion wave speed	16 km/h	V
λ	lanes	3	H
k_c, k_{jam}	critical and jam density, per lane	25, 150 veh/km	derived
δ, Δk	capacity drop, recovery margin	10 %, 3 veh/km	V
Geometry and numerics			
L, N	cell length, cells	0.5 km, 12	V, at reset
Δt	Euler step	2 s	C
—	ramp share of R_m	1/3	C
—	incident: cell 10, capacity, duration	$Q/2$, 20 min	C
Ramps			
q_{max}, $q_{1,max}$	storage, merge and upstream ramp	60, 150 veh	V
r_{min}, r_{max}	lowest and highest release	240, 1800 veh/h	V
Detectors and demand			
L_{veh}, g_{det}	vehicle length, detection zone	4.5, 2.0 m	V
T	reporting period	1 min	V
—	fluctuation, deviation and correlation	2 %, 6 min	C
—	noise seed, weekday seeds	20261001, 101–505	C

8.3.6 Initial State and Numerics

The constants are collected in Table 8.3, and the settings a student changes in Table 8.4. Every case starts at 6:00 in steady free flow at the opening demand, each cell at the density $\min(\phi_i/\lambda,\ 0.999Q)/v_f$ for the opening flow ϕ_i through it, 302.6 vehicles in all, with every queue empty, and the run pauses at 10:00. The states advance by explicit Euler at $\Delta t = 2$ s, a step that the application supplies in place of the engine's default (Section 1.6.1), because the cell transmission model is defined as that forward step, and the Courant number $v_f \Delta t / L$ is 0.089. Each step follows the order of Figure 1.10, a scripted event due at its start, then a due report and the meters' action, the Euler update, the latches with the queue accounting, the outputs, and every 10 s a log row. A row of the exported file holds the merge occupancy reported and true, the releases commanded and made, the arrivals, the queues, the speed limit, the merge's discharge and capacity, the drop flags, twelve segment densities, the running total time spent, and the disturbances, parameters and settings in force. Halving the step moves the total time spent of cases 1 and 3 by 0.005 % and 0.003 %, and halving the cell length, to 24 cells, moves it by 0.080 % and −0.027 % and moves the breakdown of case 1 by 14 s. The application passes its 58 verification checks, among them vehicle conservation to 10^{-9} of the arrivals and a jam front that travels upstream at w within 5 %.

8.3.7 Assumptions

A first-order model. Densities and flows are cell averages on a triangular diagram, so every wave inside a queue travels at w, fronts are smeared over a cell, and the stop-and-go oscillations that real queues carry are absent.

Table 8.4 Meter Settings, Defaults and Ranges

Symbol	Setting	Default	Range
—	merge metering strategy	ALINEA	4 choices
$\hat{o}$	merge occupancy setpoint	15.5 %	12–20
K_R	ALINEA gain, veh/h per % per min	70	0–200
K_P	PI-ALINEA gain, veh/h per %	0	0–200
r_{man}	manual release	1800 veh/h	0–1800
$\hat{C}$	capacity estimate, demand-capacity	5800 veh/h	4500–6500
—	stuck-detector watchdog	off	on, off
—	queue override	off	on, off
q_{lim}	override limit, share of storage	85 %	30–100
—	upstream ramp	closed	open
$\hat{o}_1$, K_{R1}	upstream setpoint and gain	15.5 %, 70	as above
—	coordination	off	on, off
—	variable speed limits	off	on, off
$\hat{o}_V$	speed-limit occupancy setpoint	16.0 %	12–20
K_V	speed-limit gain, km/h per % per min	2	0–20
v_{man}	manual speed limit	80 km/h	20–80

A deterministic drop. The capacity drop is a latch with two declared numbers. The application's own assumptions add that real breakdowns near capacity are probabilistic rather than certain past a density, and that the size of the drop differs between sites and mornings.

A lumped merge. Weaving, lane changing and the length of the acceleration lane are lumped into the ramp's one-third share of the merge cell's receiving flow.

Declared demand. Nobody diverts when a ramp queue grows or the freeway jams, and nobody leaves earlier the next day, so metering cannot change the demand it serves.

Point queues and exact counts. A ramp queue is a number against a storage, the arterial's own traffic is not modeled, the counters miss nothing and the detector's only fault is its noise or a freeze.

Obedient speed limits. Drivers hold the limit exactly, and a held cell is exempt from the drop, the most optimistic assumption in the model.

Teaching values. Every threshold is a teaching value. The application states that real capacities, critical occupancies, storages and queue limits come from the operating agency's own detector data and design standards, and it names the Highway Capacity Manual and the Ramp Management and Control Handbook of the Federal Highway Administration as such sources.

8.4 The Meters as Implemented

The controller reads the minute reports of Section 8.3, the period means of occupancy and the ramp counts, and never a density or a queue. The stations report and the meters act once per period $T = 1$ min, at the start of the step that begins on each whole minute. A priming call at 6:00 issues the first commands from the initial state, and from 6:01 on every report covers exactly the 30 integration steps of the minute just ended. Table 8.4 lists the settings.

8.4.1 ALINEA on the Shared Controller

ALINEA raises the release when the merge occupancy is below its setpoint and lowers it when it is above, as shown in Equation (8.9).

$$r(k) = \hat{r}(k-1) + K_R T \left(\hat{o} - o(k)\right) - K_P \left(o(k) - o(k-1)\right) \tag{8.9}$$

clipped to $[r_{min}, r_{max}] = [240, 1800]$ veh/h, with the occupancies in percent and $K_P = 0$ by default, while $K_P > 0$ gives PI-ALINEA [13]. The previous rate $\hat{r}(k-1)$ is the release the stop-line counter measured, clipped to the same range, not the last command, because with no queue waiting the ramp releases only what arrives, and a law that started from its own command would wind up far above it. The law runs on the velocity-form PID of Section 1.6.2, executed once per report, as shown in

Equation (8.10). It is Equations (1.23) to (1.26) with no derivative term and no bias, with s_k for S_k, c_k for v_k, the report period T for the control period and K_i for K_c/τ_I.

$$\begin{aligned} e_k &= \hat{o} - o(k) \\ s_k &= s_{k-1} + K_c\,(e_k - e_{k-1}) \\ c_k &= s_k + I_{k-1} \\ u_k &= \min\left(r_{max},\ \max(r_{min},\ c_k)\right) \\ I_k &= I_{k-1} + K_i\, e_k\, T + (u_k - c_k)\, T/T_t \end{aligned} \tag{8.10}$$

where e_k is the error in percent, s_k accumulates the proportional increments without limit, c_k is the demanded rate, u_k the emitted rate, I_k the integral, which also carries the initial rate because the bias is zero, and $T_t = T$ the tracking time of the back-calculation. The integral used at report k was accumulated through report $k-1$, so the present error rides on the proportional gain. ALINEA sets $K_c = K_P + K_R T$ and $K_i = K_R$, and before each report it shifts I_{k-1} by $\hat{r}(k-1) - u_{k-1}$, the difference between the counted release and the loop's last output. Away from the limits Equation (8.10) is then Equation (8.9) exactly, which the application's checks confirm at reporting periods of 1 and 2 min. The application passes $T_t = T$ rather than the default of Section 1.6.2, the integral time $K_c/K_i = T + K_P/K_R$, so at a limit the back-calculation absorbs the whole clip in one period and the two laws agree for any K_P. PI-ALINEA with $K_P = 50$, driven to its floor by two fabricated reports at 30 %, answers a report back at the setpoint with 965 veh/h, as its law gives, and the checks confirm it. There is no derivative term and no rate limit on the release, whose only limits are the clip to [240, 1800] veh/h, and the loop raises the release when the occupancy is below its setpoint, the direction the application's sign checks confirm.

In free flow a point of merge occupancy is 369 veh/h of discharge, so each report moves the occupancy by about $K_R/369 = 0.19$ of its error at the default gain. In a model run on the average day's plateau, with the detector noise at zero, the demand fluctuation removed (which no setting of the application does) and a queue on the ramp, a setpoint step from 15.5 to 15.0 % at 7:52 decays by a factor of 0.791 per report against the 0.810 that the static gain predicts, a time constant of 4.3 min, and by 0.901 against 0.905 at $K_R = 35$. The loop is integral on a plant whose gain near the setpoint is nearly static, so in steady flow the occupancy settles at $\hat{o}$ whatever the demand, and the meter finds the merge's room without knowing its capacity.

8.4.2 Queue Override and Coordination

The override releases at least what keeps the counted queue inside its limit at the next report, whatever the occupancy says, as shown in Equation (8.11).

$$r(k) \geq \hat{d}(k) - \left(q_{lim}\, q_{max} - \hat{q}(k)\right) \frac{60}{T} \tag{8.11}$$

with $q_{lim} = 85\,\%$, 51 of the 60 vehicles of storage, and T in minutes. When it binds, the integral is shifted to the rate applied. The upstream meter runs its own ALINEA on cell m_1 whenever the merge meter is metering, with the same override on its storage of 150. With coordination, which the application calls HERO-style after the linked strategy of [17], the merge ramp's counted queue asks the upstream ramp to store, as shown in Equation (8.12).

$$r_1(k) \leq \hat{d}_1(k) - \left(q_1^* - \hat{q}_1(k)\right) \frac{60}{\tau_c} \tag{8.12}$$

where $\tau_c = \max(T, 5\,\text{min})$ and the target q_1^* is the upstream ramp's override limit, 127.5 vehicles, from the report at which $\hat{q}$ reaches 60 % of its storage until it falls below 30 %. After that the target falls by 2 % of the upstream storage, 3 vehicles, a minute, so the held vehicles reach the merge as a trickle. The upstream ramp's own override still wins.

8.4.3 Speed Limits, Fixed Plan and Demand-Capacity

The speed limit over cells 6 and 7 is a third integral loop on the merge occupancy, $v(k) = v(k-1) + K_V T(\hat{o}_V - o(k))$, between 20 and 80 km/h and changing by at most 10 km/h a minute, on the same shared PID with $K_c = K_V T$ and $K_i = K_V$. Its tracking time is T, so the integral absorbs the clip and the rate limit in one period and $v(k-1)$ is the limit last applied. Its setpoint is its own. Below the meter's it takes over from the meter, and above it, it acts only once the meter can no longer hold the merge. The fixed-time plan replays a forecast of the average day, in quarter hours, as shown in Equation (8.13).

$$r_{plan}(t) = 60\,\mathrm{round}\Big(\frac{0.97\lambda Q - P_m(t_s)}{60}\Big) \tag{8.13}$$

clipped to $[r_{min}, r_{max}]$, where t_s is the middle of the quarter hour, which gives 1800 veh/h at 6:00, 900 veh/h at 7:45 and 1800 again from 9:15. The demand-capacity rule releases $\hat{C} - \hat{q}_{up}$, the capacity estimate less the counted mainline flow into the merge cell, while the occupancy is below the critical 16.25 %, and r_{min} above it. A watchdog, when it is on, treats a reading that three reports in a row have repeated as a failed detector and falls back to the fixed plan.

In manual mode each actuator takes its station, r_{man}, the upstream release and v_{man}, while every PID tracks the applied value, so that switching back to automatic moves no actuator, the bumpless transfer of Section 1.4.10. The simulation model is Equations (8.1) to (8.8). Nothing in the meters is identified from data. The classical controllers' internal models are the plan's forecast of the average day, the capacity estimate $\hat{C}$ of 5800 veh/h, the critical occupancy of the nominal diagram in the demand-capacity rule, and the counted queues. ALINEA uses no model at all beyond the sign of its gain.

8.4.4 Predictive Metering With Counted Inventories

The predictive option follows the arrangement of Section 1.6.3, and this subsection gives what is specific to the merge. Predictive traffic coordination [19] motivates two nominal cells, $L_a = 2$ km and $L_b = 0.5$ km, with counted queues. Existing detectors and counters supply occupancy and flows. Nominal $v_f = 80$ km/h, $Q = 2000$ veh/h per lane, $w = 16$ km/h and $\lambda = 3$ stay fixed. Occupancy gives k_a, k_b through the nominal effective vehicle length. Their balances are

$$\begin{aligned} 60\lambda L_a \dot{k}_a &= d_0 + r_1 - F, \quad 60\dot{q}_1 = d_1 - r_1 \\ 60\lambda L_b \dot{k}_b &= F + r - S(k_b), \quad 60\dot{q} = d - r \\ F &= \min(S(k_a), R(k_b)) \\ S(k) &= \lambda \min(v_f k, Q) \\ R(k) &= \lambda \max(0, \min(Q, w(k_{jam} - k))). \end{aligned} \tag{8.14}$$

Time is in minutes, flows in veh/h. Background d_0 subtracts counted upstream release from mainline flow into the merge, clipped nonnegative despite their travel-time mismatch. Priming uses nominal sending at measured upstream occupancy less public average ramp demand. Normalized states and inputs are $x = (k_a/k_c, k_b/k_c, q_1/100, q/100)$ and $u = (r_1, r)/1000$. Central differences of width 10^{-5} at the measured state and exact zero-order-hold discretization give

$$x_{j+1} = A_d x_j + B_d u_{b(j)} + E_d w_j + c_d + (b_a, b_b, 0, 0)^{\mathsf{T}}, \tag{8.15}$$

where $w_j = (d_0, d_1, d)$, c_d retains the affine remainder and $b(j)$ indexes a move block. Nominal lags are 1.5/0.375 min, without explicit delay or data fitting. Analytic discretization agrees within 5.1×10^{-12}. Noise-free one-report reconstruction using realized counts for evaluation gives merge RMSE 0.03257/2.73173/0.03839/5.55719/2.01376 percentage points for ALINEA/override/coordination/incident/frozen detector. This does not measure forecast accuracy.

Density reconstruction innovations e_i are clipped to ± 0.5 normalized, then $b_i^+ = b_i + (1 - e^{-T/5})(e_i - b_i)$ at every report, including classical control and Manual. Queues remain counted balances. Bias correction addresses offset [20], without Kalman filtering or guaranteed offset removal.

Optimization begins at priming. The detector period T_{det} (minutes) sets $T = \max(1, T_{det})$, nominally 1 min. Thirty intervals in blocks $(1,1,1,1,2,2,3,4,6,9)$ give 20 release decisions and three horizon-wide slacks. The initial release pair is selected [21] from Equation (8.16).

$$\begin{aligned} \min \quad & \sum_{j=1}^{30} \ell^{\mathsf{T}} x_j + \sum_{h=1}^{10} \|\Delta u_h\|_2^2 + s^{\mathsf{T}} W s + v^{\mathsf{T}} s + \tfrac{10^{-8}}{2} \|z\|_2^2 \\ \text{s.t.} \quad & \text{Equation (8.15)}, \quad x_j \geq 0, \quad s \geq 0 \\ & r_{min}/1000 \leq u_{h,i} \leq r_{max}/1000 \\ & |\Delta u_{h,i}| \leq 1.8, \quad x_{j,2} \leq 0.94 + s_o \\ & x_{j,3} \leq 0.85 q_{1,max}/100 + s_1 \\ & x_{j,4} \leq 0.85 q_{max}/100 + s_2. \end{aligned} \tag{8.16}$$

Here $\ell = T(2.5, 0.625, 100/60, 100/60)$ converts predicted inventory to vehicle-hours, $s = (s_o, s_1, s_2)$, $W = \text{diag}(20000, 5000, 5000)$, $v = (100, 20, 20)$ and z contains all releases and slacks. Penalty weights price the normalized variables in vehicle-hours. Move differences start from counted releases clipped to the command box. The QP adds the ridge shown (excluded from reported J), with 600 iterations allowed. Closed upstream input is fixed at zero. Current command limits set the hard release boxes. Cell/queue nonnegativity is hard. Queue rows cumulatively constrain releases by forecast arrivals, while the 15.275 % occupancy and 85 % current-storage ceilings are soft. The 1.8 change cap spans the ordinary release range and imposes no physical slew restriction. The reduced inventory cost omits most road cells and the origin queue, so it differs from scored TTS.

Preview On adds the public profile's change from now to each midpoint to measured demand. Error scales the future profile before subtracting its present average, up to ± 20 %. Off holds measured counts. Events and seeded fluctuations remain unknown. The fixed plan uses that public profile in quarter-hour slots. ALINEA/demand-capacity lack preview, making adjacent Off rows the present-information comparison.

A requested allowance differs from available flow in Equation (8.5). With a 600 veh/h floor and about 380 veh/h upstream arrivals, hard inventories make 11/11 attempts infeasible over 10 min. Invalid status, numbers, shapes, inventories or future bounds reject a plan. First failure holds commands for one report, then tracked ALINEA acts. The watchdog immediately selects the fixed plan. Protections retain priority and valid plans recover control. Inactive/watchdog-skipped attempt fields are zero, with observer history/cost current.

8.5 Ten Mornings and a Challenge

Table 8.5 lists the cases. All of them start from the defaults of Table 8.4 on the average Tuesday, with the changes that define a case written in its row. Every number below was measured headlessly, on the engine with no interface stepped as the browser steps it, with any change made at 6:00 before the run starts, at the default seed and over the whole morning from 6:00 to 10:00 unless another window is named, and the main results are collected in Table 8.7, three of whose columns Figure 8.2 draws. A setting chosen before the run and the same setting carried through a Reset give one run. The detector noise comes from the default seed of Section 1.6.1, so a case replays exactly, and where ten other noise seeds change a comparison the text says so. At its fastest speed the application plays a whole morning in 0.8 s.

Table 8.6 gives each case under the name the application shows, with the requirement for completing it under the rules of Section 1.6.4. A change that a second morning asks for must be in place

Table 8.5 The Ten Cases and the Challenge

#	Change from the defaults	Teaching point
1	signal dark	the breakdown and its drop
2	fixed-time plan, five weekdays	a forecast, right on average
3	the defaults, ALINEA at 15.5 %	the margin and its queue
4	$\hat{o} = 16.5\,\%$	a setpoint past the fold
5	queue override at 85 %	storage is the budget
6	incident at 7:40, 20 min	the honest infeasible case
7	override, detector freezes at 7:20	integral action on a frozen reading
8	override, upstream ramp open, coordination	borrowed storage
9	signal dark, speed limits at 15.2 %	the mainline as storage
10	predictive metering against ALINEA, both ramps	model and forecast error
11	demand +1 %, both ramps, fixed plan	graded morning

Table 8.6 Scenarios and Completion Requirements

#	Scenario	Kind	Required to complete
1	No metering: the breakdown and the shockwave	Demonstration	Run the dark-signal morning through the breakdown of the merge to the 10:00 end.
2	Fixed-time metering, day by day	Experiment	Run the fixed-time plan to 10:00 on Tuesday, then a second morning to 10:00 on Thursday.
3	ALINEA just under the critical occupancy	Experiment	Run ALINEA at $\hat{o} = 15.5\,\%$ to 10:00, then a second morning to 10:00 at $\hat{o} = 15.0\,\%$.
4	The setpoint a little too high	Demonstration	Run to 10:00 with $\hat{o}$ of at least 16.4 % and two or more breakdowns, then a second morning to 10:00 at $\hat{o} = 16.0\,\%$.
5	The queue override	Demonstration	Run to 10:00 with the override on, a merge ramp storage of at most 60 vehicles and the merge broken down, then a second morning to 10:00 with a storage of 115 to 125 vehicles.
6	An incident past the merge	Demonstration	Run to 10:00 with ALINEA and no override, through the breakdown the incident causes, then a second morning to 10:00 with the override on.
7	A stale detector	Demonstration	Run past 8:00 without the watchdog until the merge breaks down, then a second morning past 8:00 with the watchdog on before 7:19, so that the meter falls back to the fixed plan after the 7:20 freeze.
8	Two ramps and coordination	Experiment	Run to 10:00 with coordination on, then a second morning to 10:00 with coordination off.
9	Speed limits against metering	Experiment	Run to 10:00 with the speed limits at $\hat{o}_V = 15.2\,\%$, then a second morning to 10:00 with $\hat{o}_V$ between 14.5 and 15.0 %.
10	MPC versus feedback on the weekday forecast	Experiment	Run the predictive controller to 10:00 with both ramps open, then repeat with ALINEA after Reset.
11	Challenge, a busier Tuesday with both ramps open	Challenge	By 10:00, with 1 % more demand and both ramps open as given, a total time spent below 1650 veh·h, neither ramp queue ever past its storage, and at most 3 min in the capacity drop.[a]

[a] As shipped (fixed plan at the merge ramp, ALINEA upstream) the morning fails all three parts, with 1833.4 veh·h, 103.2 min in the drop and queues of 66.9 of 60 and 187.8 of 150 vehicles. The suggested ALINEA with the override and coordination passes with 1551.9 veh·h (98.1 veh·h below the limit), no breakdown and queues of 51.2 of 60 and 127.4 of 150.

within its first 10 min. The challenge is judged on the case as given, so a changed model parameter or disturbance voids the attempt, while the strategies, the override, coordination and the speed limits are the reader's to choose. The footnote of the table gives its outcome as shipped and under the suggested configuration.

Table 8.7 Measured Results Over the Morning, 6:00 to 10:00. Queue Is the Worst Merge Ramp Queue Against Its Storage of 60

Case and configuration	TTS, veh·h	Breaks	Drop, min	Queue
1, no metering	1840.7	7:48	104.9	0
2, fixed plan, Tuesday	1438.5	—	0	59.3
2, fixed plan, Thursday	2667.7	7:22	158.3	82.2
3, ALINEA at 15.5 %	1529.4	—	0	163.9
3, $\hat{o}$ 15.0 %	1835.7	—	0	365.2
3, $\hat{o}$ 16.0 %	1425.9	—	0	50.2
3, demand-capacity	1487.9	—	0	125.5
4, $\hat{o}$ 16.5 %	1731.5	4 times	40.4	341.3
4, $\hat{o}$ 16.2 %	1597.0	twice	19.3	257.7
5, override at 51	1766.0	7:57	89.1	51.2
5, override, storage 120	1481.0	—	0	102.2
6, incident, ALINEA	3836.5	7:41	102.9	1056.7
6, incident, no metering	3846.5	7:40	139.5	79.4
7, frozen detector	1840.7	7:48	104.9	0.2
7, with the watchdog	1434.7	—	0	51.1
8, coordination	1459.9	—	0	51.1
8, without coordination	1716.2	7:57	88.9	51.2
9, speed limits at 15.2 %	1511.0	—	0	0
9, speed limits at 14.5 %	2095.6	—	0	0

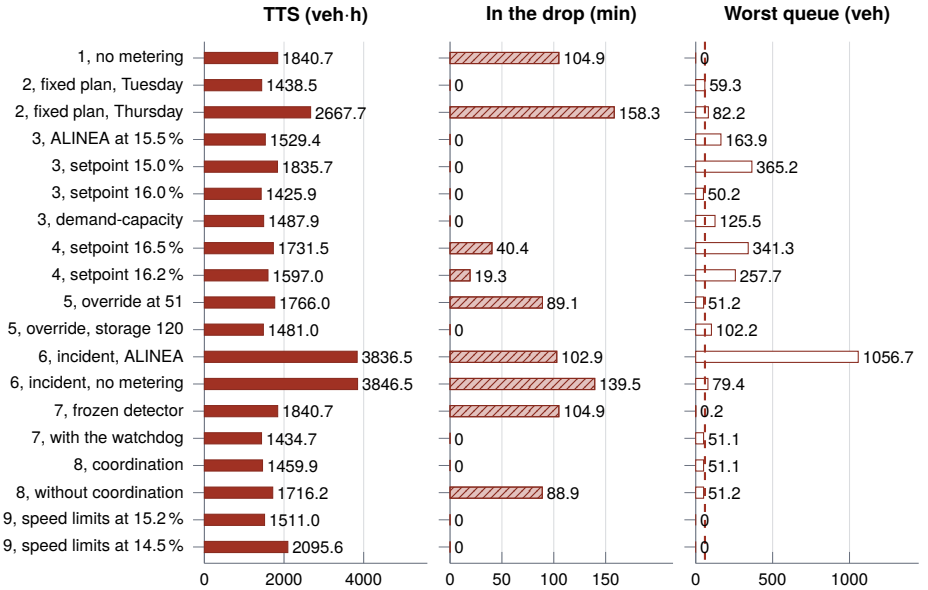


Figure 8.2 Three columns of Table 8.7 over the whole morning: total time spent (solid bars), minutes in the capacity drop (hatched) and the worst merge ramp queue (open), with the storage of 60 vehicles dashed. Notice that the meters that keep the merge out of the drop hold a queue on the ramp instead, that the speed limits of case 9 store the delay on the mainline with no queue at all, and that the incident of case 6 costs more than any other morning whatever the meter does.

8.5.1 The Breakdown and the Drop That Holds It

With the signal dark, the merge passes 5815 veh/h on average from 7:37 to 7:46, at most 5890, and 5950 at 7:47. At 7:48 its density passes 25 veh/km per lane and it latches, and from then on it discharges 5400 veh/h, as shown in Figure 8.3(b). The queue grows for as long as the demand exceeds 5400, which it does until 8:40, and its tail reaches 3.5 km behind the merge at 8:44. The

merge regains its capacity only at 9:33, when its density falls below 22 veh/km per lane, so 17 min of overload cost 104.9 min in the drop. On the diagram the merge cell leaves the free branch at its peak, holds the density of the queue behind it, up to 59.1 veh/km per lane, while it sends the dropped 1800 veh/h per lane, and returns to the free branch only below the recovery density, as shown in Figure 8.3(a). The merge cell sits off the triangle there because a cell's outflow in the model is its sending flow, and the head of a queue sends its dropped capacity.

> **TRY IT IN THE APP** **1 · No Metering: The Breakdown and the Shockwave**
>
> Load the case and press Run. Watch the road at the merge, at the 4 km mark: a little before 8:00 it turns from pale to red as the merge breaks down. Keep the run going to 10:00 and read the Capacity drop tile. The merge stayed in the drop for 104.9 min, although the demand was above capacity for only 17.

The queue's tail is a shock. From 7:58 to 8:10 the free flow arriving at it carries 1634 veh/h per lane at 21.0 veh/km per lane and the queue just inside it passes 1459 at 57.1 veh/km, and the jump condition, the difference of the flows over the difference of the densities, gives 4.8 km/h upstream against the 5.0 km/h that the time–space record shows between 1 and 2 km behind the merge. The congestion wave speed of 16 km/h is the speed of changes inside the queue, not of its tail. The morning costs 1840.7 veh·h.

Pause. Try Exercise 8.1 before reading on.

8.5.2 A Plan Built for the Average Day

The fixed-time plan releases, in each quarter hour, what should fill the merge to 97 % of its capacity on the average day. On the Tuesday it was built for it holds the merge, with a queue of 59.3 at the worst, and the morning costs 1438.5 veh·h against 1840.7. On every other day its forecast misses. Wednesday and Thursday come earlier and heavier, the merge breaks down at 7:33 and 7:22, two minutes after and in the same minute as with no meter, and on both days the queue puts 22 vehicles on the arterial, so Wednesday saves 37.0 veh·h and Thursday nothing. On Monday and Friday the merge would have flowed anyway and the plan only holds vehicles, 44.9 and 30.9 at the worst, for 25.7 veh·h and 15.5 veh·h more than no meter. Over the five days the mean falls from 1899.5 to 1819.9 veh·h, 4.2 %, and the week's net saving of 398.0 veh·h is less than the 402.2 that Tuesday saves alone. ALINEA on the same five days averages 1579.0 veh·h, because a meter that measures the merge does not need to know which morning it is.

> **DID YOU KNOW?** **The Plan That Wins on Its Own Day**
>
> From the simulator: on the Tuesday it was built for, the fixed-time plan costs 1438.5 vehicle-hours, less than the 1529.4 of ALINEA at its default setpoint. On Thursday, which comes earlier and heavier, the same plan lets the merge break down at 7:22, and the morning costs 2667.7 (Table 8.7).

8.5.3 Integral Control Under the Fold

ALINEA at 15.5 % never lets the merge break down. From 7:30 to 8:30 the merge passes 5719 veh/h on average at a true occupancy of 15.49 %, against the 5400 a broken-down merge discharges, and the release bottoms at 655 veh/h at 7:55. By 9:01 the metered merge has passed 321 more vehicles than the unmetered one, and by 10:00 both have passed the same 20 706, so what metering saves is time, 311.3 veh·h of it. The cost is the ramp queue, which passes its storage of 60 at 7:57, peaks at 163.9 vehicles at 8:27 with 103.9 of them on the arterial, and is empty only at 9:02, as shown in Figure 8.3(c). The margin cuts both ways. A setpoint of 15.0 % holds back vehicles the merge had room for, the queue reaches 365.2 and the morning costs 1835.7 veh·h, nearly what no meter costs, while 16.0 % keeps the queue at 50.2 for 1425.9 veh·h without a breakdown this Tuesday. That last margin is thinner than it looks. On each of ten other detector-noise seeds 16.0 % breaks the merge down for

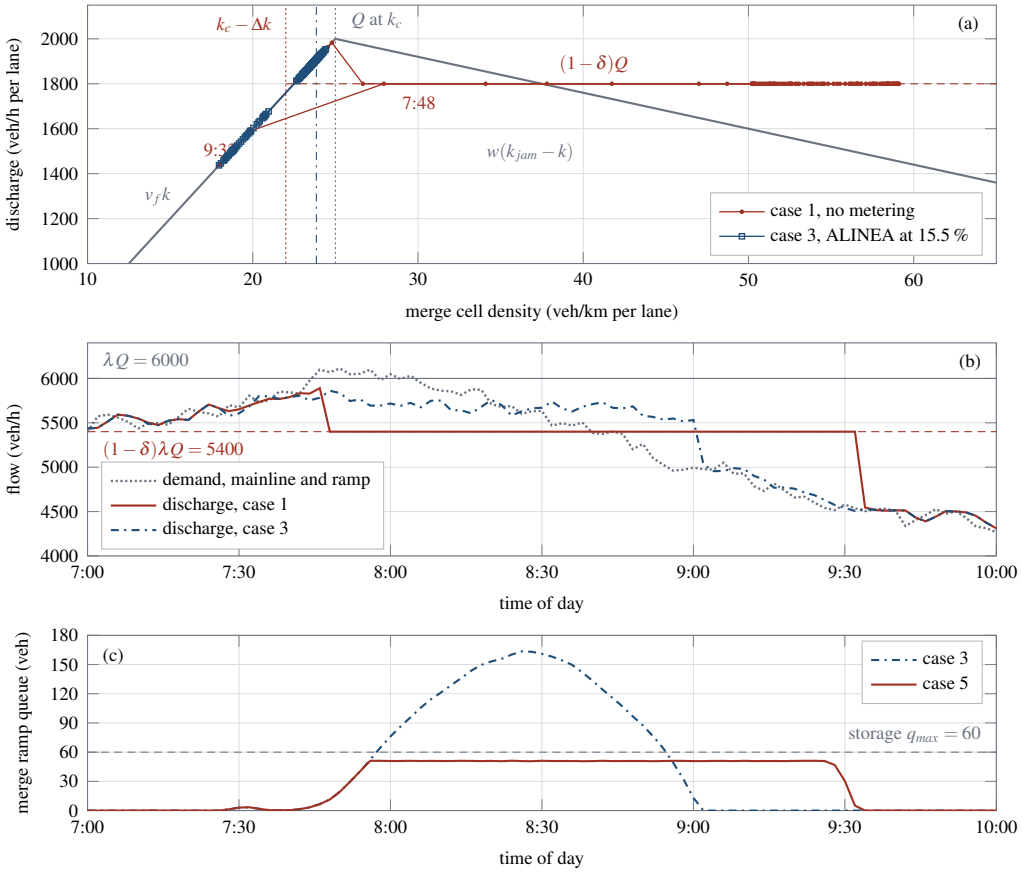


Figure 8.3 Breakdown and its prevention on the average Tuesday, from the engine. (a) The merge cell's density against its discharge per lane once a minute from 7:00 to 10:00, case 1 with filled circles and case 3 with open squares, and the setpoint's 23.85 veh/km as the dash-dotted vertical line. (b) The merge discharge of case 1 (solid) and case 3 (dash-dotted) against the demand of mainline and merge ramp (dotted), every 2 min. (c) The merge ramp queue that ALINEA costs, without the override (case 3, dash-dotted) and with it (case 5, solid).

12.7 to 24.1 min, while 15.5 % never does. The gain matters less than the setpoint, 1526.6 veh·h at $K_R = 35$ and 1537.8 at 140, and PI-ALINEA with $K_P = 50$ gives 1530.1. The demand-capacity rule with its 5800 veh/h estimate gives 1487.9 veh·h with a queue of 125.5, right for as long as its estimate is.

8.5.4 A Setpoint Past the Fold

At 16.5 %, a quarter point past the critical occupancy, the setpoint asks for 25.4 veh/km per lane, a density at which the merge cannot stay. It breaks down at 7:48, the true occupancy reaches 25.4 %, and the integral action drives the release to its 240 veh/h floor from 7:50 to 8:01. The merge regains its capacity at 8:11, once its density is below 22, and the integral action, still aiming past the fold, pushes it over again, four times in all, from 7:48 to 8:11, 8:20 to 8:26, 8:39 to 8:45 and 9:03 to 9:08, 40.4 min in the drop. Meanwhile the meter holds the ramp hard, the queue reaches 341.3, 281.3 of them on the arterial, and the morning costs 1731.5 veh·h. Ten other detector-noise seeds give three or four breakdowns. At 16.2 %, just below the critical occupancy, the merge still breaks down twice, for 19.3 min. An integral loop aimed past the peak of its map has no steady state to settle in, because

each recovery puts the merge back on the free branch, from which the integral steers it over the edge again. Volume 7, Chapter 4 treats a loop whose setpoint sits at the peak itself, where the process gain passes through zero and reverses.

> **TRY IT IN THE APP** **4 · The Setpoint a Little Too High**
>
> Load the case, whose setpoint of 16.5 % lies a quarter point past the critical occupancy, and run it to 10:00 on the Diagram view. Watch the merge point climb toward the setpoint and fall off the fold, four times in all. Then press Reset, set the setpoint to 16.0 % and run again: on this Tuesday the merge never breaks down.

8.5.5 Storage Is the Budget

The queue override keeps every vehicle off the arterial. The queue reaches its limit of 51 at 7:56, the override releases what arrives, and the merge breaks down at 7:57, nine minutes later than with no meter, as Table 8.7 lists and Figure 8.4 shows. It stays in the drop until 9:26, 89.1 min, and the morning costs 1766.0 veh·h, which gives back 236.6 of the 311.3 veh·h that metering saved, 76.0 %. A meter moves delay in time by storing it, and when the storage runs out the breakdown it postponed arrives. Volume 4, Chapter 14 meets the same budget as a vault of reserves with a floor. With the limit at the full storage the breakdown comes at 7:58. On this seed the storage needed lies between 105 and 110 vehicles, since with 105 the merge breaks down at 8:04 and with 110 it never does, and with 120 the queue reaches its new limit of 102 only at 8:07, the true occupancy peaks at 16.23 % against the critical 16.25, and the morning costs 1481.0 veh·h. The threshold moves with the detector noise. Over ten other seeds a storage of 110 still breaks down on five and 120 on one.

Pause. Try Exercise 8.3 before reading on.

8.5.6 An Incident No Meter Can Absorb

At 7:40 a crash at 5.0 km halves cell 10 for 20 min. The merge had been passing 5741 veh/h in the ten minutes before, the blocked cell latches and discharges 2700, and every strategy breaks down between 7:40:30 and 7:40:36, no metering, the fixed plan, ALINEA with and without the override, demand-capacity and speed limits at 15.2 %. From 7:45 to 7:58 the merge discharges 3038 veh/h on average, and at 8:00 the section holds 1470 vehicles whatever the meter has done. No setting of the meter can keep the merge flowing upstream of a cell that passes less than the mainline alone brings, because the queue is made of mainline traffic. What differs is the recovery. ALINEA without the override holds its ramp at the 240 veh/h floor from 7:43 to 9:23 and gives the merge to the mainline queue, so the merge regains its capacity at 9:23 after 102.9 min, while with no meter it is still in the drop at 10:00, 139.5 min on. It does so by holding up to 1056.7 vehicles at 9:26, 996.7 of them on the arterial, and the total time spent hardly moves, 3836.5 veh·h against 3846.5, every strategy within 0.56 % of the others. The speed limits recover first, at 8:29, but fall back into the drop seven more times, 61.9 min in all, and the demand-capacity rule recovers at 9:23 and falls back eight more times. Over ten other noise seeds ALINEA's morning costs 3834.7 to 3852.9 veh·h and the merge recovers at 9:23 on every one. The freeway recovers sooner and the morning does not.

8.5.7 A Frozen Detector

In case 7 the merge detector freezes at 7:20 at a reading of 14.86 %, 0.64 % below the setpoint. The frozen error is added every minute, 44.9 veh/h on top of whatever the stop line counted, so the release rides just above the arrivals, 1100 against 1063 veh/h at 7:49, the queue never holds more than 0.2 vehicles, and the meter has switched itself off with nothing on its card to show it. The merge breaks down at 7:48 and the morning costs 1840.7 veh·h, within 0.01 veh·h of no meter. The watchdog trips at the 7:22 report, the third to repeat the frozen reading, falls back to the fixed plan, which holds this Tuesday, and the morning costs 1434.7 veh·h. The frozen value is one draw of the detector noise. Over ten other seeds it lies between 14.75 and 15.69 %, and the merge breaks down on every one. Only the reading above the setpoint, 15.69 %, lets a queue form, which the override holds at its limit



Figure 8.4 Case 5, The queue override, at 10:00, where the application paused itself at the end of the morning, in the Time–space view with the 4.0 h trend window. The figure is a composite: the density map and the ramp-queue strip above the trends, with three tiles of the Learn panel moved into the empty band at the right. (1) The density map reads the breakdown as a wedge growing back from the merge at 4 km, and (2) its tail is labeled "queue tail 4 km/h upstream". (3) The strip beneath holds the merge ramp queue against its dashed storage of 60. (4) In the trends a dashed vertical line marks the breakdown, which the run records at 7:57 (1.94 h). Before it the reported merge occupancy scatters, by the run's record from 14.3 to 16.6 %, about its 15.5 % setpoint from 7:00. By the same record it stays above 30 % in the reports from 8:00 to 9:23 and falls back below the setpoint after the capacity returns at 9:26. (5) The tiles give 89.1 min in the drop, the queue held at its limit of 51, and the total time spent of 1766 veh·h, which the last trend shows with 85 on the ramps.

of 51 while the morning costs 1966.2 veh·h. The watchdog holds the merge on all ten. A freeze fired at 7:40 on the settings of case 5, which are those of case 7 without its scripted freeze, holds a reading of 15.29 %, breaks the merge down at 7:48 without the watchdog, 1841.1 veh·h, and costs 1431.7 with it.

🕮 **Pause.** Try Exercise 8.2 before reading on.

8.5.8 Borrowed Storage

Case 8 opens the upstream ramp with its storage of 150 and turns coordination on. The merge ramp's counted queue passes 36, 60 % of its storage, at the 7:54 report and the upstream meter starts to store, its queue peaks at 123.7 at 8:21 and is empty again at 9:04, the merge ramp never holds more than its limit of 51, the merge never breaks down, and the morning costs 1459.9 veh·h. Without coordination the upstream meter holds nothing until the jam reaches it, the merge ramp's override gives way, and the merge breaks down at 7:57 for 88.9 min and 1716.2 veh·h. Without the override the merge ramp

Table 8.8 The Graded Morning in Seven Configurations

Configuration	TTS	Drop	Spill, veh	Met
as shipped, fixed plan	1833.4	103.2	6.9, 37.8	no
ALINEA	1569.9	0	161.9, 0	no
ALINEA, override	1894.7	102.9	0, 0	no
ALINEA, coordination	1619.6	0	36.7, 0	no
ALINEA, override, coordination	1551.9	0	0, 0	yes
demand-capacity, both	1534.9	0	0, 0	yes
speed limits at 15.2 %	1550.9	0	0, 0	yes

TTS in veh·h, drop in minutes, spill as the most vehicles past the storage of the merge and upstream ramps.

spills 13.6 vehicles onto the arterial for 1516.3 veh·h. Coordination adds storage and not capacity. On Thursday both ramps fill and the merge breaks down at 7:53 anyway, for 2262.6 veh·h.

8.5.9 The Mainline as Storage

In case 9 the signal is dark and the speed limits aim at 15.2 %. They first come down at 7:11, stay below 80 km/h from 7:25 until 9:15 and reach 28.2 km/h at 8:23. The merge never breaks down, no vehicle waits on the ramp, and the morning costs 1511.0 veh·h, slightly less than ALINEA's 1529.4, with the delay spread over every mainline driver instead of concentrated on the ramp. That ranking lies within the noise. On seven of ten other detector-noise seeds the limits let the merge break down for 3.7 to 4.5 min and the morning costs up to 1641.5 veh·h, while ALINEA never breaks down and costs 1517.8 to 1559.2. The window is narrow. At 15.5 % the merge breaks down for 5.1 min, at 15.0 % the morning costs 1608.2 veh·h, at 14.7 % 1860.9 and at 14.5 % 2095.6, the last two more than no control. A setpoint below 14.625 % holds a steady merge under the 5400 a broken-down merge passes. The limits also act early, from 6:59 at 14.5 %, 49 min before an unmetered merge breaks down, and by 9:30 that merge has passed 183 fewer vehicles than with no control. The first setting that costs less than no control is 14.8 %, at 1762.9 veh·h, and it does so on all ten other seeds. With ALINEA and the override first and the limits at 16.0 % behind them, the limits act once the override has given up, 11.5 min in the drop instead of 89.1, and 1595.7 veh·h.

8.5.10 The Graded Morning

The challenge runs Tuesday with 1 % more traffic on every stream and both ramps open, and asks by 10:00 for a total time spent below 1650 veh·h, neither ramp queue ever past its storage, and at most 3 min in the drop. Table 8.8 compares the configurations. As shipped, with the fixed plan at the merge ramp and ALINEA at the upstream ramp, the merge breaks down at 7:53 for 103.2 min, both ramps spill, and the morning costs 1833.4 veh·h. Each piece fails alone. ALINEA alone holds the merge but puts up to 161.9 vehicles on the arterial, the override without coordination breaks the merge down at 7:52, and coordination without the override still spills 36.7. ALINEA with the override and coordination meets the target with 1551.9 veh·h, the merge ramp at 51.2 and the upstream ramp at 127.4, and it keeps meeting it with the demand scaled up by as much as 2 %, twice the challenge's, and fails at 2.5 %, while the shipped plan already fails at 0.5 %. Speed limits alone at 15.2 % also meet it, with 1550.9 veh·h and nobody on a ramp, but fail at 2 %. The detector noise separates the two answers. Over ten other noise seeds the three-part answer meets the target on all ten, with the demand scaled up by 1.5 % on nine and by 2 % on only two, while the speed limits alone meet it on one, because the merge breaks down for 3.1 to 4.1 min or the total passes 1650 veh·h on the other nine.

Table 8.9 Frozen-Setting Comparisons, 240 Minutes at Each Case's Seed and 2 s Physics Step

Case	Structure	TTS (veh·h)	Delay (veh·h)	Drop (min)	$q^{obs}_{max}/q^{obs}_{1,max}$ (veh)	IAE (%-min)	TV_r (veh/h)	Sat. (%)	Attempts/rejects	Mean QP (μs)
Case 5	ALINEA	1766.034	365.191	89.07	51.21/0.00	1999.8	6966	38.3	0/0	0
Case 5	Demand-capacity	1764.888	364.046	89.23	51.21/0.00	2004.1	5381	37.9	0/0	0
Case 5	Fixed plan	1434.689	33.847	0.00	51.08/0.00	403.1	1858	4.6	0/0	0
Case 5	MPC On	1845.881	445.039	99.77	51.21/0.00	2165.2	7006	40.8	240/23	584
Case 5	MPC Off	1851.596	450.753	100.40	51.21/0.00	2174.5	8312	37.9	240/17	312
Case 5	MPC, +20%	1843.509	442.667	101.40	51.16/0.00	2167.2	9700	37.5	240/4	362
Case 5	Fixed model rule	1864.211	463.369	98.23	51.21/0.00	2174.9	6925	42.5	0/0	0
Challenge	As shipped	1833.372	468.038	103.20	66.95/187.80	2263.5	1800	0.0	0/0	0
Challenge	ALINEA	1551.877	186.542	0.00	51.21/127.41	349.8	6590	6.7	0/0	0
Challenge	Demand-capacity	1534.883	169.549	0.00	51.09/127.33	355.0	10841	1.7	0/0	0
Challenge	Fixed plan	1833.372	468.038	103.20	51.08/127.61	2258.8	1903	6.7	0/0	0
Challenge	MPC On	1560.782	195.447	0.00	50.99/127.70	349.7	6392	1.3	240/2	486
Challenge	MPC Off	1769.882	404.547	67.97	51.16/127.70	1408.3	11046	23.3	240/4	361
Challenge	MPC, +20%	1950.138	584.803	99.47	51.21/127.58	2071.3	10352	36.3	240/0	453
Challenge	Fixed model rule	1588.991	223.656	0.00	51.21/127.67	327.1	6308	15.0	0/0	0
Two ramps	ALINEA	1459.917	108.100	0.00	51.07/123.65	379.9	6484	1.3	0/0	0
Two ramps	Demand-capacity	1475.952	124.135	0.00	47.68/116.47	380.4	8932	0.0	0/0	0
Two ramps	Fixed plan	1494.885	143.068	0.00	51.08/126.78	380.6	1858	4.6	0/0	0
Two ramps	MPC On	1469.389	117.572	0.00	29.47/120.62	376.4	6996	0.0	240/0	467
Two ramps	MPC Off	1446.587	94.771	0.00	10.92/116.26	381.7	6830	0.0	240/0	378
Two ramps	MPC, +20%	1834.909	483.092	90.23	51.21/127.61	1912.9	10455	33.8	240/0	469
Two ramps	Fixed model rule	1545.069	193.253	0.00	51.21/127.46	340.0	6780	7.5	0/0	0

The two-ramp row is an extra run that repeats case 8's physical conditions. TV and saturation are the shared merge-command metrics. IAE retains the ALINEA reference, not the MPC ceiling. Delay subtracts an arrival-matched free-travel reference from full-road TTS. Physical storage exceedance is zero except the shipped challenge (17.90/13.60 min, merge/upstream). Soft 85% bands can still be exceeded. Timing is host dependent.

8.5.11 Measured Predictive Comparison

Case 10 repeats case 8 under the predictive controller, then compares ALINEA after Reset. Runs share 240 min, seeds, 2 % noise, arrivals and protections, including the challenge's suggested controls. MPC fixes margin 0.94, move price 1 and zero forecast error throughout. One predetermined classical rule rounds $0.94(16.25)$ to 15.3 % at both meters. Nominal gain 0.00270833 percentage point/(veh/h) and 5 min correction give $K_R = 73.846$, rounded to 75 at both meters, with $K_P = 0$. No case-specific tuning follows.

The predictive controller costs 1469.389 veh·h, against 1459.917 veh·h for ALINEA and 1475.952 veh·h for the demand-capacity rule. With the forecast Off it costs 1446.587 veh·h, so the forecast adds 22.801 veh·h. A +20 % forecast error gives 1834.909 veh·h and 90.23 min drop. The fixed classical rule costs 1545.069 veh·h, also losing to shipped ALINEA. Figure 8.5 compares trajectories.

Case 5 MPC loses, 1845.881 veh·h against ALINEA's 1766.034 veh·h, with 99.77/89.07 min drop. The local cost omits the latch and protective command changes. Challenge MPC passes at 1560.782 veh·h, zero drop and 50.99/127.70-vehicle queues, losing to ALINEA/demand-capacity at 1551.877/1534.883 veh·h. Its soft storage band is exceeded by 0.198 vehicle for 2.43 min, despite passing the physical-storage grade.

Fault replicas add 300 veh/h ramp demand at minutes 90–120, persistent 15 % rain at minute 90, or detector freeze at minute 80, with watchdog enabled for every law. Event MPC/ALINEA/demand-capacity cost 2086.678/2132.295/2081.616 veh·h, and rain costs 4451.582/4467.444/4461.352 veh·h, all failing the challenge criteria. Frozen MPC costs 1495.001 veh·h, with 82 attempts, zero rejects and 158 min watchdog fallback. All 47 rows have paired half-step runs. Most critically, noisy challenge MPC-Off changes from 1769.882 veh·h, 67.97 min drop and fail at 2 s to 1493.373 veh·h, zero drop and pass at 1 s. Small report/control differences precede a discontinuous latch crossing. Neither this trace nor separate changed-input diagnoses isolate a sole cause or prove convergence. Removing

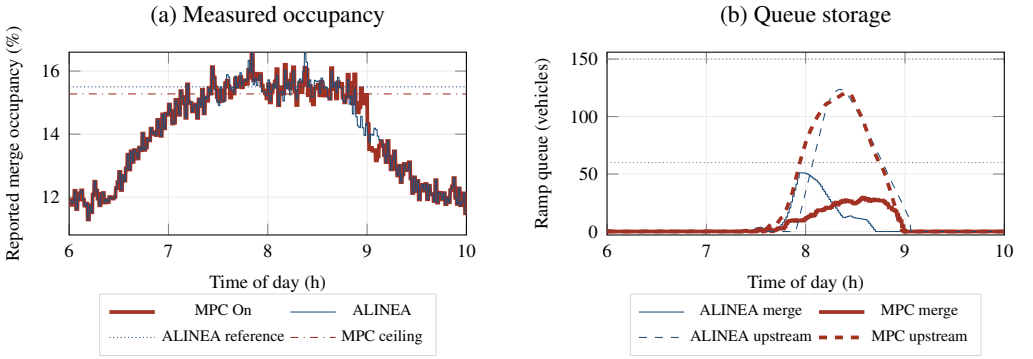


Figure 8.5 Two-ramp comparison from the original noisy runs. (a) Reported merge occupancy under ALINEA (thin line) and predictive metering with forecast On (thick line). The ALINEA reference is 15.5 % (dotted), while the predictive ceiling is 15.275 % (dash-dotted). (b) Counted merge queues (solid) and upstream queues (dashed) under the same laws, thin for ALINEA and thick for the predictive controller. Neither run enters the capacity drop or exceeds physical storage, yet their full-road TTS differs by 9.472 veh·h.

detector noise alone gives 1479.561/1479.596 veh·h at the two steps. Additionally suppressing demand fluctuation gives 1505.590/1505.584 veh·h. Both changed-input pairs pass without replacing the noisy result.

At one simulated minute per second, 20 warm Chrome decisions average 1730 μs for the complete controller (0.173% of one core). Solver/model/observer/decision means are 740/620/295/1395 μs. At 300 min/s, guarded and unguarded playback reach 298.24 and 295.37 min/s at 59.98 and 60.07 frames/s, with zero budget stops. These samples establish no pacing benefit. The 25 ms frame-work guard retains reached decisions and physics steps but cannot bound a single slow solve. Forced guard stops retain 240 solves and exact non-timing records. Same-Chrome speed/guard comparisons are byte-identical, while Node/Chrome differ in exported TTS by up to 10^{-5} veh·h.

8.6 Technology in Practice

The meters of Section 8.4 read a minute report of the merge occupancy and the counts at each ramp, and act through the release of a ramp signal. Figure 8.6 draws the merge loop with the devices a real ramp carries in place of the schematic's symbols.

What Measures It

The occupancy comes from inductive loops cut into each lane of the merge, the detectors with which ramp metering is mainly carried out [22]. Equation (8.6) stands for that loop: the period mean of the cell's density times a vehicle length plus a detection zone, with 2 % noise, and a freeze that repeats the last reading. Loops cannot measure a ramp queue directly, only estimate it [22], and the exact counters of the model are one of the assumptions of Section 8.3.7.

> **TECHNOLOGY** **Sensor: The Inductive Loop Detector**
>
> A loop is one or more turns of wire in a slot sawn into the pavement, wired to an electronics unit in the roadside cabinet that excites it at 10 to 200 kHz. A vehicle over the loop induces eddy currents that lower its inductance, and the unit senses a fall in frequency and signals a vehicle present. Loops are the common standard for accurate occupancy, while density and speed must be inferred [23]. Pavement wear, poor installation and weather degrade them. Some units flag a step of 25 % in inductance as a loop failure, and an open loop that calls constantly can be taken off line [23].

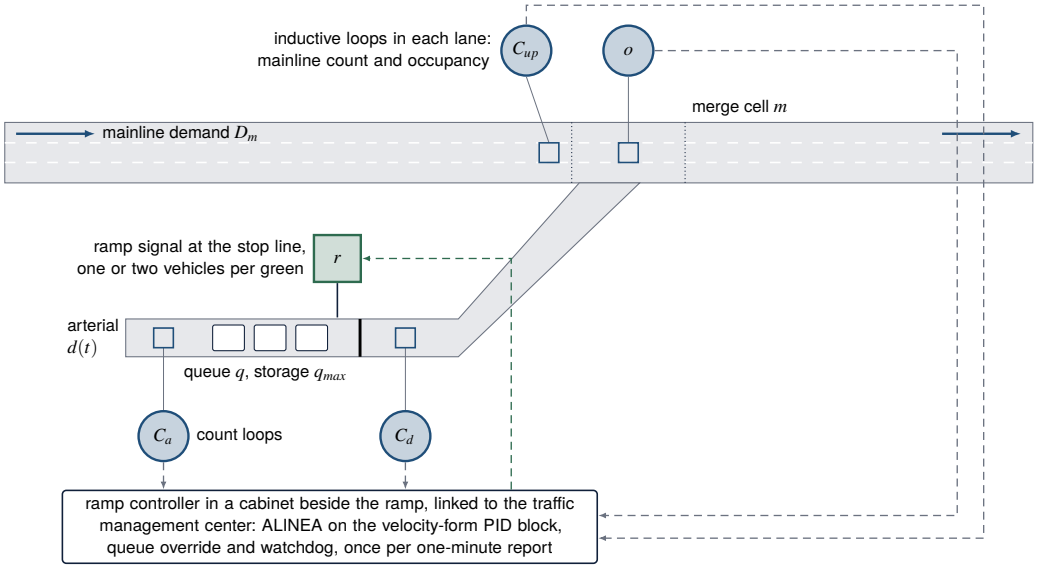


Figure 8.6 The merge meter with the devices of a real ramp in place of the schematic's symbols. Loops cut into the pavement give the merge occupancy o and the mainline count C_{up}, count loops on the ramp give C_a and C_d, and a controller in the roadside cabinet sets the release r of the ramp signal. Notice that the meter never measures the queue it is holding: it counts what enters and what leaves the ramp.

What Moves It

The final element is the ramp signal. The model's release is a continuous flow clipped to [240, 1800] veh/h, the floor being the lowest practical rate of a signal and the ceiling that of a two-lane meter. A real meter releases whole vehicles, one or two per green.

> **TECHNOLOGY** **Final Element: The Ramp Signal**
>
> The meter is a traffic signal at a stop line. With one vehicle per green, a demand detector before the stop line calls the green and a passage detector just beyond it ends it, and a single lane passes about 900 vehicles an hour. A minimum cycle of 4 s, 2.5 s of red and 1.5 s of green, gives 1000 an hour, and the lowest practical rate is 240 an hour, a 15 s cycle. For a ramp volume of 1600 to 1800 an hour the guidance is two metered lanes, each releasing two vehicles per green [6]. The release is therefore a train of single vehicles, and the cycle sets its floor and its ceiling.

Where the Controller Runs

The ramp controller sits in a cabinet beside the ramp with its load switches and loop amplifiers. It is a microprocessor-based traffic signal controller that also acts as a data station, and it reaches the traffic management center by telephone line, fiber, microwave or radio [6]. Field algorithms work on smoothed reports: running averages over 5 min, a one-minute occupancy per lane in one agency's local algorithm, and filtered 30 s data in a zone algorithm [6, 23]. The meters of this chapter act once per one-minute report. The detector handbook describes ALINEA as a recursive law that holds the mainline occupancy at a preselected value, with a rate that violates a limit adjusted to a feasible value [23]. In this chapter that recursion is the velocity-form block of Equation (8.10), and the adjustment is its clip with back-calculation.

> **IN PRACTICE** **A Fixed Plan Behind the Feedback**
>
> Traffic-responsive metering often does five to ten percent better than a pre-timed plan, yet the low cost of pre-timed systems makes them attractive backups when the primary approach fails. Queue storage is sized by rules of thumb, one agency's being ten percent of the peak-hour volume before metering. When the queue outgrows it, the rate is raised, which costs the freeway but protects the arterial, and the queue must not be dumped onto the freeway all at once. A queue detector near the surface street raises the rate, and a second one further up raises it sharply [6]. The watchdog of case 7 and the override of case 5 are these two practices.

What Is Modernizing It

Three developments touch this loop. Over-roadway sensors, video image processors and frequency-modulated microwave radar, report lane occupancy without a cut in the pavement, and radar of that kind also sees stopped vehicles [23]. Messages broadcast by connected vehicles add data from which a robust filter estimates the ramp queue, with a bound on its long-term error even when the share of connected vehicles is unknown [22]. Learning-based meters trained on loop data have been tested against false data injected into their inputs, because a richer input is also a larger surface for attack [24]. The frozen detector of case 7 is the plain form of that risk.

8.7 Limitations

The cases inherit the assumptions of Section 8.3.7. With a first-order model and a deterministic latch the merge breaks down at one density and never by chance, every wave inside a queue moves at one speed and no stop-and-go wave forms, so the minutes in the drop belong to the declared δ and Δk. Volume 7, Chapter 14 shows how a line of cars amplifies a brake tap into such a wave. With declared demand no driver avoids a long ramp queue, and with point queues and exact counts the override always sees the true queue. The total time spent weighs every vehicle-hour the same, so fuel, emissions, reliability and fairness between ramp and mainline drivers are outside the score. Every capacity, storage, threshold and target is a teaching value chosen for the lesson, and no number here is a measurement of a real road.

> **OPEN CHALLENGE** **Where Is the Fold Today?**
>
> A local meter holds its setpoint just below the critical occupancy, and that number is hard to know. A study of ramp metering calls the critical density most difficult to estimate correctly online, and the calibration of the fundamental diagram far from obvious [25]. In this chapter a setpoint a quarter point too high breaks the merge down four times. Progress would be a meter that finds its margin from its own data and keeps it as weather, traffic mix and the morning move the fold.

The predictive approximation omits the capacity-drop latch, downstream receiving restrictions, distributed transport, origin queue, speed-limit dynamics and substep empty-queue release cap. Reconstruction using realized counts does not establish multi-step demand accuracy. The 600 veh/h floor experiment exposes request/flow infeasibility rather than permitting negative inventory. The losses in case 5 and the challenge, the wrong-forecast failure and the noisy refinement reversal prevent a general advantage or convergence claim.

> **OPEN CHALLENGE** **A Fair Share of the Wait**
>
> Conventional metering often spreads delay very unevenly across on-ramps, which undermines its acceptance, and strategies seen as unfair meet signal violation and public opposition [26]. The Twin Cities shutdown of 2000 began with such opposition [6]. The score of this chapter weighs every vehicle-hour the same. Progress would be a metering law, simple enough for existing controllers, that keeps the merge flowing while it bounds the wait at the worst-served ramp.

8.8 Conclusion

A freeway merge is a plant whose steady-state map folds over, and the simulator of this chapter makes the fold, the drop below it and the meter that avoids it measurable on a morning that replays exactly. With the signal dark, 17 min of overload hold the merge in the capacity drop for 104.9 min and the morning costs 1840.7 veh·h. ALINEA at 15.5 % keeps the merge out of the drop for 1529.4 veh·h, while its ramp queue reaches 163.9 vehicles against a storage of 60. A setpoint past the fold breaks the merge down four times, the override that protects the arterial hands back 76.0 % of the saving, coordination stores the difference upstream for 1459.9 veh·h, a frozen detector quietly switches the meter off, and an incident that halves the road defeats every strategy, whose totals agree within 0.56 %. The graded morning improves from 1833.4 veh·h with a breakdown and two spilling ramps to 1551.9 with neither. The model is first order with a deterministic drop and declared demand. A fuller model would add a probabilistic breakdown, demand that diverts from long queues, a second-order model with stop-and-go waves, a chain of merges metered together, and setpoints chosen against the spread of the detector noise. The simulator treats a freeway as a feedback problem whose setpoint is a margin and whose actuator spends a budget.

The predictive comparison costs 1469.389 veh·h against ALINEA's 1459.917 veh·h, with 1446.587 veh·h when the forecast is withheld, demonstrating that a nominal constrained plan need not improve the scored morning.

KEY POINTS

- A merge passes the most traffic just before it breaks down, and once it breaks down its queue discharges a tenth less until it is gone.
- ALINEA is integral control on the merge occupancy: in steady flow it settles at its setpoint whatever the demand, without knowing the merge's capacity.
- A setpoint past the fold has no steady state, so the integral action breaks the merge down again after each recovery.
- A meter moves delay in time by storing it, and when the storage runs out the breakdown it postponed arrives.

Where else this applies. Volume 4, Chapter 2 holds a compressor short of its own fold with a protective override.

References

[1] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.

[2] P. I. Richards, "Shock waves on the highway," *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956.

[3] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, 1994.

[4] M. Papageorgiou and A. Kotsialos, "Freeway ramp metering: an overview," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 271–281, 2002.

[5] M. Papageorgiou, H. Hadj-Salem, and F. Middelham, "ALINEA local ramp metering: Summary of field results," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1603, no. 1, pp. 90–98, 1997.

[6] L. Jacobson, J. Stribiak, L. Nelson, and D. Sallman, "Ramp management and control handbook," Federal Highway Administration, U.S. Department of Transportation, Washington, DC, Tech. Rep. FHWA-HOP-06-001, Jan. 2006.

[7] D. C. Gazis and R. S. Foote, "Surveillance and control of tunnel traffic by an on-line digital computer," *Transportation Science*, vol. 3, no. 3, pp. 255–275, 1969.

[8] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, 1988.

[9] E. M. Greitzer, "Surge and rotating stall in axial flow compressors—part I: Theoretical compression system model," *Journal of Engineering for Power*, vol. 98, no. 2, pp. 190–198, 1976.

[10] D. Levinson and L. Zhang, “Ramp meters on trial: Evidence from the Twin Cities metering holiday,” *Transportation Research Part A: Policy and Practice*, vol. 40, no. 10, pp. 810–828, 2006.

[11] A. D. May, “Experimentation with manual and automatic ramp control,” in *Traffic Control: Theory and Instrumentation*, T. R. Horton, Ed. Springer US, 1965, pp. 157–208.

[12] L. Isaksen and H. J. Payne, “Freeway traffic surveillance and control,” *Proceedings of the IEEE*, vol. 61, no. 5, pp. 526–536, 1973.

[13] Y. Wang, E. B. Kosmatopoulos, M. Papageorgiou, and I. Papamichail, “Local ramp metering in the presence of a distant downstream bottleneck: Theoretical analysis and simulation study,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2024–2039, 2014.

[14] M. J. Cassidy and J. Rudjanakanoknad, “Increasing the capacity of an isolated merge by metering its on-ramp,” *Transportation Research Part B: Methodological*, vol. 39, no. 10, pp. 896–913, 2005.

[15] A. Srivastava and N. Geroliminis, “Empirical observations of capacity drop in freeway merges with ramp control and integration in a first-order model,” *Transportation Research Part C: Emerging Technologies*, vol. 30, pp. 161–177, 2013.

[16] G. Gomes and R. Horowitz, “Optimal freeway ramp metering using the asymmetric cell transmission model,” *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 4, pp. 244–262, 2006.

[17] I. Papamichail and M. Papageorgiou, “Traffic-responsive linked ramp-metering control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 111–121, 2008.

[18] R. C. Carlson, I. Papamichail, M. Papageorgiou, and A. Messmer, “Optimal motorway traffic flow control involving variable speed limits and ramp metering,” *Transportation Science*, vol. 44, no. 2, pp. 238–253, 2010.

[19] A. Hegyi, B. De Schutter, and H. Hellendoorn, “Model predictive control for optimal coordination of ramp metering and variable speed limits,” *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 3, pp. 185–209, 2005.

[20] G. Pannocchia and J. B. Rawlings, “Disturbance models for offset-free model-predictive control,” *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[21] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[22] Y. Tang, K. Ozbay, and L. Jin, “Robust queue length estimation for ramp metering in a connected vehicle environment,” arXiv preprint arXiv:2305.17921, 2023.

[23] L. A. Klein, M. K. Mills, and D. R. P. Gibson, “Traffic detector handbook: Third edition, volume I,” Federal Highway Administration, U.S. Department of Transportation, McLean, VA, Tech. Rep. FHWA-HRT-06-108, Oct. 2006.

[24] D. Liu, L. Liu, and L. D. Han, “Analyzing robustness of the deep reinforcement learning algorithm in ramp metering applications considering false data injection attack and defense,” arXiv preprint arXiv:2301.12036, 2023.

[25] C. Join, H. Abouaïssa, and M. Fliess, “Ramp metering: modeling, simulations and control issues,” arXiv preprint arXiv:2111.06610, 2021.

[26] K. Riehl, O. Alami Badissi, A. Kouvelas, and M. A. Makridis, “C-EQ-ALINEA: Distributed, coordinated, and equitable ramp metering strategy for sustainable freeway operations,” arXiv preprint arXiv:2601.06311, 2026.

8.9 Exercises

8.1 ○ Concept check · 2 min

With the signal dark, the demand at the merge exceeds its capacity of 6000 veh/h for 17 min, yet the merge stays in the capacity drop for 104.9 min. Which explanation is right?

(a) The detectors report once a minute, so the breakdown is seen too late for the queue to clear

(b) The congestion wave moves at 16 km/h, so the tail of the queue needs that long to reach the merge

(c) The broken-down merge discharges 5400 veh/h, so its queue grows until the demand is below that and drains only afterward

(d) The demand stays above the capacity of 6000 veh/h until 9:33, the minute at which the merge recovers

8.2 Concept check · 2 min

In scenario 7 the merge detector freezes at 7:20 on a reading of 14.86 %, below the 15.5 % setpoint, and the watchdog is off. What does ALINEA do through the rest of the peak?

(a) It holds the release at its 7:20 value, because a constant reading gives a constant output

(b) It ramps the release up to the 1800 veh/h ceiling within minutes and winds up there

(c) It drives the release down to the 240 veh/h floor, and the ramp queue spills onto the arterial

(d) It adds one fixed increment to the counted release each minute, so the release rides just above the arrivals

8.3 Concept check · 1 min

Scenario 5 adds the queue override, with its limit at 51 vehicles, to the ALINEA meter of scenario 3. What does the override do on the default morning?

(a) It keeps every vehicle off the arterial, and the merge breaks down at 7:57 and loses most of what metering saved

(b) It keeps every vehicle off the arterial at no cost, because ALINEA still holds the occupancy at its setpoint

(c) It lets the queue pass the storage for a few minutes, and in exchange the merge never breaks down

(d) It delays the breakdown until the peak has passed, so the total time spent equals that of scenario 3

8.4 Theory and calculation · 5 min

The merge has three lanes with a capacity of 2000 veh/h per lane, a free speed of 80 km/h and a congestion wave speed of 16 km/h. The effective vehicle length at the detector is 6.5 m.

(a) Find the critical density and the jam density per lane. *Checkpoint:* 25 and 150 veh/km.

(b) Find the critical occupancy, and the free-flow discharge of the merge that one percentage point of occupancy stands for. *Checkpoint:* 16.25 % and 369 veh/h.

(c) Find the density and the discharge at the default setpoint of 15.5 %, and the margin between that discharge and the capacity. *Checkpoint:* 23.85 veh/km per lane, 5723 veh/h and 277 veh/h.

8.5 Theory and calculation · 8 min

A queue waits on the ramp and the merge is in free flow. Treat the merge as a static plant: the occupancy of report k, in percent, is $o(k) = \big(q_{up} + r(k-1)\big)/g$ with $g = 369\,\text{veh/h}$ per point and a constant mainline flow q_{up}. ALINEA follows Equation (8.9) with $K_P = 0$ and $T = 1\,\text{min}$.

(a) Show that the error $e(k) = \hat{o} - o(k)$ shrinks by a fixed factor per report. Find the factor and the time constant at the default $K_R = 70$. *Checkpoint:* 0.810 per report and 4.8 min.

(b) Find the gain that removes the error in one report and the gain above which the error grows. Find the factor at the top of the slider range, $K_R = 200$. *Checkpoint:* 369 and 738 veh/h per percent per minute, and 0.458.

(c) Give the proportional gain and the integral time of the PID of Equation (8.10) that reproduce ALINEA at $K_R = 70$, and PI-ALINEA at $K_R = 70$ with $K_P = 50$. *Checkpoint:* $K_c = 70$ with 1 min, and $K_c = 120$ with 1.71 min.

8.6 Theory and calculation · 6 min

The merge ramp stores 60 vehicles and its override limit is 85 % of the storage. Reports arrive every minute, and the entrance counter reads 1050 veh/h of arrivals.

(a) Find the least release that Equation (8.11) demands at a counted queue of 45, 51 and 55 vehicles. *Checkpoint:* 690, 1050 and 1290 veh/h.

(b) ALINEA holds the release at 655 veh/h, its lowest value in scenario 3. Find the time for an empty ramp to reach the override limit. *Checkpoint:* 7.7 min.

(c) The upstream ramp stores 150 vehicles. Find the counted merge-ramp queues at which coordination starts and ends, the target the upstream ramp fills to, and the time the target takes to fall to zero afterward. *Checkpoint:* 36 and 18 vehicles, 127.5 vehicles and 42.5 min.

8.7 App challenge · 8 min

Load scenario 1, No metering: the breakdown and the shockwave. Use the default seed, and run each morning to 10:00, where the app pauses.

(a) Change no setting. Read the Capacity drop and Total time spent tiles. *Checkpoint:* 104.9 min and 1841 veh·h.

(b) Press Reset, open Show Model, set Recovery hysteresis below the critical density to 0 and run again. Read the same two tiles. *Checkpoint:* 104.5 min and 1841 veh·h.

(c) Reload the scenario, set Capacity drop once a queue forms to 0 % in Show Model and run again. Read the Total time spent tile, and say which of the two parameters holds the merge in the drop. *Checkpoint:* 1405 veh·h, within 1.

8.8 App challenge · 8 min

Load scenario 3, ALINEA just under the critical occupancy. Use the default seed and run each morning to 10:00. Make each change after Reset and before Run. Read the Total time spent, Capacity drop and Ramp queue, worst tiles.

(a) Run the scenario as loaded, then with the Merge occupancy setpoint at 15.0 %, then at 16.0 %. *Checkpoint:* 1529, 1836 and 1426 veh·h, with queues of 164, 365 and 50 vehicles and no breakdown.

(b) Keep 16.0 %, switch to Advanced and set Demand scaling to 2 % under the disturbances. Run again. *Checkpoint:* 1773 veh·h and 24.5 min in the drop.

(c) Keep the demand and return the setpoint to 15.5 %. Run again, and choose a setpoint for a site whose demand is known only to within 2 %. *Checkpoint:* 1725 veh·h, no breakdown, 284 vehicles.

8.9 App challenge · 8 min

Load scenario 5, The queue override. Use the default seed and run each morning to 10:00. Make each change after Reset and before Run.

(a) Change no setting. Read the Total time spent, Capacity drop and Ramp queue, worst tiles. *Checkpoint:* 1766 veh·h, 89.1 min and 51 of 60.

(b) Open Show Model and set Merge ramp storage to 105 vehicles, run, then set it to 110 vehicles and run again. Read the same tiles. *Checkpoint:* 1708 veh·h with 74.6 min, then 1474 veh·h with no breakdown.

(c) Scenario 3 needed a queue of 164 vehicles to hold the merge without the override. Explain why a storage of 110 is enough here, and say whether a designer should rely on it. *Checkpoint:* The true occupancy peaks at 16.23 %.

8.10 Application area · 3 min

At 7:40 a crash one kilometer past the merge halves the capacity of one cell for 20 min. The mainline alone brings about 4900 veh/h.

(a) Find the capacity of the blocked cell, and its discharge once a queue stands behind it. *Checkpoint:* 3000 and 2700 veh/h.

(b) Estimate the vehicles that must queue during the 20 min if the ramp is closed completely, and say what metering can and cannot do here. *Checkpoint:* About 730 vehicles.

8.11 Application area · 4 min

Table 8.8 shows two configurations that meet the graded morning at nearly the same total time spent: ALINEA with the override and coordination at 1551.9 veh·h, and speed limits alone at 1550.9 veh·h.

(a) Say who bears the delay under each configuration. *Checkpoint:* Ramp drivers in one, mainline drivers in the other.

(b) Choose one to commission, using the chapter's results at higher demand and at other detector-noise seeds. *Checkpoint:* Ten of ten seeds against one of ten.

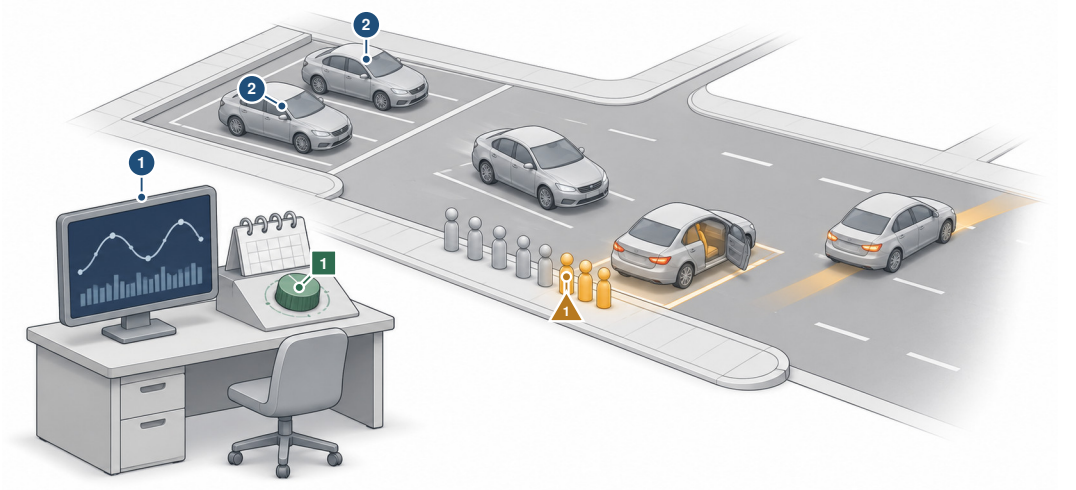
(c) Name two things that matter to a road agency and that the total time spent leaves out. *Checkpoint:* The Limitations section names four.

8.12 Review the agent · 5 min

An AI assistant is asked for the best setpoint of the ALINEA meter and replies as follows. "The merge carries the most at its critical occupancy of 16.25 %. ALINEA is an integral controller, and integral action removes the error, so the occupancy will sit at the setpoint it is given. Set the setpoint to 16.2 %. The merge will then pass 5982 veh/h instead of 5723 veh/h, the queue will shrink, and the margin of 0.05 points keeps the merge on the free branch." Find the flaw in the reply and say how it could be checked.

CHAPTER 9

Surge Pricing as Feedback on a Ride-Hailing Market With a Throughput Fold



The controller moves 1 the price multiplier, changed only at a pricing decision

The controller reads 1 the time a car needs to reach a rider 2 the count of free cars

What disturbs it 1 a sudden crowd of ride requests, such as a stadium letting out

Notice the three kinds of car, parked and free, driving to a rider and carrying one: the price has to keep enough cars of the first kind. Illustration, not to scale.

IN PLAIN TERMS A Price That Keeps Rides Moving

The main idea. When many people want a ride at once, a ride app can raise its price. Fewer people then ask for rides, more drivers come to work, and enough cars stay free to reach riders quickly.

The challenge. If too few cars are free, each driver spends a long time driving empty to a far rider, so the same cars finish fewer trips. Riders answer a price at once but drivers come slowly, so a price that rises late or is held down cannot stop this.

Why it matters. The pricing rule decides whether a crowd leaving a stadium gets rides in minutes or gives up waiting.

THREE WAYS TO READ THIS CHAPTER

Explore	11 min	the first page, the fact on trips in one city (Section 9.1), the history of the moving price (Section 9.2), the try-it segments of cases 2 and 3 and the fact on the rain case (Section 9.6), and the key points
Study	50 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	20 min	the first page, the results (Section 9.6), Technology in Practice (Section 9.7), Limitations (Section 9.8) and the open challenges

The control question A surge multiplier reaches riders at once and drivers only through their earnings, and past the fold of its throughput map a zone completes fewer trips as demand grows. When does a feedback price keep the zone on the near side of that fold, and when do a slow decision calendar, a price cap or the controller's own logic let it cross?

Learning objectives

- Derive the pickup time, the abandoned share and the fixed-fleet throughput map from the three-pool model, and locate the fold.
- Explain the two speeds and the two static gains of a price, and compute the settled fleet as a power of the multiplier.
- Determine the action of the pickup-time PI from the sign of the process gain in the convention of Chapter 1.
- Predict how the decision calendar and the controller gain change the response of the PI to a stadium release.
- Compare a lookup table, the PI and a receding-horizon optimizer with and without event preview on service, abandonment and price.

Prerequisites Sections 1.4.2 and 1.6.2 (velocity form), 1.4.3 (controller action), 1.4.9 (back-calculation), 1.4.11 (sample time) and 1.4.13 (a gain that moves with the operating point). Chapter 8 holds a road below its fold. Section 1.6.3 is needed only for the optimizer.

The app https://apmonitor.com/apps/surge-pricing/

See also Volume 7, Chapter 4 for a process gain that reverses at the peak of a friction curve and Volume 7, Chapter 3 for a plant that the setpoint chooses. Volume 9, Chapter 6 for a price that coordinates consumers and Volume 9, Chapter 5 for releasing ahead of announced demand.

9.1 Introduction

A stadium that empties into a city zone asks for rides faster than any fleet can answer them, and a platform that keeps its fare unchanged answers by sending its last idle cars across the zone. In the simulator of this chapter, a release that quadruples potential requests for 40 minutes, with the fare held at its base, drains the idle pool from 432 to 20 drivers within 9 minutes. The pickup time then reaches 10.2 min, two of every five drivers online are driving empty toward a rider, and up to 2,584 of the 4,800 requests an hour are abandoned. From 12 minutes after the release to its end the zone completes 1,932 to 2,152 trips an hour while the same online fleet, allocated with a fuller idle pool, could sustain 317 to 359 more. A price that rises in time keeps the idle pool and the throughput, and a price that is capped or late does not.

Pricing a perishable capacity over time has long been posed as a control problem. The sale of an inventory that loses its value at a deadline was formulated as intensity control, with structural results for the optimal price as a function of the stock and the time remaining [1]. Ride-sharing platforms were then studied with a queueing-theoretic economic model that captures the incentives of both sides and the high temporal resolution at which they price [2], with stylized models of self-scheduling providers [3], and with networks of locations among which drivers choose where to serve [4], a literature since organized into demand and pricing, supply and incentives, platform operations, and competition, impacts and regulation [5]. Fleets of shared vehicles have been written as fluid models whose rebalancing policy comes from a linear program [6] and coordinated by model predictive control (MPC) with a Lyapunov stability proof [7]. The plant is a set of driver stocks whose flows depend on a price, the actuator is the multiplier, and the controller is a rule, a loop or an optimizer, so the zone is a control problem in the plain sense of the term.

✦ DID YOU KNOW? Nearly 159 Million Trips a Year

In New York City more than 80,000 drivers worked in 2018 for the four largest app-dispatched for-hire services, whose trips grew from roughly 42 million in 2015 to nearly 159 million in 2017 [8].

The belief this plant tests is that a higher price can only cost trips. On the idle-rich side of the market the belief holds, and at the reference demand a multiplier of 2 lowers the completed trips at steady state from 1,185 to 840 an hour. At three times that demand the same model completes 2,563 trips an hour at the base fare and 3,200 at 1.2 times it, because the price buys back the idle drivers whose absence was sending the fleet on long pickups. This failure was anticipated in the theory of dispatched taxis [9] and later modeled and established in platform data as the wild goose chase [10]. The shape is general. Traffic on a crowded road obeys a functional relation between flow and concentration [11], which Chapter 8 draws with a maximum at the road's capacity and defends with a ramp signal, and a centrifugal compressor throttled toward low flow passes the peak of its pressure-rise characteristic, which Volume 4, Chapter 2 guards with a recycle valve. In each the steady map from load to output has a maximum, and past it more load yields less output. This chapter defends such a maximum with a price whose two effects arrive at two speeds.

However, the economic models of the market are solved for equilibria and for optimal static, spatial or contractual prices [2–4, 10], while the fleet controllers move vehicles that the platform owns [6, 7], so neither gives a student one market on which a lookup table, a feedback loop and an optimizer each move a quantized price on a calendar while riders answer at once and drivers later. This chapter works with a market of this kind. It determines when a feedback price keeps a ride-hailing zone on the near side of its throughput fold, and when a decision calendar, a price cap or a controller's own logic lets the zone cross it. The pooled market, the square-root pickup law, the PI loop and receding-horizon control are all established. Every result is measured on one deterministic simulator or derived from its own equations, by linearization and by a map of steady states. Each defect found in the application during this work was corrected there and is reported with its effect before and after.

The chapter is organized as follows. Section 9.2 traces the domain from the regulated taxi to the surge multiplier. Section 9.3 states the zone with every equation, constant and limit, a market whose driver count closes to 5.3×10^{-11} drivers in every case. Section 9.4 derives its two speeds, its two gains and the fold. Section 9.5 describes the pricing rules, among them an optimizer that rolls 44 candidate price paths through the nonlinear model at each decision. Section 9.6 reports nine cases, a constant-demand reference and a 24-hour challenge, each scored on service, abandonment and price together. Section 9.7 describes the devices and platform of a real pricing loop, Section 9.8 states the limitations, Section 9.9 summarizes the results, and the exercises follow in Section 9.10.

9.2 From the Dispatched Taxi to the Surge Multiplier

The price of a taxi ride was long a constant rather than an actuator. In most cities the taxi industry is highly regulated, with restricted entry [12], and a regulated cab charges a metered fare that does not move with demand [13]. A study of 43 cities and counties in the United States and Canada found that such entry controls act differently on the two markets a cab serves. Without them the cab-stand and street-hail market draws an oversupply of cabs and a deterioration of vehicles and drivers, while in the telephone-order, or dispatch, market entry restrictions often leave taxis unavailable [14]. With the fare fixed, only the wait could adjust, and the dispatch market is the one the simulator models.

The geometry that makes waiting fall as supply grows is older than the platforms. For response units scattered over a region as a two-dimensional Poisson process of intensity γ, the mean straight-line distance from a call to the nearest unit is $\frac{1}{2}\gamma^{-1/2}$ [15], so the drive to the nearest idle car shortens only with the square root of the density of idle cars. A model of a dispatched taxi market concluded, in its own title, that taxi travel should be subsidized [9], and a later analysis of the platforms traces the conclusion to the increasing returns of a denser fleet, gives its pickup time as the inverse square

root of the density of idle taxis, and credits it with first anticipating the failure that the platforms would meet [10]. Pickup time in the simulator follows the same law in the size of its idle pool.

Ride-sharing services that connect passengers and drivers over the mobile internet joined dispatch to a price that moves. The history box below follows that price through its first measurements, and the technology boxes of Section 9.7 describe how it is measured and set. The practical problems of designing such a pricing system, and the way it coordinates workers who earn on a flexible schedule, were set out from the platform's side at the 2016 conference on economics and computation [16]. The simulator's multiplier has a floor of one, a quantum of 0.1 and a demand elasticity of 0.5.

> ⧖ HISTORY **Fifty Million Quotes at a Moving Price, 2015**
>
> Ride-sharing services that connect passengers and drivers over the mobile internet began to compete with regulated taxis. In most cities with data, the drivers of one service's core product spent a higher fraction of their time, and drove a higher share of their miles, with a passenger in the car than taxi drivers did [12]. Its price moved. In the first 24 weeks of 2015, around 21 percent of that product's sessions in its four largest United States markets saw a multiplier above 1. Because a continuous internal value was rounded to discrete steps, nearly identical market conditions on either side of a step met different prices. A regression discontinuity design on nearly 50 million sessions gave price elasticities of demand mostly between -0.4 and -0.6, and -0.55 for the whole sample [13].

The theory that followed explains why the price has to move. Optimal pricing in such a two-sided platform needs both an economic model of the two sides and a stochastic model of drivers and passengers over time [2]. With providers who decide for themselves how often to work, surge pricing is not the optimal contract but generally earns nearly the optimal profit, and as labor becomes more expensive both providers and consumers fare better under it [3]. Across a network of locations, the balance of the demand pattern captures a network's profit potential [4]. The wild goose chase is the subject of this chapter. An overburdened platform depleted of idle drivers sends cars on long pickups that occupy them, reduce the trips served and the earnings, and so take drivers off the road, and its authors established the failure in platform data and showed that surge pricing, and to a lesser extent other market design interventions, keep it from crippling the market [10].

Control engineers reached the same fleets through the vehicles. A fluid model of customers and shared vehicles at stations gives a rebalancing policy, by linear programming, under which every station reaches an equilibrium with excess vehicles and no waiting customers [6], and a discrete-time model of an autonomous fleet supports an MPC that solves a mixed integer linear program for scheduling and routing in real time [7]. Those controllers move cars. The controllers in this simulator cannot move a single car, and they reach the fleet only through a price that riders see at once and drivers feel through their earnings.

9.3 A Zone of Three Pools

The simulated zone is one aggregate market, drawn in Figure 9.1 with the riders entering at the left, the three pools of drivers in the middle and the pricing desk above. Time is in hours, pools in drivers, flows in drivers or requests an hour, the price as a multiple m of the base fare f, and pickup times in minutes where the application displays them. Table 9.1 classifies the signals.

9.3.1 Pools and Flows

Drivers move among the three pools, and only the logging on and off of drivers changes their total, as shown in Equation (9.1).

$$\begin{aligned} \dot{O} &= c - a + u_s, \qquad \dot{E} = a - E/\tau_p, \\ \dot{T} &= E/\tau_p - c, \qquad c = T/\tau_t, \qquad N = O + E + T \end{aligned} \tag{9.1}$$

where c is the rate of completed trips, a the rate of accepted dispatches, u_s the net rate at which drivers log on, $\tau_t = 0.25\,\mathrm{h}$ the passenger trip and τ_p the pickup time. A dispatch takes a driver out

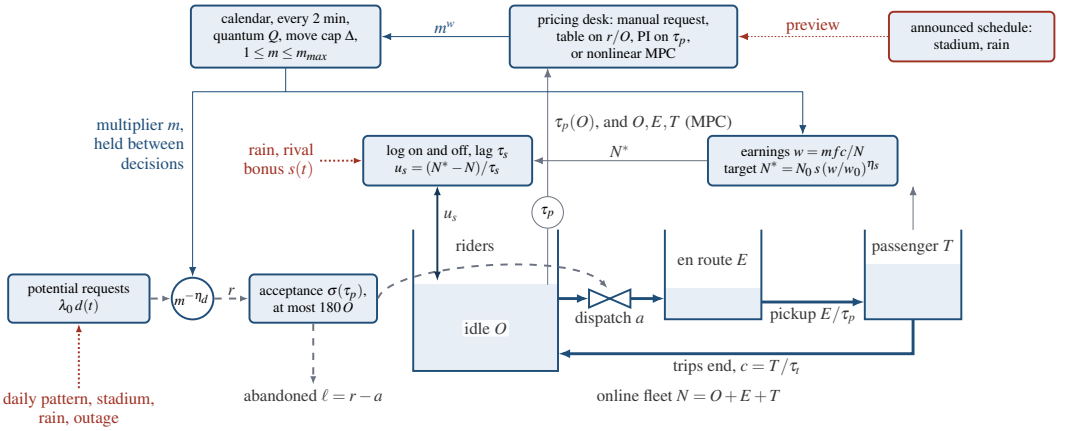


Figure 9.1 One ride-hailing zone as three pools of drivers, not to scale. Potential requests $\lambda_0 d(t)$ pass the price gate $m^{-\eta_d}$ and an acceptance test on the predicted pickup time τ_p, bounded by one dispatch per idle driver every 20 s, and the rest are abandoned. A dispatch a moves a driver from the idle pool O to the en-route pool E, the pickup moves the driver into the passenger pool T, and the end of the trip returns the driver to O. Drivers log on and off at u_s toward a target N^* set by their earnings w and the supply factor $s(t)$. The pricing desk reads τ_p (r/O for the table, and the three pools and the logged requests for the optimizer), and its request m^w becomes the multiplier m only at a calendar decision, rounded to the quantum Q and held within the move cap Δ and the bounds. Riders move along dashed gray arrows. Dotted red arrows mark the disturbances and the preview of the announced schedule, which event preview passes to the optimizer.

Table 9.1 Signals of the Zone by Role

Symbol	Role	Unit
τ_p	CV, pickup time, SP τ^* (3 to 10)	min
m	MV, multiplier in force, 1 to m_{max}	—
m^w	request of the desk or a controller	—
$d(t)$	DV, potential-demand factor	—
$s(t)$	DV, available-supply factor	—
O, E, T	states, idle, en-route and passenger drivers	drivers
N	output, online fleet $O+E+T$	drivers
r, a, ℓ	outputs, requests, dispatches, abandonments	/h
c	output, completed trips	/h
w, N^*	outputs, earnings per driver, supply target	\$/h, drivers
O^*, q^*	fold at the current N, idle pool and trips	drivers, /h

of the idle pool at once, the pickup moves the driver into a trip, and the end of the trip returns the driver to the idle pool. The quantity $N - N(0) - \int u_s \, \mathrm{d}t$ is zero in exact arithmetic and is logged as a conservation residual. The pickup time grows as the idle pool shrinks, as shown in Equation (9.2).

$$\tau_p(O) = \tau_{min} + \frac{a_p}{\sqrt{\max(0.25, O)}} \tag{9.2}$$

where $\tau_{min} = 2\,\text{min}$ and $a_p = 0.5$ h drivers$^{1/2}$, so 400 idle drivers give 3.5 min, 100 give 5 min, 25 give 8 min, and the floor of 0.25 drivers bounds the pickup at 62 min. The second term is the nearest-neighbor law of Section 9.2 with the zone's area and speed lumped into a_p.

Table 9.2 Events of the Market and Their Factors

Event	Effect
stadium	$\delta = 4$ for 40 min
rain	$\delta = 1.5$ and $\varsigma = 0.8$ for 2 h (40 min in case 8)
rival bonus	$\varsigma = 0.7$ for 2 h
app outage	$\delta = 0$ for 10 min, then $\delta = 3$ for 10 min

9.3.2 Riders

Requests answer the price at once, and riders decline a predicted pickup that is too long, as shown in Equation (9.3).

$$\begin{aligned} r &= \lambda_0\, d(t)\, m^{-\eta_d}, \qquad \sigma(\tau_p) = \left[1 + \exp \frac{\tau_p - \tau_{ab}}{1.5\,\text{min}}\right]^{-1} \\ a &= \min\big(\sigma(\tau_p)\, r,\ 180\, O\big), \qquad \ell = r - a \end{aligned} \tag{9.3}$$

where $\lambda_0 = 1200/\text{h}$ is the reference potential demand, $\eta_d = 0.5$, the abandonment midpoint is $\tau_{ab} = 10\,\text{min}$ and the exponent is clipped to ± 50. A predicted pickup of 5 min loses 3.4 % of requests, 8 min 20.9 % and 10 min half. The second bound lets each idle driver take one dispatch per 20-second matching cycle. Rejected and unavailable dispatches both count as abandoned, and a rider who never requests because of the price is not counted at all.

9.3.3 Drivers and Their Earnings

The fleet relaxes toward a level set by the hourly earnings of a driver, as shown in Equation (9.4).

$$\begin{aligned} w &= \frac{m f c}{N}, \qquad N^{*} = N_0\, s(t) \left(\frac{\max(w, 0.1)}{w_0}\right)^{\eta_s} \\ u_s &= \max\big(-6\, O,\ (N^{*} - N)/\tau_s\big), \qquad w_0 = f \lambda_0 / N_0 \end{aligned} \tag{9.4}$$

where $f = \$12$, $N_0 = 800$ drivers, so $w_0 = \$18$ an hour, $\eta_s = 0.5$ and $\tau_s = 0.5\,\text{h}$. Drivers leave only from the idle pool and at no more than six times its size an hour. The multiplier in force pays every trip that ends, whatever price the trip was accepted at, so w moves the instant m does. Gross fares $\int m f c \,\mathrm{d}t$ are logged and are neither the platform's profit nor the riders' welfare.

9.3.4 Demand, Supply and Events

The two disturbance factors combine a slider, a daily pattern and the events of Table 9.2, as shown in Equation (9.5).

$$\begin{aligned} d(t) &= D\, p(h) \prod_i \delta_i(t), \qquad s(t) = S \prod_i \varsigma_i(t) \\ p(h) &= 0.8 + \mathrm{e}^{-((h-18)/1.8)^2} + 0.3\,\mathrm{e}^{-((h-8)/1.2)^2} \end{aligned} \tag{9.5}$$

where D (0.2 to 4) and S (0.3 to 1.5) are sliders at 1, h is the hour of the day from the case's starting hour, $p(h)$ is replaced by 1 unless the daily pattern is on, and δ_i and ς_i are the factors of the events in force. The pattern is 1.80 at 18:00, 1.10 at 08:00 and 0.80 through the night. A scripted event is announced in the case's schedule, and an event fired from the panel joins the same list at the moment it is fired, after which a forecast can see the rest of it. Nothing in the model is random.

9.3.5 The Fold

At a steady state every dispatched driver spends τ_p en route and τ_t in a trip, so $E = c\tau_p$ and $T = c\tau_t$, and a fleet of N drivers with O of them idle can sustain the trip rate of Equation (9.6).

$$q(O; N) = \frac{N - O}{\tau_t + \tau_p(O)} \tag{9.6}$$

Table 9.3 Teaching Values and Where Each Is Set (P a Model-Dialog Parameter, C a Card Control, A an Advanced Control, S the Source)

Symbol	Meaning	Default	Range		Symbol	Meaning	Default	Range	
N_0	baseline drivers	800	200–1600	P	τ^*	pickup target (min)	3.5	3–10	C
λ_0	baseline requests (/h)	1200	300–2400	P	K_c, τ_I	PI gain (/min), integral time (h)	0.45, 0.25	0–2, 0–2	C
η_d, η_s	demand, supply elasticity	0.5, 0.5	0.2–1.5, 0–1.2	P	h	surge-off hysteresis (min)	0.3	0–1.5	C
τ_s	supply lag (h)	0.5	0.1–2	P	m_{max}	price cap, off as shipped	2.5	1–6	C
τ_{min}, a_p	minimum pickup (min), geometry (h drivers$^{1/2}$)	2, 0.5	1–5, 0.1–1.5	P	—	decision calendar	2 min	10 min, 30 s	C
τ_t	passenger trip (h)	0.25	0.1–0.5	P	Q, Δ	quantum, largest move per decision	0.1, 1	0–0.5, 0.1–3	A
τ_{ab}	abandonment midpoint (min)	10	5–30	P	H	forecast horizon (h)	1.5	0.5–2	A
f	base fare ($/trip)	12	5–30	P	w_p, w_a	wait, abandonment weights	1, 2	0–5, 0–10	C
D, S	demand, supply sliders	1, 1	0.2–4, 0.3–1.5	C	w_u, w_c	price-move, trips weights	0.1, 0.4	0–5, 0–5	A
—	abandonment scale (min), dispatch cycle (s)	1.5, 20	—	S	w_e, w_r	earnings, gross-fare weights	0, 0	0–10, 0–10	A
—	exit limit (/h), earnings floor ($/h)	6O, 0.1	—	S	—	price-level weight, in the MPC	0.22	—	S
—	late pickup, challenge (min)	8	—	S	Δt, Δt_c, log	physics, control (s), log (min)	10, 30, 1	—	S

The map vanishes when every driver is idle and falls again as $O \to 0$, because the pickups lengthen, so it has an interior maximum q^* at an idle pool O^*, the fold. The same accounting of idle, en-route and driving drivers underlies the analysis of [10]. The application finds the maximum at the current fleet by ternary search, draws the curve and the live point in its Goose chase view and raises the WILD GOOSE CHASE status whenever $O < O^*$. The live point is transient and need not lie on the curve.

9.3.6 Start, Integration and Checks

Every case starts at the equilibrium of Equations (9.1) to (9.4) at $m = 1$ and constant demand and supply, found by bisection on the idle pool and then relaxed through the complete model, with $O = 432.41$, $E = 68.00$, $T = 296.26$ and $N = 796.66$ drivers and a pickup of 3.443 min. There 1.25 % of the 1,200 requests are abandoned, so earnings are $17.85 an hour, a little below w_0, and the fleet settles 3.34 drivers below N_0. The framework's fourth-order Runge–Kutta (Section 1.6.1) advances the state on a step of $\Delta t = 10$ s (1/360 h), the controller executes every 30 s and the historian logs every simulated minute. At the fastest speed, three simulated hours a second, the four hours of case 2 run in under two seconds. Over the full duration of every case the conservation residual stays below 5.3×10^{-11} drivers and no pool falls below 9.31 drivers (the idle pool of case 8), halving the step changes case 8's peak pickup by 0.0007 min and its 3,338 abandonments by 0.01, and the reference holds its equilibrium for three hours. The application passes its own 12 checks and the framework's 17 shared ones.

9.3.7 Assumptions

One zone, no streets. The zone is one aggregate market, so drivers never relocate, demand has no destinations and the pickup law averages over the zone. Spatial imbalance between locations, which shapes optimal prices on a network [4], is outside the model.

Pools, not agents. Drivers are fluid pools that answer current earnings without delay, strategy or memory, and the first idle driver is always dispatched, with no limit on the dispatch radius. The City view's dots illustrate the pools and are not simulated individually. The fluid model is least credible when the idle pool falls to the 13 and 9 drivers reached in cases 2 and 8.

Riders without a queue. A rider accepts or declines the predicted pickup at the moment of request, so there is no age-structured queue of unmatched riders, and riders the price deters are not counted. Requests are expected values and need not be whole numbers.

Teaching values. Every elasticity, lag, geometry coefficient, threshold and event size is chosen to expose a mechanism, and the model is not fit to any city. The demand elasticity lies inside the range estimated from platform data [13], the pickup law follows the nearest-neighbor result [15], and the supply elasticity and lag are not taken from a published estimate.

9.4 Two Speeds and a Fold

Near the reference the zone is well damped. The Jacobian of Equations (9.1) to (9.4) in (O, E, T) at $m = 1$, by central differences of the application's own flow function, has eigenvalues of −17.72/h, the en-route pool draining at the pickup rate in 3.4 min, and a pair at $-3.36 \pm 0.36j$ per hour, an 18 min mode that couples the trips with the fleet's supply response. The input column is $(1389, -593, 0)$ drivers an hour per unit of m. The price enters the idle pool through the dispatches it prevents and through the drivers its earnings attract, and it reaches the passenger pool only through the en-route pool.

9.4.1 Two Gains

The price acts on the two sides with very different static gains. At the reference demand a multiplier of 2 held for six hours removes 29.3 % of the requests (1,200 to 848.5 an hour) but adds only 12.3 % to the fleet (796.66 to 894.81 drivers), and the idle pool grows from 432 to 640, the pickup falls from 3.44 to 3.19 min, the completed trips fall from 1,185 to 840 an hour and the earnings rise from \$17.85 to \$22.52. With little abandonment, $c \approx r$, so $w \propto m^{1-\eta_d}/N$ and $N \propto w^{\eta_s}$, and the fleet follows $m^{(1-\eta_d)\eta_s/(1+\eta_s)}$, the sixth root of m at the defaults. A multiplier can therefore buy far fewer drivers than it removes riders, and in steady state its main effect on the idle pool comes from the demand side.

9.4.2 Two Speeds

The two sides also answer at different speeds, as shown in Figure 9.2(b). With the reference run in MANUAL and the multiplier request raised from 1 to 2 at hour 0.5, the requests fall by the factor $2^{-1/2}$ at the decision itself. The earnings double at the same instant, from \$17.85 to \$35.70 an hour, because the trips then ending are paid at the new price, so the supply target jumps from 796.7 to 1,126.6 drivers and the fleet starts to grow at once. Timed from the decision, it covers half of its eventual rise in 5.3 min, 63 % in 7.3 min and 95 % in 13.8 min, then overshoots to 913.2 drivers 33 min after the move, while the trip pool empties and the earnings fall back, and settles at 894.8. The lag τ_s is therefore a first-order relaxation and not a dead time. Paying each trip at the multiplier in force when it was accepted would hold the earnings signal back by a pickup and a trip, about 18 min here, but would move every run in which the price moves, so the application keeps the jump and its texts now state it.

9.4.3 The Fold and a Gain That Changes Sign

The fold belongs to the allocation of a fleet and not to any controller. At the fixed fleet $N_0 = 800$, which the model reaches with $\eta_s = 0$, the map $q(O; 800)$ peaks at 2,128.3 trips an hour with 65.66 idle drivers and a pickup of 5.70 min, as shown in Figure 9.2(a). The steady states at the base fare for potential demand from 0.5 to 5 times the reference lie on that curve and walk left along it. At 1.5 times the reference the zone keeps 241.9 idle drivers and completes 1,769 trips an hour, the steady state crosses the maximum at 1.875 times, and beyond it every increase completes fewer trips, 2,069 at twice the reference, 1,897 at three times and 1,839 at four times, with pickups of 7.2, 9.8 and 10.7 min and 13.8 %, 47.3 % and 61.7 % of requests abandoned. These states are reached by integrating the model, so they are stable, and abandonment is the likely reason, because a longer predicted pickup turns more riders away and spares the idle pool.

With the fleet free to answer its earnings, the fold moves with the fleet, but the consequence for a controller is the same, a plant gain that changes by more than an order of magnitude and a trip response that changes sign. Section 1.4.13 shows what such a move does to the loop gain $K_c K_p$ of a

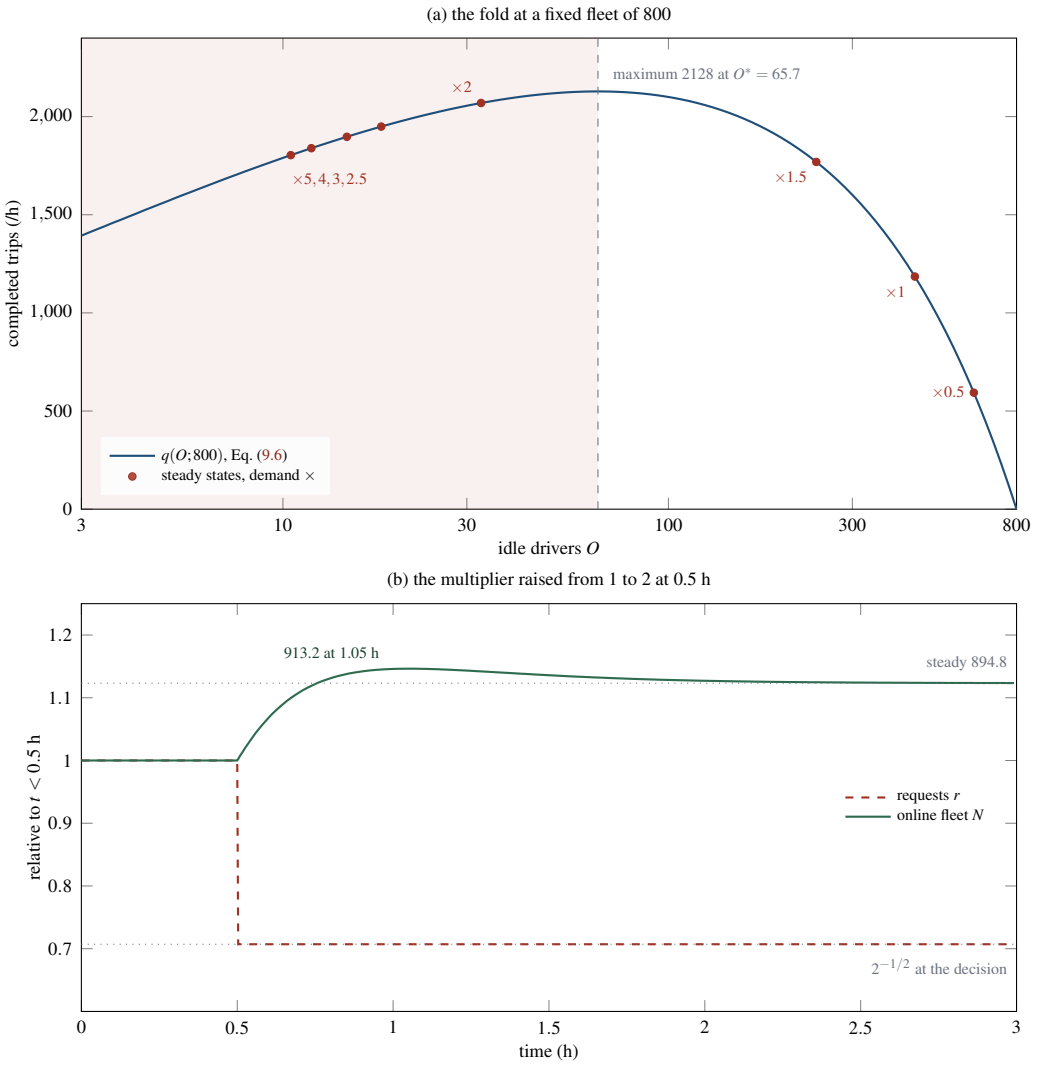


Figure 9.2 Two characteristics of the zone from the application's own functions and engine. (a) The fixed-fleet throughput map of Equation (9.6) at $N_0 = 800$ and the steady states that the full model reaches at the base fare, with the fleet held at N_0, for potential demand from 0.5 to 5 times the reference, drawn as dots on the solid curve. The shaded side is the wild goose chase. (b) The reference with the multiplier request raised from 1 to 2 at hour 0.5, requests (dashed) and online fleet (solid) each divided by its value before the move.

fixed tuning. The steady-state sensitivity of the pickup time to the multiplier, taken between 1 and 1.1, is –0.49 min per unit at the reference demand, −1.27 at 1.8 times, −6.96 at 2.5 times and −10.0 at three times. At three times the reference a multiplier of 1.2 instead of 1 shortens the steady pickup from 8.64 to 4.59 min and raises the completed trips from 2,563 to 3,200 an hour, and at four times a multiplier of 1.5 raises them from 2,373 to 3,460. On the idle-rich side a higher price costs trips, and beyond the fold it buys them, which is the sense in which surge pricing solves the wild goose chase [10]. Volume 7, Chapter 4 meets the same reversal of the process gain at the friction peak of a braking tire.

🕮 **Pause.** Try Exercise 9.2 before reading on.

9.5 Pricing on a Calendar

The controller executes every 30 s, $\Delta t_c = 1/120$ h, and an execution becomes a pricing decision only when the calendar says so. At a decision the selected rule forms a request m^w, and the request is shaped into the multiplier that then holds until the next decision. The manual desk and every automatic rule share the shaping.

9.5.1 The Calendar and the Shape of a Price

Both pieces come from the framework module `core/cadence.js`. The calendar decides every 2 minutes as shipped, every 10 minutes, or at every 30 s execution, and each instant is taken by the first execution that reaches it, beginning at $t = 0$. Both intervals are whole numbers of executions, so the realized period is the nominal one. A decision turns the request into the multiplier in force by Equation (9.7).

$$\begin{aligned} \tilde{m} &= Q\,\mathrm{round}(m^w/Q), \qquad \Delta_Q = Q\max\left(1, \lfloor \Delta/Q \rfloor\right) \\ m_k &= \Pi_{[1,m_{max}]}\left(m_{k-1} + \mathrm{sat}_{\Delta_Q}(\tilde{m} - m_{k-1})\right) \end{aligned} \tag{9.7}$$

where $Q = 0.1$, the move cap is $\Delta = 1$ per decision, m_{max} is the price cap when it is on and 6 otherwise, ties round away from zero, the bounds are moved onto the grid first and a request that is not finite leaves the multiplier unchanged. The application's check over the 150 decisions of case 7 finds every decision on a 2-minute instant, every multiplier on the 0.1 grid and no move larger than the cap, and consecutive decisions there are exactly 120 s apart.

9.5.2 The Desk and the Table

In MANUAL the learner's slider, 1 to 6 in steps of 0.1, sets the request, which the next decision shapes like any other, and a helper button returns it to the base fare. The lookup table reads the requests per idle driver at the price in force, $\rho = r/\max(1, O)$, and asks for 1 when $\rho < 4$, 1.5 when $\rho < 8$, 2.5 when $\rho < 16$ and 4 above. Because r carries the price in force, the price enters the table's own input with no delay.

9.5.3 A PI Loop on Pickup Time

The PI loop is the framework's velocity-form PID with no derivative, executed at decisions only, as written in Equation (9.8).

$$\begin{aligned} e_k &= \tau_{p,k} - \tau^*, \qquad \Sigma_k = \Sigma_{k-1} + K_c e_k - P_{k-1} \\ \hat{m}_k &= \Sigma_k + \mathscr{I}_{k-1}, \qquad m_k^w = \Pi_{[1,m_{max}]}(\hat{m}_k) \\ \mathscr{I}_k &= \mathscr{I}_{k-1} + \Delta t_k \left[K_c e_k/\tau_I + (m_k^w - \hat{m}_k)/T_t\right] \end{aligned} \tag{9.8}$$

where $K_c = 0.45$ per minute of pickup, $\tau_I = 0.25$ h, $T_t = 0.1$ h and Δt_k is the interval back to the last decision. These are Equations (1.23) to (1.26) of Section 1.6.2 with $\tau_D = 0$, in the symbols $\Sigma_k = S_k$, $\hat{m}_k = v_k$, $m_k^w = u_k$ and $\mathscr{I}_k = I_k$, with the bias held in the integral. P_{k-1} is the proportional term $K_c e_{k-1}$ formed at the last decision with the gain then in force, so the running sum is never clipped and a gain change acts on the whole error at once. Back-calculation acts on the integral $\mathscr{I}_k$ only [17]. An integral time of zero removes the integral and its back-calculation.

The error $e_k = \tau_{p,k} - \tau^*$ is the framework's $a(\mathrm{SP} - \mathrm{PV})$ with $a = -1$. A higher price shortens the pickup, so the process gain is negative, and the controller must raise its output as the PV rises. A long pickup raises the price. In the convention of Section 1.4.3 this is a direct-acting controller, and the loop gain aK_cK_p is positive. The app's briefing for case 4 also calls it direct-acting. Named by its push against the pickup error, the same loop would be called reverse-acting, so the error equation, not the name, fixes the sign.

🕮 **Pause.** Try Exercise 9.1 before reading on.

When the shaped multiplier differs from the request by more than one quantum, because the move cap held it, the integral is corrected again by $(m_k - m_k^w)\min(1, \Delta t_k/0.1\,\mathrm{h})$. In MANUAL, or while

the table or the optimizer owns the price, the loop tracks the multiplier in force, which makes every switch between rules or modes bumpless (Section 1.4.10).

The surge-off hysteresis $h = 0.3$ min acts on the request after the PI. A latch turns on when $\tau_p > \tau^* + h$ or $m_k^w \geq 1.3$ and off when $\tau_p < \tau^* - h$, and while it is off any request below 1.3 is replaced by 1. A band of zero removes the latch and this floor. As first built, only the pickup could set the latch, and Section 9.6 measures both versions.

9.5.4 A Receding-Horizon Price

The optimizer plans six blocks of price over a horizon of $H = 1.5$ h by direct shooting on the simulator's own flow function, evaluated with the declared values of Table 9.3 and the base fare in force. It is a nonlinear model predictive controller [18] in that a nonlinear model predicts each candidate and only the first move is applied, but it is solved by a bounded search rather than a gradient method. It differs from the linear controller of Section 1.6.3, which solves a quadratic program on a linear model. The horizon is cut into 90 steps of one minute and the blocks into 15 minutes each, and every candidate path is rolled forward from the present (O, E, T) by six midpoint substeps per minute with the pools clipped at zero. The cost of Equation (9.9) is summed along the roll-out.

$$J = \sum_{j=0}^{89} \frac{1}{60}\Big[w_p(\tau_{p,j} - \tau^*)_+^2 + \frac{12 w_a \ell_j}{\lambda_0} + 0.22(m_j - 1)^2 - \frac{w_c c_j}{\lambda_0} - \frac{w_e w_j}{20} - \frac{w_r m_j c_j}{\lambda_0}\Big] + w_u \sum_{b=0}^{5} (m_b - m_{b-1})^2 \tag{9.9}$$

where τ_p is in minutes, the stage terms are evaluated at the start of each minute, m_{-1} is the multiplier in force, and the weights default to $w_p = 1$, $w_a = 2$, $w_c = 0.4$, $w_u = 0.1$ and $w_e = w_r = 0$. The level term $0.22(m-1)^2$ is fixed in the source and is what keeps a permanent high price from looking free. With event preview on, the forecast at minute j is the market of Equation (9.5) at $t + j/60$ with D and S at their declared value of 1, including the daily pattern, the announced schedule and any event already fired, and its demand is scaled by the requests logged at the price in force over those the declared market gives at that price, a ratio of exactly 1 in every case reported here. With preview off, the present market so formed is held over the horizon.

The search starts from the previous plan, tries six constant paths at 1, 1.5, 2, 3, 4 and m_{max}, then moves each block by ± 0.8, ± 0.3 and ± 0.1 in turn, keeping any improvement, 43 evaluations in all and a 44th to draw the prediction. Every candidate is projected so that the first block lies within Δ of the multiplier in force, each later block within $n_b \Delta$ of the one before, and all within $[1, m_{max}]$, where n_b is the number of decisions in a block, 7.5 at the 2-minute calendar, 1.5 at the 10-minute one and 30 at every execution. The first block is the request, rounded at the decision by Equation (9.7), and the plan is solved again at the next decision. One decision therefore costs 51,480 evaluations of the flow function, and the search is local, with no claim to a global optimum. As first built, the projection allowed only Δ between blocks, so with $\Delta = 0.1$ the preview case began its pre-surge at hour 0.73 instead of 1.17 and peaked at 4.37 min instead of 4.11, with 77.8 instead of 75.0 abandonments at a mean of 1.454 instead of 1.398.

9.5.5 Zone, Predictions and Literature

The simulation model is Equations (9.1) to (9.5), integrated with every clip. The table and the PI hold no model at all. The optimizer's internal model is the same flow function at the declared values of Table 9.3, integrated by a different method, with pools clipped at zero, the declared market either previewed or frozen and scaled by the logged requests, the block-to-block move limit above and the three pools read from the platform's books without noise. An edited coefficient or supply factor is therefore a model mismatch, as on a real plant (Section 1.6.3), that it meets only through these measurements, and the PI and the table read only the logged pickup time, requests and idle drivers.

Table 9.4 The Cases and the Challenge

#	Change from the defaults	Teaching point
1	MANUAL at $m = 1$, daily pattern from 15:00, 6 h	two speeds by hand
2	MANUAL at $m = 1$, stadium at hour 1, 4 h	the fold
3	table, every 10 min, daily pattern from 15:00, 6 h	a relay
4	PI, daily pattern from 15:00, 6 h	a band and a quantum
5	MPC, preview off, $\Delta = 0.1$, stadium at hour 2, 4 h	no foresight
6	case 5 with preview on	pre-surge
7	PI, daily pattern from 16:00, rain at hour 1, 5 h	both sides move
8	MPC, cap 1, stadium and 40 min of rain at hour 1, 4 h	infeasible
9	MPC, $w_e = 3$, stadium at hour 2, 4 h	whose objective
ch	PI, $\tau^* = 5$, cap 2, preview off, daily from 00:00, rain at 17:00, stadium at 23:00, 24 h	a city day
ref	MANUAL at $m = 1$, 3 h	the balance

Table 9.5 Measured Cases Over Their Full Duration

Case	peak τ_p (min)	least O	aband.	completed	mean m	moves
1	3.76	289.6	130.2	9423	1.000	0
2	10.23	13.3	1420.8	5779	1.000	0
3, every 10 min	3.56	368.6	112.5	8902	1.125	16
3, every 2 min	3.54	381.9	112.2	8894	1.125	68
4, $h = 0.3$	3.71	308.1	117.2	9037	1.078	8
4, $h = 0$	3.60	353.2	117.0	9043	1.074	4
5	5.22	86.7	98.7	5994	1.287	27
6	4.11	202.1	75.0	5797	1.398	24
7, PI	4.15	195.3	116.6	8346	1.496	18
7, MPC	3.79	282.0	108.3	8284	1.497	18
7, $m = 1$	9.33	16.7	1925.0	8528	1.000	0
8, cap 1	11.83	9.3	3338.1	5462	1.000	0
8, cap 2.5	9.02	18.2	656.2	5874	1.613	7
9, $w_e = 3$	3.54	377.6	48.1	4522	2.470	42
9, $w_e = 0$	4.15	195.0	75.2	5804	1.396	23
ref	3.44	432.4	44.9	3555	1.000	0

Abandoned and completed are totals over the case, and every run conserves drivers to 5.3×10^{-11}. Variants are measured through Reset.

The models these stand for are the steady-state accounting and calibrated market of [10], the queueing model of [2] and the dispatch model of [9], whose coefficients are not copied here. Nothing in the application is fit to data.

9.6 Eleven Runs of One Zone

Table 9.4 lists the cases. Each starts from the defaults of Table 9.3 and changes only what its row lists. The app lists the challenge as case 10 and the reference as case 11. The values below were measured headless on the current engine. A setting chosen before Run is included in the priming decision at $t = 0$, exactly as the same setting carried through Reset, which keeps the reader's settings (Section 1.6.1). Every variant was checked on both startup paths. Table 9.5 gathers the main values, and Figure 9.3 draws three of its columns. Each case but the challenge is credited once it has run for its full duration and the reader records the comparison of pickup, price and driver pools that it is built around. The challenge is judged on the case as given, so a changed model parameter or disturbance voids the attempt.

9.6.1 Rush Hour by Hand and a Stadium at the Base Fare

The first case runs the evening rush from 15:00 at the base fare. Requests grow from 1,309 an hour at 16:00 to 2,160 at 18:00, the pickup peaks at 3.76 min at 17:59 with 289.6 drivers idle, and 130.2 of

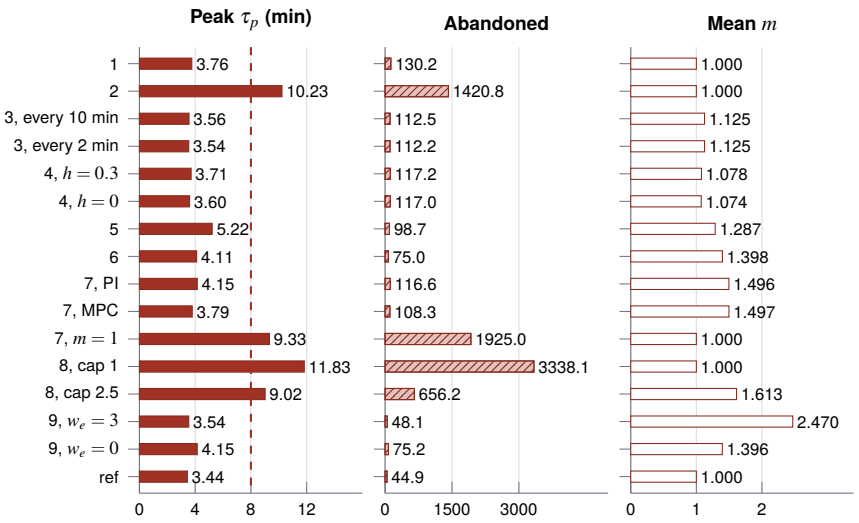


Figure 9.3 Three columns of Table 9.5, each case over its full duration: the peak pickup time (solid bars) with the 8 min limit dashed, the abandoned requests (hatched) and the mean multiplier (open). Notice that every run whose peak crosses the limit, case 2, case 7 at $m = 1$ and both runs of case 8, abandons more requests than any run that stays below it, and that case 9 with $w_e = 3$ abandons the fewest requests of the stadium runs at the highest mean multiplier of the table.

9,518 requests are abandoned, so the rush alone never approaches the fold. A request of 1.5 made at 16:30 takes effect at that decision, cutting the requests from 1,559 to 1,273 an hour, and the fleet grows from 814.6 to 928.0 drivers by 17:00 and 1,020.5 by 19:00. The pickup never exceeds 3.54 min, the abandonments fall to 95.3, and the six hours complete 8,115 trips instead of 9,423 for $135,470 of gross fares instead of $113,073.

The second case holds the base fare while a stadium quadruples the potential requests for 40 minutes from hour 1. The idle pool falls below the fold 7.2 min after the release, the pickup passes 8 min at 8.3 min and peaks at 10.23 min with 13.28 drivers idle, and between 40.7 and 41.5 % of the fleet drives empty to a rider from hour 1.2 to the end of the event (both ends included), against 8.5 % before it. The fleet grows to 965.7 drivers by hour 1.93 because the earnings rise, but over the same window the throughput stays 317 to 359 trips an hour below the maximum of the current fleet. The state at hour 1.5 is shown in Figure 9.4. The zone is in the wild goose chase from hour 1.119 to 1.711, and when the crowd has gone the refilled idle pool lets it complete 2,423 trips an hour at hour 1.78, more than at any moment of the event. The four hours abandon 1,420.8 requests and keep the pickup below 8 min 86.6 % of the time.

> **TRY IT IN THE APP** **2 · No Surge and the Capacity Fold**
>
> Load the case and press Run. Watch the status line and the live point of the Goose chase view after the stadium empties at hour 1. The status turns to WILD GOOSE CHASE, the point slides onto the shaded side of the curve, and the pickup climbs past 8 minutes to about 10.2 min while the price stays at 1.

Under any automatic rule with the shipped settings the same stadium never reaches the fold, although the PI passes within 7.5 idle drivers of it. The PI peaks at 5.36 min, the table at 3.71 and the optimizer at 3.73 without preview and 4.15 with it. The calendar decides whether the PI keeps up. Deciding every 30 s it peaks at 5.10 min with 45 moves, every 2 minutes at 5.36 with 17, and every 10 minutes the zone spends 0.28 h in the goose chase, the idle pool falls to 13.4 drivers, the pickup reaches 10.19 min and 232.8 requests are abandoned. Its gain matters as much. With K_c of

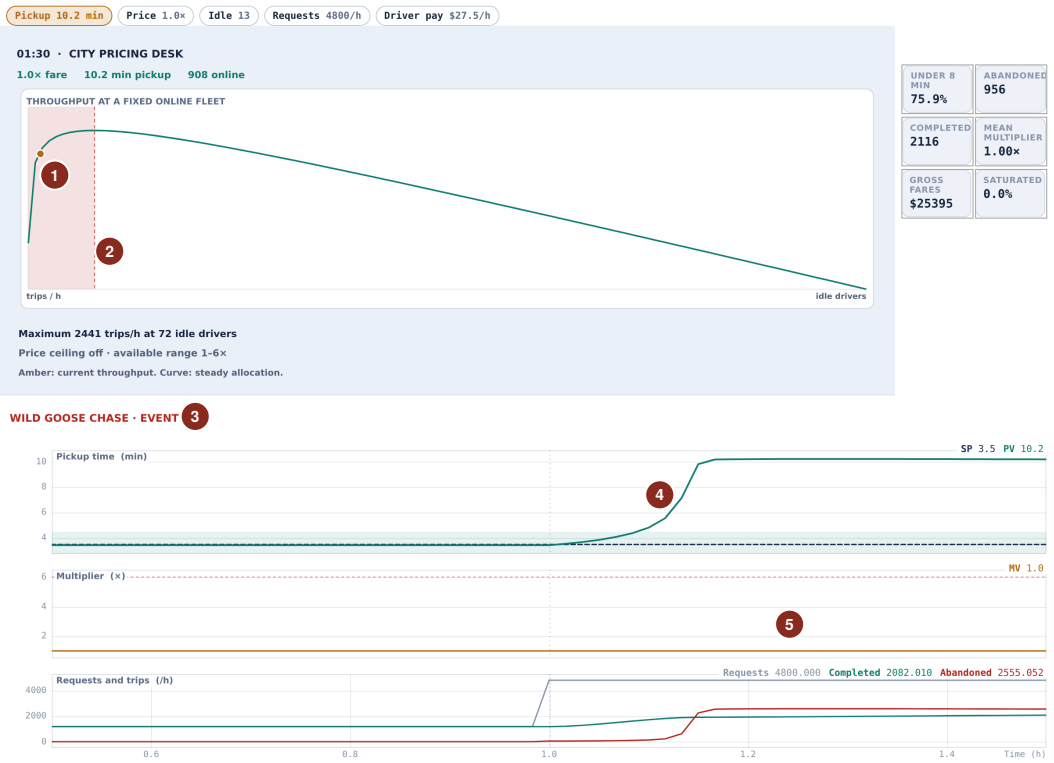


Figure 9.4 Case 2, the stadium at the base fare, at $t = 1.50$ h in the Goose chase view, in MANUAL as loaded, with a one-hour trend window. The figure is a composite of the screenshot: the plant view above the trends, with six tiles of the Learn panel framed at its right and the header and Control panel cut. The clock reads 01:30, although the time, a running sum of steps, is $1.5 - 4.7 \times 10^{-15}$ h, because it counts whole minutes with a tolerance. The badges read pickup 10.2 min, price 1.0, idle 13, requests 4,800 an hour and driver pay $27.5 an hour, and the status line (3) reads WILD GOOSE CHASE and EVENT. The plot holds the online fleet at its present 908 drivers, with a maximum of 2,441 trips an hour at 72 idle drivers, and the live point (1) sits on the shaded side, at 13.4 idle drivers and 2,082 trips an hour by the run's record, left of the dashed line at the maximum (2). In the trends the pickup (4) climbs from just below its target of 3.5 min, 3.44 min by the run's record, after the release at hour 1, the multiplier (5) stays at 1, and the zone completes 2,082 trips an hour while 2,555 of the 4,800 requests an hour are abandoned. The tiles read 75.9 % of the time below 8 min, 956 abandoned, 2,116 completed and $25,395 of gross fares, and the Saturated tile reads 0.0 %, the share of time the price request has sat at its cap.

0.1, 0.2, 0.45, 1 and 2 the peak pickup is 8.80, 7.14, 5.36, 4.50 and 4.08 min, because the stadium is a disturbance that the price can meet only if it moves within minutes.

9.6.2 A Table That Chatters

The third case prices the evening rush with the lookup table every 10 minutes. From hour 1.50 to 4.17 the table alternates between 1 and 1.5, moving at 16 of its 17 decisions. At the decision at hour 1.5 the ratio ρ is 4.11 at $m = 1$ and would be 3.36 at $m = 1.5$, on either side of the threshold of 4, so each move reverses the one before. Deciding every 2 minutes gives 68 moves and every 30 s 264, while the mean multiplier stays 1.125, the abandonments 112.5, 112.2 and 112.1 and the peak pickup 3.56, 3.54 and 3.53 min. The table is a relay closed through the price's direct effect on its own input,

so every move reverses the last on any calendar, and only the number of price changes riders see depends on the calendar. The table has no band around its thresholds, the usual remedy for a relay that chatters.

> **TRY IT IN THE APP 3 · A Supply–Demand Lookup Table**
>
> Load the case and run it. From hour 1.5 the multiplier trend steps between 1 and 1.5 at almost every 10-minute decision, while the pickup never passes 3.56 min. Then set the decision calendar to every 2 minutes, press Reset and run again. The price moves 68 times instead of 16, and the pickup hardly changes.

9.6.3 A Band, a Floor and a Quantum

The fourth case prices the same rush with the PI. As first built, only a pickup above 3.8 min could set the latch, which a case peaking at 3.709 min never does, so every request below 1.3 became 1 while the PI asked for 1.291 to 1.311, and the price alternated between 1.0 and 1.3, 30 moves from hour 2.43 to 4.03. With the latch also set by a request of 1.3, the price jumps from 1 to 1.3 at the decision at hour 2.43, when the PI's request reaches 1.31, and the latch stays on for the rest of the case because the pickup never falls below 3.2 min (its least value after the latch is 3.39 min). Five moves between 1.2 and 1.3 follow from hour 3.00 to 3.23, while the PI asks for 1.247 to 1.251, on the boundary between two quanta, and the price returns to 1.1 at hour 4.17 and to 1 at 4.73, 8 moves in all, at a mean of 1.078 with 117.2 abandonments. With no hysteresis the price moves four times, to 1.1 at hour 1.63, 1.2 at 2.43, 1.1 at 3.90 and 1 at 4.63, at a mean of 1.074 with 117.0 abandonments. The band delays the first surge by 48 minutes, makes it three quanta tall and cuts the time the pickup spends above its target from 2.26 to 1.56 h. Bands from 0.3 to 1.5 min give the same run, 0.2 gives 10 moves and 0.1 gives four, and the reversals that remain come from the quantum, not from the band. Gain makes its own switching, because with no hysteresis and $K_c = 2$ the price moves 20 times with 15 reversals.

9.6.4 Foresight Against a Rate Limit

Cases 5 and 6 release the stadium at hour 2 to the optimizer with a move cap of 0.1 per decision. Without preview it sets 1.1 at hour 0.17 and holds it until the crowd arrives, then climbs 0.1 at every decision to 2.4 at hour 2.40, and the pickup peaks at 5.22 min at hour 2.27 with 86.7 drivers idle. With preview it begins to raise the price 0.83 h before the event and reaches 2.2 when the crowd arrives, with 907.5 drivers online and 637.0 idle at the release against 809.7 and 461.6 without preview. Its peak pickup is 4.11 min and comes at the end of the event, after the optimizer, knowing the crowd will leave, has already lowered the price to 1.5. Preview cuts the abandonments from 98.7 to 75.0 and raises the mean multiplier from 1.287 to 1.398, and it also suppresses 234 requests and completes 197 fewer trips (5,797 against 5,994), so foresight buys service with price and not with capacity.

The advantage belongs to the rate limit. With the shipped move cap of 1 per decision the blind optimizer reaches its price in two decisions, peaks at 3.73 min and abandons 75.3 requests at a mean of 1.3025, while preview peaks at 4.15 and abandons 75.2 at 1.396. With a move cap of 0.3 the two abandon 78.0 and 75.2. With a move cap of 0.1 and decisions every 10 minutes the blind optimizer falls into the goose chase (9.29 min, 695.6 abandoned) and preview holds 4.01 min and 75.2. Each pair runs the same through Reset as before Run. A forecast substitutes for actuator speed, and where the actuator is fast enough it adds little. Volume 9, Chapter 5 uses announced demand in the same way, releasing water one travel time ahead of published orders.

Pause. Try Exercise 9.3 before reading on.

9.6.5 Rain on Both Sides

The seventh case runs the evening rush from 16:00 and adds rain from 17:00 to 19:00, which raises the requests by half and cuts the supply target by a fifth at the peak of the rush. At the base fare the zone falls into the wild goose chase from 17:17 to 19:03, the pickup reaches 9.33 min, only 67.2 %

of the time stays below 8 min and 1,925.0 requests are abandoned. The PI first raises the price at 17:04, after the rain has begun, reaches 2.0 at 18:02 and peaks at 4.15 min with 116.6 abandonments at a mean of 1.496. The optimizer, whose preview includes the daily pattern and the announced rain, raises the price at the priming decision at 16:00, reaches 2.0 at 17:22 and peaks at 3.79 min with 108.3 abandonments at a mean of 1.497, and the table peaks at 3.76 min with 118.8 and 58 moves. As first built, the case ran at the reference demand, where even the base fare peaked at only 4.04 min.

> ✦ DID YOU KNOW? **More Trips, Far More Abandoned**
>
> From the simulator: in the rain case the base fare completes 8528 trips and the PI 8346, yet the base fare abandons 1925.0 requests against 116.6 (Table 9.5). The PI's price turns riders away before they request, and the model does not count them as abandoned.

9.6.6 What a Cap Removes

The eighth case adds 40 minutes of rain to the stadium and caps the price at 1. Every legal multiplier is then 1, so the PI, the table and the optimizer produce the same run, a peak pickup of 11.83 min at hour 1.10 with 9.31 drivers idle, 3,338.1 abandonments and 84.8 % of the time below 8 min. With the cap raised to 2.5, as the briefing suggests, the optimizer with preview starts to surge at hour 0.37, reaches the cap at 0.40 and still peaks at 9.02 min with 656.2 abandonments, the PI peaks at 10.02 with 1,123.1 and the table at 9.27 with 884.3, and all three spend 0.45 to 0.625 h in the goose chase. With no cap the optimizer climbs to 3.7 and holds 5.23 min with 88.1 abandonments, while the PI still peaks at 10.02 min, because it reacts after the crowd has drained the idle pool. A cap of 2.5 therefore puts the 8 min limit out of reach of every rule here, although no rule in this case comes near the 15 min that the design plan asked the cap case to exceed.

9.6.7 Whose Objective

The ninth case gives the stadium to the optimizer with a weight of 3 on driver earnings, for a mean multiplier of 2.470, earnings of $33.56 an hour averaged over the four hours and 4,522 trips, against 1.396, $25.72 and 5,804 trips with that weight at zero. A weight of 3 on gross fares instead drives the price to the ceiling of 6 for 0.47 h, a mean of 3.483, $192,540 of fares and 4,074 trips, and a weight of 3 on completed trips gives a mean of 1.191 and 6,194 trips for $91,772. The run that pays drivers most, $40.25 an hour under the fares weight, also grosses the most and completes the fewest trips, and the run that charges riders least completes the most, so each weight buys its own outcome.

9.6.8 A City Day

The challenge runs 24 hours from midnight with the daily pattern, rain from 17:00 to 19:00 at the peak of the rush and a stadium at 23:00, decisions every 2 minutes and a quantum of 0.1. It asks for a pickup below 8 min at least 95 % of the day, at most 420 abandonments and a mean multiplier of at most 1.3. Table 9.6 compares ten configurations through Reset, as the briefing prescribes. The same settings chosen before Run give identical trajectories, including constant prices applied at the priming decision. Every automatic run keeps the pickup below 8 min all day. The shipped PI with a 5 min target and a cap of 2 fails on abandonment with 538.3. Its cap never binds, and the run is identical with the cap off. Its target is the problem, because the PI first raises the price at 17:16, inside the rain, and 222.6 of its abandonments come before 17:00 and 178.3 during the rain, against 70.4 during the rain with a 3.5 min target, which passes.

The suggested optimizer passes with 356.6 abandonments, 63.4 inside the limit, at a mean of 1.198, but the blind optimizer with the cap in place passes as well, with 354.1 at 1.194 and the cap binding for 0.20 h, so the event calendar is not what wins the day, and the table passes too. As first built, the mean limit was 1.4, and a constant 1.4 passed with 398.3 abandonments. At 1.3 a constant 1.4 fails on the mean and a constant 1.3 on abandonment (608.8), so no constant price passes, which the application now checks, and every automatic run that passed still passes. Every passing run

Table 9.6 The 24-Hour Challenge in Ten Configurations

Setting	below 8 (%)	aband.	mean m	peak τ_p (min)	
PI, $\tau^* = 5$, cap 2 (shipped)	100.0	538.3	1.036	5.62	fail
PI, $\tau^* = 5$, no cap	100.0	538.3	1.036	5.62	fail
PI, $\tau^* = 3.5$, cap 2	100.0	384.3	1.130	4.60	pass
table, cap 2	100.0	380.2	1.094	3.84	pass
MPC, no preview, cap 2	100.0	354.1	1.194	3.77	pass
MPC, preview, no cap (suggested)	100.0	356.6	1.198	4.18	pass
MPC, preview, no cap, $\tau^* = 3.5$	100.0	348.1	1.212	3.97	pass
MANUAL, $m = 1$	91.3	2878.7	1.000	9.43	fail
MANUAL, $m = 1.3$	100.0	608.8	1.300	6.88	fail
MANUAL, $m = 1.4$	100.0	398.3	1.400	5.00	fail

abandons between 1.25 and 1.33 % of its requests, against 1.25 % in the reference at the base fare, so the limit of 420 is met by holding pickups near the reference and partly by the requests the price removes.

9.7 Technology in Practice

The loop of Section 9.5 reads the pickup time of Equation (9.2) exactly and acts through a multiplier that both sides feel at the decision. Figure 9.5 draws it with the parts of a real platform in place of the schematic's symbols.

What Measures It

No instrument on a street measures a pickup time. The platform estimates it for each request from the positions that drivers' phones report and from a routing engine that adds up the traversal times of the road segments on the best path, a naive estimate that a learned model can refine [19]. The positions are good to a few meters under open sky and degrade among tall buildings [20]. In the model the pickup time is the zone average of the nearest-neighbor law of Equation (9.2), free of noise and of rounding, and the counts that the table and the optimizer read come from the platform's books, which Section 9.3 treats as exact.

> **TECHNOLOGY** **Sensor: A Pickup Estimate From Phone Positions**
>
> The pickup time a platform reads is an estimate. A routing engine divides the road network into short segments with traversal times, finds the best path from the driver to the rider and adds the times. A learned model can then correct that naive estimate for the route actually taken and the time lost near the ends of the trip [19]. The driver's position comes from the phone's satellite receiver, typically accurate to within 4.9 m under open sky. Near buildings, bridges and trees, blocked and reflected signals degrade it and can show a car in the wrong place [20], so the estimate can move while the car does not. One platform's wait estimate was continuous but shown to riders rounded to whole minutes [13].

What Moves It

The final element is the multiplier printed in the quote and carried into the driver's pay. The quantum Q and the move cap Δ of Equation (9.7) stand for the discrete steps riders see and for the business rules that hold back sharp moves, and the floor of 1 is the base fare. The model pays every trip that ends at the multiplier in force, the proportional pay that the survey cited below takes as the common setting.

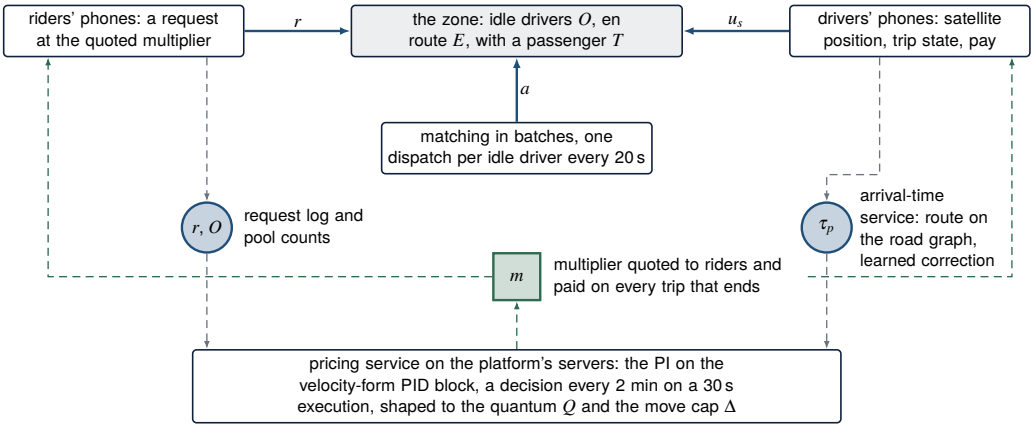


Figure 9.5 The pricing loop with the parts of a real platform in place of the schematic's symbols. The drivers' phones report positions from which an arrival-time service estimates the pickup time τ_p, the platform's books give the requests r and the idle pool O, and a pricing service sets the multiplier m that riders see in their quote and drivers in their pay. Notice that the controller never moves a car: it acts only through the price, at once on the riders and through the pay on the drivers.

> **TECHNOLOGY** **Final Element: The Multiplier in the Quote**
>
> The actuator is a number in the rider's quote. One platform's surge algorithm monitored rider demand and available drivers and placed a multiplier on the base price when demand outstripped supply at the base price, never lowering the price below the base fare. It computed a continuous value, riders saw discrete steps with 1.2 the lowest, and business rules overrode the algorithm in some cases, mainly to keep riders and drivers from very sharp price changes [13]. Driver pay is commonly taken as roughly proportional to the fare, so one price acts on demand and on supply together [21].

Where the Controller Runs

The controller runs as software on the platform's servers, not in a device in the field. A typical ride-sharing system has five modules: pricing, matching, repositioning, pooling and routing. Pricing sits upstream of the others as a macro-level lever on the balance of supply and demand, adjusting prices in real time, while matching usually batches open requests and available drivers within windows of fixed length and pairs them at predefined review times [21]. The sources read here give no period for either module. In the simulator the matching cycle is 20 s, the controller executes every 30 s, and the PI of Equation (9.8) is the framework's velocity-form PID block, which acts only at a decision of the calendar, every 2 minutes as shipped.

> **IN PRACTICE** **A Pay Floor Divided by Utilization**
>
> New York City's taxi regulator adopted a minimum pay rule for app-dispatched drivers in December 2018. Each trip must pay at least \$0.631 a mile and \$0.287 a minute, divided by the company's utilization rate: the time its drivers spend carrying passengers over the time they are available for dispatch. At the industry average of 58 percent, a 7.5-mile, 30-minute trip pays at least \$23.00. A company whose drivers wait longer for a dispatch pays more per trip, and the regulator assesses each company's utilization on a regular basis and publishes the adjusted rates [8].

What Is Modernizing It

Three developments touch this loop. The first is the learned arrival-time estimate: a residual network that corrects the routing engine's estimate lowered the mean and median absolute errors in offline

experiments and online tests, and it was built for low-latency, high-throughput serving [19]. The second is pricing optimized jointly with matching or with the repositioning of idle cars, for which reinforcement learning is a promising direction because the interaction of price with supply and demand is hard to model explicitly. The third is the matching window itself, now chosen by learning methods that trade the wait before a match against the pickup after it [21]. The optimizer of Section 9.5 is the model-based counterpart of the second: it plans the price on the simulator's own flow function and leaves matching fixed.

9.8 Limitations

The zone is one aggregate market of fluid pools without streets, strategy, noise or a rider queue, and every coefficient is a teaching value. Drivers are paid the multiplier in force on every trip that ends, so the earnings respond to a price at once and the fleet's lag starts at the decision. The abandonment count leaves out riders whom the price deters, which flatters high prices, and the optimizer reads the platform's pools without noise, plans with the declared coefficients and searches locally.

> **OPEN CHALLENGE** **Elasticities No Simulator Knows**
>
> A pricing loop is only as good as its picture of how riders and drivers answer a price. A survey of learning methods for ride-sharing found that no public simulator supported pricing decisions and that the simulators of pricing papers make strong assumptions on passenger and driver price elasticities. The supply response shapes every sequence of price decisions, yet it was not yet widely considered in learning approaches to pricing, and no public simulator was faithful enough for a policy trained in it to go straight to production [21]. This chapter's supply elasticity and lag are likewise not taken from a published estimate. Progress would be rider and driver behavior learned from data.

The challenge does not reward foresight over a blind optimizer, and its cap never binds for the shipped PI. Choosing a setting before Run includes it in the priming decision, so startup comparisons agree with Reset.

> **OPEN CHALLENGE** **A Price or a Queue at the Peak**
>
> A rising price is not the only answer to a peak. A platform can also queue requests when no vacant car is near, and in some markets the queue is perceived as more socially acceptable than surge pricing. A queue raises its own open questions: how large an area each queue should cover, whether that area should change over time, and when incoming requests should start to queue [21]. A model that answers them would let a price and a queue be compared on the same zone, the comparison this chapter's scorecard makes among pricing rules only.

9.9 Conclusion

This chapter's simulator writes surge pricing as a feedback loop on a market whose throughput map folds. A multiplier of 2 removes 29.3 % of the requests at the decision and adds 12.3 % to the fleet, the sixth root of the price, and the fleet's response starts at once and overshoots to 913.2 drivers. At a fixed fleet of 800 the throughput peaks at 2,128 trips an hour with 65.7 idle drivers, and past that point three times the reference demand completes fewer trips than twice it, while with the fleet free a multiplier of 1.2 raises the trips at three times from 2,563 to 3,200. A stadium at the base fare puts 41 % of the fleet on empty pickups and the pickup at 10.2 min, a PI deciding every 2 minutes holds it to 5.36 min and one deciding every 10 minutes does not, and preview helps only when the price is slow, cutting the abandonments from 98.7 to 75.0 at 0.1 per decision and from 75.3 only to 75.2 at 1. In the capped case a cap of 2.5 keeps every rule above 8 min.

The surge-off latch, the optimizer's move limit and the challenge's mean limit were corrected in the application during this work, and the rain case moved onto the rush. A fuller model would add zones with relocation, a rider queue, earnings paid at the accepted price and a challenge that

rewards foresight. The simulator makes the price that keeps a ride-hailing zone working a measurable property of its fold.

KEY POINTS

- A price reaches riders at once and drivers only through their pay, and at the defaults the fleet follows only the sixth root of the multiplier.
- Past the fold a zone completes fewer trips as demand grows, and there a higher price buys trips instead of costing them.
- A higher price shortens the pickup, so the process gain is negative and the direct-acting PI raises the price as the pickup rises.
- A PI deciding every 2 minutes holds the stadium to 5.36 min, and one deciding every 10 minutes lets the zone into the goose chase.

Where else this applies. Chapter 8 holds a freeway merge short of the same kind of fold with a ramp signal.

References

[1] G. Gallego and G. van Ryzin, "Optimal dynamic pricing of inventories with stochastic demand over finite horizons," *Management Science*, vol. 40, no. 8, pp. 999–1020, 1994.

[2] S. Banerjee, R. Johari, and C. Riquelme, "Pricing in ride-sharing platforms: A queueing-theoretic approach," in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. New York, NY, USA: ACM, 2015, p. 639.

[3] G. P. Cachon, K. M. Daniels, and R. Lobel, "The role of surge pricing on a service platform with self-scheduling capacity," *Manufacturing & Service Operations Management*, vol. 19, no. 3, pp. 368–384, 2017.

[4] K. Bimpikis, O. Candogan, and D. Saban, "Spatial pricing in ride-sharing networks," *Operations Research*, vol. 67, no. 3, pp. 744–769, 2019.

[5] H. Wang and H. Yang, "Ridesourcing systems: A framework and review," *Transportation Research Part B: Methodological*, vol. 129, pp. 122–155, 2019.

[6] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for mobility-on-demand systems," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 839–854, 2012.

[7] R. Zhang, F. Rossi, and M. Pavone, "Model predictive control of autonomous mobility-on-demand systems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1382–1389.

[8] New York City Taxi and Limousine Commission, "Notice of promulgation: Rules amending provisions regarding driver income and vehicle lease transparency," Adopted December 4, 2018, 2018, accessed 2026-11-20. [Online]. Available: https://www.nyc.gov/assets/tlc/downloads/pdf/driver_income_rules_12_04_2018.pdf

[9] R. Arnott, "Taxi travel should be subsidized," *Journal of Urban Economics*, vol. 40, no. 3, pp. 316–333, 1996.

[10] J. C. Castillo, D. Knoepfle, and G. Weyl, "Surge pricing solves the wild goose chase," in *Proceedings of the 2017 ACM Conference on Economics and Computation*. New York, NY, USA: ACM, 2017, pp. 241–242.

[11] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.

[12] J. Cramer and A. B. Krueger, "Disruptive change in the taxi business: The case of Uber," *American Economic Review*, vol. 106, no. 5, pp. 177–182, 2016.

[13] P. Cohen, R. Hahn, J. Hall, S. Levitt, and R. Metcalfe, "Using big data to estimate consumer surplus: The case of Uber," National Bureau of Economic Research, Cambridge, MA, NBER Working Paper 22627, 2016.

[14] B. Schaller, "Entry controls in taxi regulation: Implications of US and Canadian experience for taxi regulation and deregulation," *Transport Policy*, vol. 14, no. 6, pp. 490–506, 2007.

[15] R. C. Larson and A. R. Odoni, *Urban Operations Research*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[16] M. K. Chen, "Dynamic pricing in a labor market: Surge pricing and flexible work on the Uber platform," in *Proceedings of the 2016 ACM Conference on Economics and Computation*. New York, NY, USA: ACM, 2016, p. 455.

[17] K. J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2006.

[18] J. D. Hedengren, R. Asgharzadeh Shishavan, K. M. Powell, and T. F. Edgar, "Nonlinear modeling, estimation and predictive control in APMonitor," *Computers & Chemical Engineering*, vol. 70, pp. 133–148, 2014.

[19] X. Hu, T. Binaykiya, E. Frank, and O. Cirit, "DeeprETA: An ETA post-processing system at scale," arXiv:2206.02127, 2022.

[20] GPS.gov, National Coordination Office for Space-Based Positioning, Navigation, and Timing, "GPS accuracy," 2026, accessed 2026-11-20. [Online]. Available: https://www.gps.gov/gps-accuracy

[21] Z. Qin, H. Zhu, and J. Ye, "Reinforcement learning for ridesharing: An extended survey," arXiv:2105.01099; published in Transportation Research Part C 144 (2022) 103852, 2021.

9.10 Exercises

9.1 Concept check · 2 min

The pickup-time loop of Equation (9.8) forms its error as $e_k = \tau_{p,k} - \tau^*$. In the terms of Section 1.4.3, which statement is right?

(a) Reverse acting with $a = +1$, because a long pickup makes the controller push the price up

(b) Direct acting with $a = -1$, because the process gain is negative and the price must rise with the PV

(c) Reverse acting with $a = +1$, because near the reference a higher price lengthens the pickup time

(d) Direct acting with $a = -1$, because the completed trips fall whenever the multiplier rises

9.2 Concept check · 2 min

At three times the reference demand, raising the multiplier from 1 to 1.2 raises the steady completed trips from 2563 to 3200 an hour. Which explanation is right?

(a) The price attracts about a fifth more drivers, enough to carry the extra trips

(b) The price shortens each passenger trip, so every driver completes more trips an hour

(c) Fewer requests leave a larger idle pool, so pickups shorten and fewer drivers drive empty

(d) The price lowers the abandonment midpoint, so riders accept longer predicted pickups

9.3 Concept check · 2 min

With a largest price move of 0.1 per decision, event preview cuts the abandonments of the stadium case from 98.7 to 75.0. With the shipped move of 1 per decision it cuts them only from 75.3 to 75.2. What do the four runs show?

(a) Preview is worth little here, because the optimizer's horizon already sees the stadium coming

(b) The blind optimizer with the fast move is better tuned, so its weights leave preview nothing to add

(c) A large move cap makes the price chatter, and the chatter cancels what preview would gain

(d) A forecast substitutes for actuator speed, and a price that can move fast needs little of it

9.4 Theory and calculation · 5 min

Use Equations (9.2) and (9.3) with their defaults. The minimum pickup is $\tau_{min} = 2\,\text{min}$, the geometry coefficient is $a_p = 0.5\,\text{h drivers}^{1/2}$, which is $30\,\text{min drivers}^{1/2}$, the abandonment midpoint is $\tau_{ab} = 10\,\text{min}$ and its scale is 1.5 min.

(a) Find the idle pool at which the pickup reaches the 8 min limit of the challenge. *Checkpoint:* 25 drivers.

(b) Find the share of requests that riders abandon at that pickup. *Checkpoint:* 20.9 %.

(c) During the stadium release 4800 requests an hour reach the zone at the base fare. With 25 drivers idle, say which bound of Equation (9.3) limits the dispatches, and find the abandonments an hour. *Checkpoint:* The riders' acceptance binds, about 1001 an hour.

9.5 Theory and calculation · 7 min

At steady state with little abandonment, the completed trips equal the requests, $c \approx r = \lambda_0 m^{-\eta_d}$, and the online fleet equals its target N^* of Equation (9.4).

(a) Show that N is proportional to $m^{(1-\eta_d)\eta_s/(1+\eta_s)}$, and evaluate the exponent at the defaults $\eta_d = \eta_s = 0.5$. *Checkpoint:* 1/6.

(b) The reference holds 796.66 drivers and 1200 requests an hour at $m = 1$. Predict the requests and the settled fleet at $m = 2$, and compare with the 848.5 and 894.81 that the chapter reports. *Checkpoint:* 848.5 an hour and 894.2 drivers.

(c) Find the exponent and the factor at $m = 2$ for a supply elasticity of 1, and the exponent for a demand elasticity of 1 with $\eta_s = 0.5$. *Checkpoint:* 1/4 and 1.189, then 0.

9.6 Theory and calculation · 7 min

The fixed-fleet throughput map of Equation (9.6) with the pickup law of Equation (9.2) has its fold where $\partial q/\partial O = 0$. Work in minutes, with $\tau_t = 15\,\text{min}$, $\tau_{min} = 2\,\text{min}$ and $a_p = 30\,\text{min drivers}^{1/2}$.

(a) Show that for $O > 0.25$ the fold satisfies $\tau_t + \tau_{min} + a_p O^{-1/2} = (N - O)\,\frac{1}{2} a_p O^{-3/2}$. *Checkpoint:* Differentiate the quotient.

(b) For $N = 800$, show that $O = 65.66$ satisfies the condition, and find the largest trip rate and the pickup there. *Checkpoint:* Both sides 20.70 min, 2128 trips an hour, 5.70 min.

(c) The reference completes 1185 trips an hour with 432 idle drivers and a fleet near 800. Find the share of the largest trip rate it uses, and say on which side of the fold it sits. *Checkpoint:* 55.7 %, the idle-rich side.

9.7 App challenge · 8 min

Load scenario 11, Reference · constant demand, which starts in MANUAL at the base fare. Before Run, set Multiplier request to 2.0. Run the full 3 h at the default seed and export the CSV file.

(a) In the first row of the file, read `requested`, `earnings` and `target`. *Checkpoint:* 848.5 an hour, \$35.70 an hour and 1126.6 drivers, within 0.1.

(b) Find the largest value of the online fleet `N` and the time at which it occurs. *Checkpoint:* 913.2 drivers at 0.55 h, within 0.2 drivers and 0.02 h.

(c) Read `N` in the last row, divide it by 796.66 and compare the ratio with the sixth root of the multiplier. *Checkpoint:* 894.8 drivers, a ratio of 1.123 against 1.122.

9.8 App challenge · 8 min

Load scenario 2, No surge and the capacity fold, and switch the header to AUTO before Run. The controller is the PI on pickup time with the shipped tuning. Set Decision calendar to Every 30 seconds and run the full 4 h at the default seed. Then set Every 2 minutes, press Reset and run again, and do the same with Every 10 minutes.

(a) For each run, read the Abandoned tile and find the largest `pickup` in the exported file. *Checkpoint:* 5.10 min and 97, 5.36 min and 99, 10.19 min and 233, within 0.02 min and 1.

(b) For the run on the 10-minute calendar, count the rows in which the idle pool `O` is below the fold's idle pool `foldO`, and convert the count to hours. *Checkpoint:* 16 rows, about 0.27 h.

Explain why the slowest calendar fails when the other two succeed.

9.9 App challenge · 9 min

Load scenario 10, Challenge · a complete city day, and run the full 24 h as shipped at the default seed.

(a) Read the tiles Under 8 min, Abandoned, Mean multiplier and Challenge. *Checkpoint:* 100.0 %, 538, 1.04 and not met.

(b) In the exported file, read `totalAbandoned` at 17:00 and at 19:00, and split the abandonments into those before the rain and those during it. *Checkpoint:* 222.6 before and 178.3 during, within 0.5.

(c) Set Pickup target to 3.5 min, press Reset and run the day again. Repeat (a), and find the abandonments during the rain. *Checkpoint:* 100.0 %, 384, 1.13, met. 70.4 during the rain.

9.10 Application area · 3 min

At the reference demand a multiplier of 2 held to steady state cuts the requests from 1200 to 848.5 an hour and the pickup from 3.44 to 3.19 min.

(a) Use Equation (9.3) to find the abandonments an hour before and after the move. *Checkpoint:* 15.0 and 9.0 an hour.

(b) Find the riders an hour whom the price turns away and the Abandoned tile never counts, and say what this means for ranking pricing rules by that tile. *Checkpoint:* 351.5 an hour.

9.11 Application area · 4 min

Scenario 8 caps the price at 1 during a stadium release with rain. Under the optimizer with preview, Table 9.5 gives a peak pickup of 11.83 min with 3338.1 abandonments at the cap of 1, and 9.02 min with 656.2 at a cap of 2.5.

(a) Explain why the PI, the table and the optimizer produce the same run at the cap of 1. *Checkpoint:* Every legal price is 1.

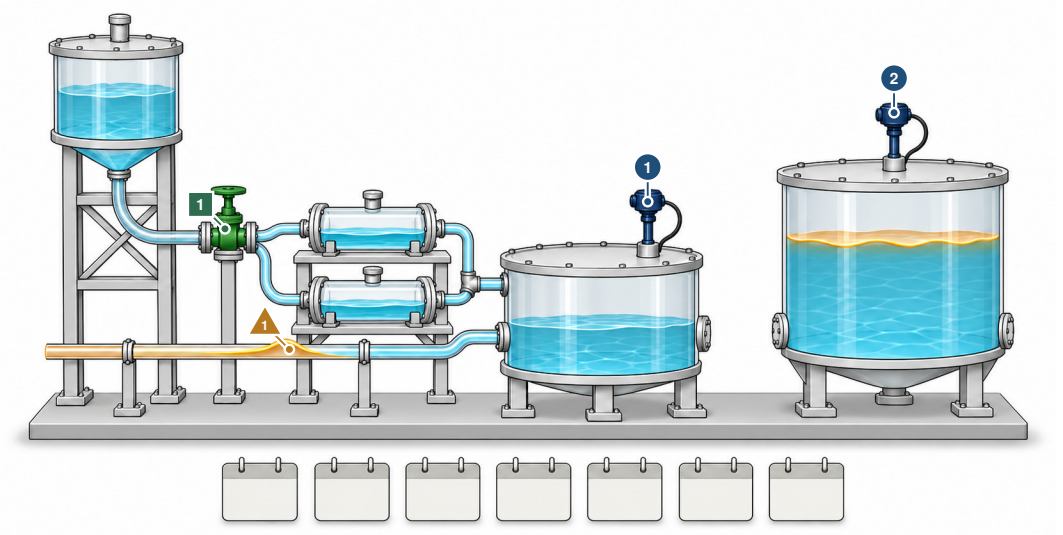
(b) A regulator weighs a cap of 2.5 against a cap of 1 for such events. Find what the higher cap saves in abandonments, and say why the 8 min limit stays out of reach. *Checkpoint:* 2682 abandonments, about 80 %.

9.12 Review the agent · 5 min

An AI assistant is asked to make the pickup-time PI safer and replies as follows. “The steady-state gain from multiplier to pickup is −0.49 min per unit at the reference demand and −10.0 min at three times it, twenty times larger. With $K_c = 0.45$ fixed, the loop gain grows from 0.22 to 4.5, far beyond any gain margin. Schedule the gain on measured demand as $K_c = 0.45 \times 0.49/|K_p|$, which gives 0.022 at three times the demand and holds the loop gain at 0.22, so the PI stays well damped through a stadium release.” Find the flaw in the proposal and say how it could be checked.

CHAPTER 10

Phillips's Stabilization Policy, Budget Lags and a Folding Public Debt



The controller moves 1 the budget, decided once a year and delivered through two slow pipes, spending and transfers

The controller reads 1 the published output gap, read late and with errors 2 the public debt as a share of output

What disturbs it 1 a fall or rise in private demand

Notice the two delay vessels between the budget valve and the output tank: a budget set once a year reaches demand only through them, while the debt tank keeps every deficit. Illustration, not to scale.

IN PLAIN TERMS Steering an Economy With a Yearly Budget

The main idea. When people and firms spend less, a country makes less than it could. A government can then spend more, and some help comes by itself as taxes fall and benefits rise.

The challenge. The government sees a slump late and sets its budget once a year, so the money can come after the recovery and cause a boom. The extra spending is borrowed, and a debt whose interest outruns growth keeps growing by itself.

Why it matters. The timing of a budget decides whether it calms a slump or makes the swings larger.

THREE WAYS TO READ THIS CHAPTER

⊘	**Explore**	11 min	the first page, the history of the debt brake and the fact on interest rates (Section 10.2), the try-it segments of cases 4 and 6 and the fact on case 7 (Section 10.6), and the key points
🕮	**Study**	50 min	the whole chapter with its three pause points and the twelve exercises at its end
△	**Practice**	20 min	the first page, the results (Section 10.6), Technology in Practice (Section 10.7), Limitations (Section 10.8) and the open challenge

The control question A treasury decides its budget once a year on late figures, spends it the year after, and adds every deficit to a debt whose ratio to output has dynamics of its own. How much feedback do those lags and that calendar allow, and what does holding output up cost the debt?

Learning objectives

- Compute the multiplier and the offset that the automatic stabilizers leave as built-in proportional feedback.
- Explain why an annual budget halves the proportional gain that monthly decisions allow, and how a per-decision cap bounds the resulting cycle.
- Predict a debt path when the interest rate exceeds growth, and the offset of a proportional rule on the debt.
- Locate the fold of a debt with a risk premium, and judge whether a bounded consolidation can stop a spiral past it.
- Compare PID, rules and MPC on one crisis, and identify a constraint that acts procyclically.

Prerequisites Sections 1.4.2 (proportional offset), 1.6.2 (velocity form), 1.4.8 (derivative action), 1.4.9 (windup) and 1.4.11 (sample time). Section 1.6.3 is needed only for the predictive controller. Chapter 9 uses the same calendar, quantum and cap.

The app https://apmonitor.com/apps/fiscal-stabilization/

See also Volume 5, Chapter 9 for a stock refilled through a capped, year-late contribution. Volume 9, Chapter 4 for offset-free MPC with a disturbance state in the estimator, the role the private shock plays here.

10.1 Introduction

A recession that begins a quarter into the year barely shows in the treasury's figures when its budget is adopted that October. In the simulator of this chapter, private investment starts to fall at year 0.25, the true output gap is −1.15 % at the October budget of year 0.75, and the published gap reads only −0.51 %. A rule that asks for one point of stimulus per point of published gap raises the budget to 2.5 % a year later, the most one decision may add, and to 2.9 % at the October of year 2, when the published gap reads −2.89 % and the true one has already recovered to −1.46 %. The money reaches demand during the recovery. The slump's trough improves only from −6.23 % to −5.48 %, while the recovery, which peaks at +0.97 % with no budget at all, becomes a boom of +4.71 % at year 4.04.

Economists wrote this problem as feedback control before the vocabulary of control was common in their field. In 1954 stabilization policy was posed as proportional, integral and derivative correction of the gap between production and its desired level [1], in 1957 the same author asked what the lags of real policy do to that correction [2], and he had already built a hydraulic machine of the economy to show its dynamics [3, 4]. By 1973 it was a starting point that such policies could make a multiplier-accelerator model oscillate or go unstable, and stabilization was then set as a tracking problem in optimal control on a linearized model [5]. Two later results made the public debt part of the loop. The primary surplus of the United States rises with its debt ratio [6], which is a proportional feedback law on a measured state, and the stance of fiscal policy toward the debt was named active or passive in the same terms as monetary policy [7]. The feedback, its lags and the stock it accumulates are the subject matter, which is why control is the natural frame for it.

The structure is general. The controller acts through a decision calendar and a delivery lag, so its useful gain is bounded by delay as in any loop with dead time [8], and the integral of its own output is a stock with dynamics of its own, the debt, which here is an integrating state whose self-regulation changes sign. A plant with the same shape appears wherever holding one variable draws down a store, such as a battery's charge spent holding a vehicle's speed, a reservoir released to hold a flow downstream, or a propellant budget spent holding an orbit. Volume 4, Chapter 14 spends a vault of reserves to hold an exchange rate in its band. The belief this plant corrects is that a loop is finished once its error is gone. In the third case of the simulator an integral term brings the gap to zero against

a lasting fall in demand by holding the budget at 2.0 % of output for good, and the debt that pays for it climbs from 90 % to 111.6 % of output in twelve years. A second belief, that more feedback stabilizes better, fails too. On clean data, deciding once a year allows half the proportional gain that the same lags allow when the budget is decided every month, 0.76 against 1.53.

However, the work that estimates fiscal reaction functions, debt limits and interest-rate effects does so on historical data [6, 9, 10], where the budget calendar, the recognition lag and the rule are given. Such data rarely offer an economy in which a minister, Phillips's three terms, a debt rule, a brake and an optimizer each decide the same budget through one calendar and can be compared run against run. This chapter works with such an economy, built as a browser simulator, and measures what the lags, the calendar and the debt's own arithmetic do to each way of deciding. The multiplier-accelerator, Phillips's policy, Bohn's rule, the fold of a debt with a premium and model predictive control all come from the literature cited below. The simulator joins them in a continuous economy in which a budget passes through a recognition lag, a calendar with a quantum and a cap and two delivery pipelines before it moves output, while the debt integrates every deficit at an interest rate that depends on the debt. Every equation, constant and limit is stated in this chapter, and the app carries nine cases, a reference and a crisis challenge, each measured on the running engine.

The chapter is organized as follows. Section 10.2 traces the domain from the hydraulic machine to the debt brake. Section 10.3 writes out the economy, its budget pipelines and the debt ratio, and Section 10.4 describes its behavior before any decision, with the stabilizers as droop, the multiplier-accelerator cycle and the fold of the debt. Section 10.5 sets out the six ways the budget is set, from the minister's hand through Phillips's PID to a predictive controller, and Section 10.6 reports the measured cases, including the gain each calendar allows and the crisis challenge. Section 10.7 places the loop in the institutions of a real budget, Section 10.8 states the limitations, Section 10.9 summarizes the results, and the exercises follow in Section 10.10.

10.2 From Water Tanks to Debt Brakes

The oscillation the simulator starts from was written down in 1939, when the multiplier, which turns spending into income, was combined with the principle of acceleration, which turns the growth of income into investment [11]. Their interaction gives a second-order economy whose own response can oscillate, and its continuous form, with a production lag and an investment lag, is the plant of this simulator.

A decade later the dynamics were built in water. A. W. Phillips described mechanical models of economic dynamics in 1950 [3], and his machine was a hydraulic model of the British economy about 7 feet high, 5 feet wide and 3 feet deep, in which red water flowing through clear plastic tubes stood for the circular flow of income [12]. It used hydraulic components to simulate a dynamic system where other analog computers of the period used electrical or mechanical ones, and it showed the dynamic behavior of an economy to students and to professional economists alike [4]. The simulator's main view draws the same tanks and pipes, with the output tank read against a potential mark.

Phillips then wrote government into the machine as a controller. In 1954 the policy correction applied to the gap in production had three parts, one in proportion to the gap, one to its accumulated sum and one to the speed at which it was changing [1], the same three terms as the PID of Section 1.4.2, and in 1957 the paper on the time-forms of lagged responses asked how the shape of the lags between a decision and its effect changes what that correction does [2]. In 1973 Phillips's finding that such policies could produce undesired oscillations or instabilities in multiplier-accelerator models was the opening premise of a study that posed stabilization as a dual tracking problem in optimal control, with any limit on the states or the controls expressed as a higher cost on its deviation [5]. That treatment of limits as prices is the one the simulator's MPC uses for its deficit limit and its debt ceiling. The objection to such designs followed within three years. Relationships estimated under one policy can shift when the policy changes, because the people whose behavior they summarize respond to the policy [13], and that critique sets the first limitation of this chapter.

The debt entered the loop later. In 1991 a government's fiscal stance toward its debt was classified as active or passive alongside the stance of its central bank [7]. Evidence of corrective action followed in 1998. Standard tests cannot reject a unit root in the debt-to-output ratio of the United States, but its primary surplus is an increasing function of that ratio, so the ratio reverts to its mean once war-time spending and the cycle are controlled for [6]. Built-in stabilization was measured in the same years. Over 1962 to 1995 individual federal taxes in the United States offset perhaps as much as 8 % of initial shocks to output [14], which is the droop of Section 10.4 measured on a real tax code.

Rules came next. The Swiss debt brake of 2003, a ceiling on spending written into a constitution, is the subject of the history box below [15].

> ⧖ HISTORY **A Spending Ceiling in a Constitution, 2001**
>
> Swiss voters approved a debt brake for the federal government by a majority of 85 percent in a mandatory referendum in 2001, and it first governed the federal budget of 2003. The constitution requires income and expenditure to balance over time, caps total expenditure at revenue adjusted for the cycle, and requires spending above the cap in the final accounts to be made up in later years. Federal debt fell from 28.2 percent of output at the end of 2002 to 19.5 percent in 2012. A review after ten years found that the binding ceiling on expenditure was the rule's most important feature, and that the fall in nominal debt came in a favorable economy that the budgets had not anticipated, mainly through revenue and growth that were underestimated [15].

The price of borrowing joined the problem as a function of the debt. Estimates from projected deficits put the effect of one point of projected deficit-to-output ratio at roughly 25 basis points on long-term rates [9]. When a government's ability to raise its primary balance cannot keep pace with rising debt, the debt has an endogenous limit beyond which it cannot be rolled over, and data on 23 advanced economies from 1970 to 2007 show a fiscal reaction function of that shape [10]. The fold of this simulator is a static, deterministic cousin of that limit. The low interest rates of the 2010s reopened the other side. Safe rates below growth rates were shown to be more the historical norm than the exception for the United States, so that a debt may be rolled over without later taxes, while equilibria in which investors who believe a debt risky raise its cost by demanding a premium remain a relevant argument against high debt [16]. The fifth case of the simulator raises the interest rate above growth, and the sixth starts past the fold.

> ✦ DID YOU KNOW? **Rates Below Growth in Four Economies, 2019**
>
> When a lecture on public debt was published in 2019, 10-year government bond rates were about 3 percent in the United States, 1.3 in the United Kingdom, 1.2 in the euro area and 0.1 in Japan, against forecasts of 10-year nominal growth of about 4, 3.6, 3.2 and 1.4 percent [17]. Case 5 reverses the sign.

10.3 The Machine, Its Budget Pipelines and the Debt

The simulated economy is closed and real, and every flow and stock is a fraction of potential output, a year's worth for a flow. Output Y is a fraction of potential, so the output gap is $y = Y - 1$, and time is in years. The economy is drawn in Figure 10.1 in the manner of the application's MONIAC view, and Table 10.1 classifies its signals.

10.3.1 Production, Households and Demand

Production follows demand through Phillips's production lag, and households spend a share of disposable income, as shown in Equation (10.1).

$$\begin{aligned} \tau_p \dot{Y} &= D - Y, \qquad D = C + I_{tot} + G + X + d \\ C &= c\,(Y - T + tr), \qquad T = \bar{T} + t\,y \\ tr &= tr_0 - s_{tr}\,y + G_t + tr_{ex}, \qquad G = G_0 + G_s \end{aligned} \tag{10.1}$$

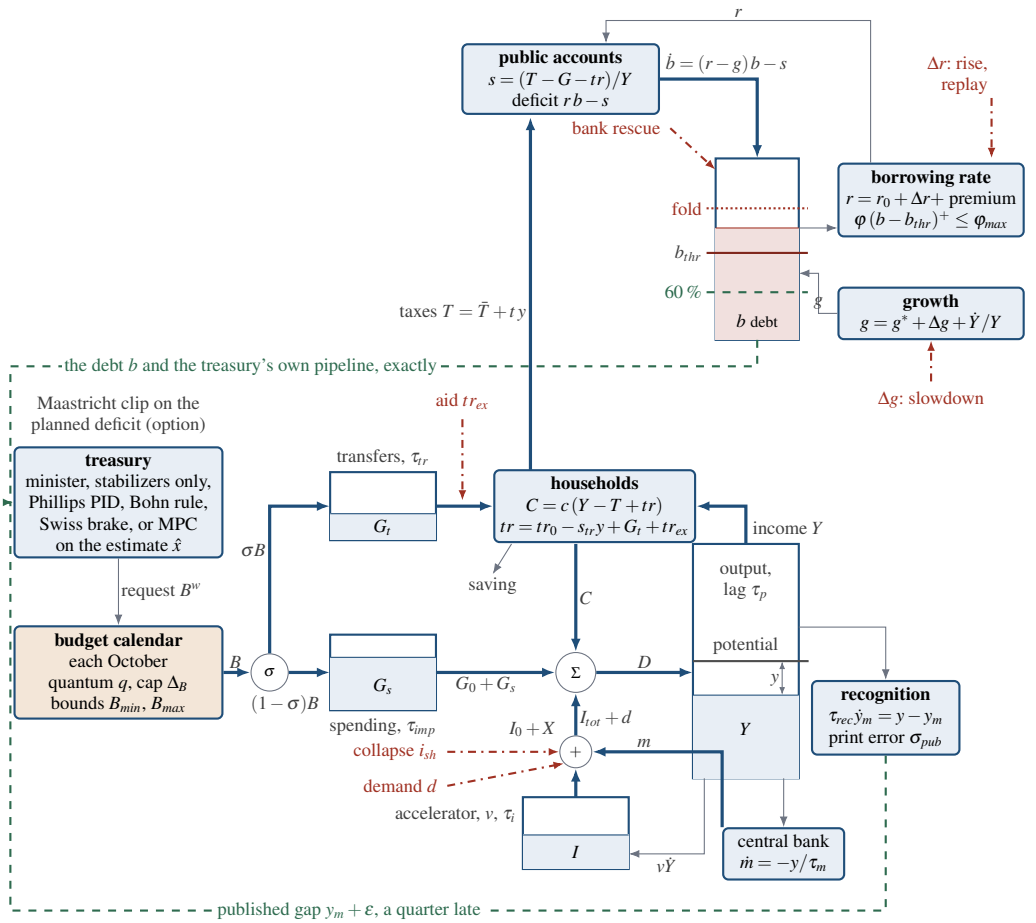


Figure 10.1 The economy as Phillips drew it, not to scale. Demand D, the sum of consumption C, government spending $G_0 + G_s$, private demand $I_{tot} + d$ and the constant X, fills the output tank, which is read against potential, so the gap y is the distance between the level and the mark. Income Y returns to households, which pay taxes T to the public accounts, receive benefits tr and spend C. The accelerator's induced investment I and the central bank's offset m join private demand. The treasury's request B^w passes the calendar's quantum q, cap Δ_B and bounds, and the budget B splits into spending G_s and transfers G_t, each with its own delivery lag. The primary balance s and the borrowing rate r fill the debt ratio b, marked with the 60 % reference (dashed), the premium threshold b_{thr} (solid) and the fold (dotted). Red dash-dotted arrows are disturbances, and the dashed path is all the treasury sees.

where $\tau_p = 0.25\,\text{yr}$, $c = 0.75$, $\bar{T} = 25\,\%$ of output is the tax revenue at potential, $t = 0.25$ the marginal tax rate, $tr_0 = 5\,\%$ the benefits paid at potential, $s_{tr} = 0.3$ the benefits added per point of negative gap, and $G_0 = 20.9\,\%$ the government spending at potential. The constant $X = 1 - c\,(1 - \bar{T} + tr_0) - I_0 - G_0 = 2.1\,\%$ is fixed so that demand equals potential output at $Y = 1$ with no policy and no shock. The disturbance d is other private demand, a lasting shift from a slider or the lockdown of a pandemic, and tr_{ex} is exogenous transfers, the emergency aid of a pandemic or a demographic ramp. The terms t and s_{tr} are the automatic stabilizers. They act at once and without any decision, so they are feedback built into the plant.

Table 10.1 Signals of the Economy and the Treasury by Role

Symbol	Role	Default	Range
y	CV, the true output gap, judged but never seen	0	—
$y_m + \varepsilon$	measured, the published gap, PV of the PID	0	SP 0
b	CV of the debt rules, measured exactly	90 %	start 0–200 %
B	MV, the discretionary budget in force	0	−5–8 %
B^w	request, from the minister's slider or a controller	0	−6–10 %
i_{sh}, d	DVs, investment shock, other private demand	0	events, ±6 %
tr_{ex}	DV, exogenous transfers (pandemic aid, ageing)	0	events
R	DV, a bank rescue added to the debt	0	10 %
Δr, Δg	DVs, borrowing-rate and growth shifts	0, 0	−2–6, −2–1
Y, I, m	states, output, induced investment, offset	1, 0, 0	—
G_s, G_t, y_m	states, spending and transfer pipelines, recognized gap	0	—
I_d, I_y	states, integrals of the deficit and of the gap	0	—
s, r, g, u	outputs, primary balance, rate, growth, unemployment	−0.9, 1, 2, 4 %	—

Flows and stocks in percent of potential output, rates in percent a year.

10.3.2 The Accelerator and the Central Bank

Firms invest in proportion to the growth of output, through a lag of their own, and an optional central bank adds private demand while output is below potential, as shown in Equation (10.2).

$$\tau_i \dot{I} = v\dot{Y} - I, \qquad I_{tot} = I_0 + I + m + i_{sh}, \qquad \dot{m} = -y/\tau_m \tag{10.2}$$

where $v = 0.6$ yr of output growth is invested, $\tau_i = 1$ yr, $I_0 = 17\,\%$ is autonomous investment, i_{sh} is the investment shock and $\tau_m = 4$ yr. With the offset switched off, $\dot{m} = 0$, which is Phillips's 1954 economy or a member of a currency union. The offset is the application's one addition to its original design. Without it a permanent cut of 1 % of output in spending leaves a permanent gap of –1.51 % and improves the primary balance by only 0.16 %, which the application's check measures, so no debt could be reduced without a lasting slump. The offset is an integral on the gap, not a model of monetary policy, and the borrowing rate does not follow it.

10.3.3 The Budget, Its Pipelines and Recognition

The decided budget B is held between decisions and reaches the economy through two first-order pipelines, while the treasury sees the gap through a recognition lag, as shown in Equation (10.3).

$$\begin{aligned} \tau_{imp}\dot{G}_s &= (1-\sigma)B - G_s, \qquad \tau_{tr}\dot{G}_t = \sigma B - G_t \\ \tau_{rec}\dot{y}_m &= y - y_m \end{aligned} \tag{10.3}$$

where $\tau_{imp} = 1$ yr is the implementation lag of spending, $\tau_{tr} = 0.25$ yr that of transfers, σ the share of the budget paid as transfers, zero by default, and $\tau_{rec} = 0.25$ yr the recognition lag. The published gap is $y_m + \varepsilon$, where ε is a print error with standard deviation σ_{pub}, zero by default and 0.5 % in the fourth case, drawn once a quarter from the engine's seeded generator so that a reset replays it.

10.3.4 The Debt Ratio

The debt ratio integrates the primary deficit at an interest rate less the growth rate, as shown in Equation (10.4).

$$\begin{aligned} \dot{b} &= (r-g)\,b - s, \qquad s = (T - G - tr)/Y \\ r &= r_0 + \Delta r + \min\left(\varphi_{max},\ \varphi\,(b - b_{thr})^+\right) \\ g &= g^* + \Delta g + \dot{Y}/Y \end{aligned} \tag{10.4}$$

where s is the primary balance, $r_0 = 1\,\%$ the real borrowing rate without a premium, $g^* = 2\,\%$ a year potential growth, $\varphi = 0.03$ points of premium per point of debt above $b_{thr} = 100\,\%$, capped at $\varphi_{max} = 10$ points, and Δr and Δg the shifts that a slider, an interest-rate rise of two points, a

Table 10.2 Constants of the Economy, With M for a Parameter of the Model Dialog and S for a Value Fixed in the Source

Symbol	Meaning	Value	Range	Class
c	propensity to consume	0.75	0.5–0.9	M
$\bar{T}$, tr_0	taxes and benefits at potential	25, 5 %	15–45, 0–15	M
G_0	spending at potential	20.9 %	10–35	M
I_0	autonomous investment	17 %	5–30	M
τ_p	production lag	0.25 yr	0.05–1	M
τ_i, v	investment lag, accelerator	1, 0.6 yr	0.25–3, 0–1.5	M
τ_m	central bank's integral time	4 yr	1–15	M
τ_{rec}	recognition lag	0.25 yr	0.02–1	M
τ_{imp}, τ_{tr}	spending and transfer lags	1, 0.25 yr	0.02–2, 0.02–1	M
σ_{pub}	print error, a new draw each quarter	0	0–1 %	M
b_0	debt at the start	90 %	0–200	M
r_0, g^*	rate without premium, potential growth	1, 2 %	−2–8, 0–5	M
φ, b_{thr}	premium slope, threshold	0.03, 100 %	0–0.3, 30–200	M
φ_{max}	largest premium	10 points	2–30	M
u^*, β	unemployment at potential, Okun	4 %, 2	2–8, 1–3.5	M
X	other autonomous demand	2.1 %	computed	S
—	investment collapse, boom	−2.5, +3 %	1, 1, 3 or 2 yr	S
—	pandemic lockdown, aid	−10.5, 10 %	one year	S
—	bank rescue, demographic ramp	10 %, 0.2 %/yr	20 yr	S
—	rate rise, growth slowdown	+2, −1 points	for good	S
—	bad quarter, deficit limit	±2 %, 3 %	—	S
—	market access lost	250 %	—	S
Δt	integration step, control period	1/48, 1/12 yr	—	S
—	log rate, seed	48 per yr, 20261005	—	S

growth slowdown of one point or the replayed policy rate of the inflation-targeting application make. The overall deficit, the quantity that the Maastricht limit of 3 % applies to, is $rb - s$. The ratios use $\max(Y, 0.2)$ in place of Y, a floor far outside any run. Unemployment is $u = u^* - y/\beta$, with y in percent, $u^* = 4\,\%$ and $\beta = 2$ points of gap per point of unemployment. Two integrals, $\dot{I}_d = rb - s$ and $\dot{I}_y = y$, give the exact calendar-year deficits and quarterly gaps, interpolated to the boundary inside a step. A quarter whose average true gap is outside ±2 % is a bad quarter, a calendar year whose deficit exceeds 3 % is a deficit year, two quarters of falling output raise RECESSION, and a debt above 250 % raises MARKET ACCESS LOST.

10.3.5 Shocks, the Start and the Integrator

Each event shock is a pure function of the time since it started. An investment collapse lowers i_{sh} linearly by 2.5 % over a year, holds it a year and recovers over three, and an investment boom raises it 3 % in the same shape with a recovery of two years. A pandemic sets $d = -10.5\,\%$ and $tr_{ex} = 10\,\%$ for one year. The demographic ramp adds 0.2 % a year to tr_{ex} for twenty years. A bank rescue adds $R = 10\,\%$ to the debt at the end of the step in which it fires, and a financial crisis is a collapse and a rescue together. A case starts at $Y = 1$ with every other state at zero and $b = b_0$, which is an equilibrium for the reference budget because $s = \bar{T} - G_0 - tr_0 = -0.9\,\%$ is exactly $(r_0 - g^*)\,b_0$ at $b_0 = 90\,\%$, with an overall deficit of 1.8 %, the growth rate times the debt. The application's check finds every derivative within 7×10^{-18} of zero and nothing moving over ten years. The engine of Section 1.6.1 advances the nine states by classical fourth-order Runge-Kutta in steps of $\Delta t = 1/48$ year, four to a month so that the monthly controller lands on the grid. It logs a row after every step, and the default seed is 20261005. Halving the step leaves both verdicts of the challenge unchanged and moves its final debt by 1.0×10^{-5} of output. Table 10.2 gathers the constants.

10.3.6 Assumptions

Expectations look backward. Households spend out of today's disposable income and do not anticipate the taxes a debt implies, firms invest on today's growth, and the premium is a fixed function of today's

debt. The model has no inflation and so no anchoring of expectations. This breaks down whenever the public learns the rule, which is the subject of Section 10.8.

Output is set by demand. Potential grows at g^* whatever happens, and only the central bank's offset brings output back to potential, as an integral that repays the area of a slump with a boom of the same area. An economy with a supply side or with prices that adjust would recover by itself.

One closed, real economy. There is no exchange rate, no inflation to erode the debt and no maturity structure, so the whole debt pays today's rate. The deficit is real, which makes its 3 % limit looser than a nominal one. The replayed policy-rate path is added one for one to the real borrowing rate as a stress test, although the rate in the inflation application rose because inflation did.

Teaching values. Every coefficient, lag, threshold and shock size is a declared teaching value. The premium slope of 0.03 points per point and the Bohn coefficient of 0.05 are orders of magnitude the application attributes to the empirical literature [6, 9], and none of them is estimated here.

10.4 What the Economy Does Before Anyone Decides

10.4.1 Stabilizers as Droop

At a steady state $\dot{Y} = 0$, so induced investment is zero and the gain from a lasting change in spending to output is the multiplier of Equation (10.5).

$$\frac{dY}{dG_s} = \frac{1}{1 - c\,(1-t) + c\,s_{tr}} = \frac{1}{0.6625} = 1.509 \tag{10.5}$$

Without the stabilizers it would be $1/(1-c) = 4$. The application's check measures 1.509 on the engine after thirty years of a step with the accelerator removed, and a budget paid entirely as transfers measures 1.132, which is c times the multiplier, because a transfer reaches demand only after households have spent their share of it. Each point of gap changes taxes by t and benefits by s_{tr}, so the stabilizers are a proportional controller from the gap to the net tax, of gain 0.55, built into the plant as a governor's droop is built into an engine. Like droop they leave an offset, the proportional offset of Equation (1.3). A lasting fall of 2 % in private demand settles at a gap of $-2 \times 1.51 = -3.02\,\%$ when nothing else acts.

🕮 **Pause.** Try Exercise 10.1 before reading on.

10.4.2 The Multiplier-Accelerator Cycle

Linearized at potential with no policy and no offset, output and induced investment have the eigenvalues $-0.625 \pm 1.503i$ per year, a damped cycle of 4.18 yr, and the application's check finds the same period within 0.003 years in the free response to a one-quarter demand impulse. The benefit response does much of the damping. Without it the pair moves to $-0.175 \pm 1.311i$. The original design of the application set the investment lag at half a year, where the same accelerator is unstable at zero policy, with eigenvalues $+0.075 \pm 2.301i$, so the lag was set to a year and the economy rings but settles.

10.4.3 The Debt's Own Dynamics and the Fold

With output at potential and no premium, Equation (10.4) is linear in b with the eigenvalue $r - g$. The check confirms it at -0.005, 0 and $+0.005$ per year for r_0 of 1.5, 2 and 2.5 %. While $r < g$ a debt fades under any fixed budget, and when $r > g$ it grows by itself. A fixed primary balance s holds the debt still where $h(b) = s$, with

$$\begin{aligned} h(b) &= \big(r(b) - g\big)\,b \\ h'(b) &= r_0 - g^* + \varphi\,(2b - b_{thr}), \quad b > b_{thr} \end{aligned} \tag{10.6}$$

the surplus that keeps each ratio constant at potential, and the debt is stable where $h'(b) < 0$. Below the threshold h is a line of slope -0.01. Above it the premium turns h up, and the same budget gains a second, unstable equilibrium, the fold, as shown in Figure 10.2(a). For the reference balance of

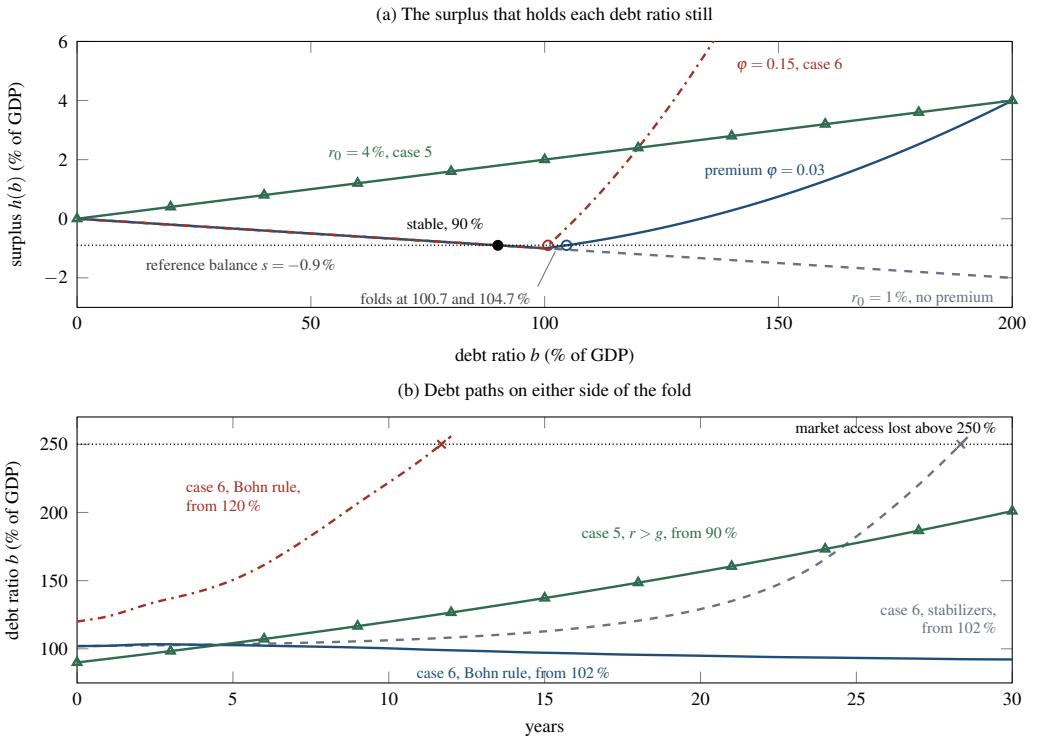


Figure 10.2 The debt's own dynamics, from the application's functions and engine. (a) The surplus $h(b)$ that holds each debt ratio still at potential, for the reference economy without a premium (gray dashed), with the shipped premium (blue solid), with the case-6 premium (red dash-dotted) and at $r_0 = 4\,\%$ (green with triangles), against the reference balance of –0.9 % (thin dotted). Each crossing is an equilibrium, stable where h falls through the line and a fold where it rises. (b) The debt ratio over thirty years for the case-6 economy from 102 % with the stabilizers alone (gray dashed) and with the shipped Bohn rule (blue solid), from 120 % with the rule (red dash-dotted), and for case 5 from 90 % (green with triangles), each start set in the Model dialog before Run. The crosses mark the loss of market access at years 11.69 and 28.35.

–0.9 % the equilibria are 90 % and 104.67 %, and the debt's eigenvalue at the fold is +0.023 per year. With the case-6 slope of 0.15 the fold moves down to 100.71 %, close enough to the threshold that a debt starting at 102 % with the stabilizers alone drifts slowly, to 106.3 % at year 10 and 129.1 % at year 20, and then runs away, losing market access at year 28.35, while one starting at 100 % falls slowly, to 97.4 % at year 30, as shown in Figure 10.2(b). Bisecting the starting debt on the engine with the stabilizers alone places the threshold at 100.71 %, the analytic fold. The original design's values, $r_0 = 2\,\%$ and a threshold at 60 %, would have put the reference debt of 90 % on the unstable branch, with $h'(0.9) = +0.036$, which is why the application moved both.

10.5 Six Ways to Set the Budget

The treasury's code runs once a month. At each run it draws a fresh print error if a quarter has closed, advances its estimate of the economy, obtains a request B^w from whichever controller is selected, clips that request when the Maastricht option is on, and then lets the calendar decide whether the month is a budget month. The minister and the automatic controllers share every step after the request. Table 10.3 lists the settings.

Table 10.3 Treasury Settings, Defaults and Ranges. A Marks an Advanced Control

Symbol	Setting	Default	Range
Budget and built-in feedback			
—	controller in AUTO	stabilizers only	5 controllers
—	calendar	each October	3 calendars
σ	transfer share of the budget	0	0–1
t, s_{tr}	marginal tax rate, benefit response	0.25, 0.3	0.1–0.5, 0–0.7
—	central bank closes the gap	on	on, off
q, Δ_B	quantum, largest change (A)	0.1, 2 %	0–1, 0.5–10
B_{min}, B_{max}	deepest cut, largest stimulus (A)	−5, 8 %	−10–0, 0–12
—	Maastricht deficit clip (A)	off	off, on
Phillips PID			
K_c	budget per point of gap	0.5	0–3
τ_I, τ_D	integral and derivative times	8, 0 yr	0–20, 0–2
Rules on the debt			
ρ_B, b^*	Bohn coefficient, debt target	0.05, 90 %	0–0.5, 0–150
K_{br}, T_{rep}	brake stimulus per point, repayment time	0.3, 3 yr	0–2, 1–12
η	brake band, share of spending (A)	0.06	0.02–0.2
MPC			
ν	move penalty	2	0.1–50
b_{ceil}	terminal debt ceiling	100 %	40–160
H	horizon (A)	8 yr	2–12
—	deficit limit (A)	strict	strict, soft, off
—	information (A)	estimator	estimator, truth

A disturbance slider shifts private demand by up to ±6 % of output.

10.5.1 Calendar, Quantum and Cap

Both the calendar and the rounding of each budget are those of `core/cadence.js`, a module of the framework of Section 1.6. The same module paces the price of Chapter 9 and the policy rate of Chapter 11. The annual calendar has one instant a year at $t = k + 0.75$, each October, the quarterly calendar one every quarter from $t = 0$, and the continuous calendar makes every monthly run a budget. A run becomes a budget if some calendar instant has passed since the previous budget, so an instant between two runs is taken at the later one. Every instant of the three calendars falls on a monthly run, so each budget is taken on time. The request then becomes a budget through Equation (10.7).

$$\begin{aligned} \tilde{B} &= q\left[B^{w}/q\right], \qquad \Delta_q = q\max\left(1, \lfloor \Delta_B/q \rfloor\right) \\ B_k &= \mathrm{clip}_{[B_{min}, B_{max}]}\left(B_{k-1} + \mathrm{sat}_{\Delta_q}\left(\tilde{B} - B_{k-1}\right)\right) \end{aligned} \tag{10.7}$$

where $[\cdot]$ rounds a tie away from zero, B_{min} and B_{max} are moved inward onto the grid before the clip, sat limits a move to $\pm\Delta_q$, and an undefined request leaves the budget where it was. With $q = 0.1\,\%$ and $\Delta_B = 2\,\%$, a budget moves in tenths of a point of output and by at most two points a year. The check finds every budget of nine runs, three calendars by three controllers through the crisis of case 8, on its calendar instant, on the grid, inside the cap and inside the bounds. A switch of mode or of controller during a run holds the budget in force at the next decision, and one made before the run starts does not.

10.5.2 The Minister and the Stabilizers

In MANUAL the request is the minister's slider, from −6 % to 10 % in steps of 0.1 %, and the calendar shapes it as it shapes a controller's. While a controller decides, the minister's slider shows what that controller requested before any rounding, capping or clipping, which the check confirms in case 6 at a Bohn coefficient of 1, with a request of −45.7 % against an applied −4 %. With the stabilizers selected the request is zero and only t and s_{tr} act.

10.5.3 Phillips's PID

The PID is the framework's shared controller in its velocity form, the algorithm of Section 1.6.2, run every month on the published gap against a set point of zero, as shown in Equation (10.8).

$$\begin{aligned} e_k &= -p_k, \quad P_k = K_c e_k, \quad D_k = -K_c \tau_D \left(p_k - p_{k-1}\right)/T_c \\ \Sigma_k &= \Sigma_{k-1} + \left(P_k - P_{k-1}\right) + \left(D_k - D_{k-1}\right) \\ B_k^w &= \Sigma_k + \mathscr{I}_{k-1}, \qquad u_k = \mathrm{clip}_{[B_{min}, B_{max}]}\left(B_k^w\right) \\ \mathscr{I}_k &= \mathscr{I}_{k-1} + \left[K_c e_k/\tau_I + \left(u_k - B_k^w\right)/\tau_I\right] T_c \end{aligned} \tag{10.8}$$

where $p_k = y_{m,k} + \varepsilon_k$ is the published gap at execution k, $T_c = 1/12$ year, and $K_c = 0.5$ and $\tau_I = 8\,\mathrm{yr}$ by default. In the symbols of Equations (1.23) to (1.26) the action is $a = +1$, the bias is zero, Σ_k is S_k, B_k^w is v_k, $\mathscr{I}_k$ is I_k and the tracking time is τ_I, so only the output that leaves the controller is clipped and only the integral is corrected back toward it. A negative gap gives a positive error and a stimulus, which the application's direction check confirms with a request of +0.72 % at the monthly run one month after the recognized gap is set to −2 %. The request handed to the calendar is the unclipped B_k^w. The derivative is taken on the published gap without a filter, $\tau_I = 0$ removes the integral, and there is no back-calculation without one. The integral is not told about the quantum, the cap, the annual hold or the Maastricht clip, so it keeps integrating the gap between budgets, a form of windup that the back-calculation of Section 1.4.9 does not correct, since it acts only against the clip. While another controller or the minister decides, the PID resets $\mathscr{I}$ each month so that its output equals the budget in force B, which makes the switch to it bumpless, as in Section 1.4.10.

10.5.4 Bohn's Rule, the Brake and the Maastricht Clip

Bohn's rule is a proportional controller on the debt, $B^w = -\rho_B\left(b - b^*\right)$, so the primary surplus rises with the debt as the 1998 estimate found [6]. The Swiss-style brake allows a stimulus proportional to the published gap, debits a compensation account a with every budget above zero and credits it with every budget below, and never lets the account fall below its band, as shown in Equation (10.9).

$$\begin{aligned} B^w &= -K_{br}\left(y_m + \varepsilon\right) + \min(0, a)/\max(0.5, T_{rep}) \\ \bar{B} &= \left(a + A_b\right)/\max(1, T_d) \\ a &\leftarrow a - B_k T_d \quad \text{at each budget} \end{aligned} \tag{10.9}$$

where $A_b = \eta\left(G_0 + tr_0\right) = 1.55\,\%$ of output is the band, T_d the interval between budgets, one year on the annual calendar, and $\bar{B}$ the largest budget that keeps the account inside the band until the next one. A request above $\bar{B}$ is clipped to it, and a shaped budget above it is cut to the grid point below it, overriding the cap and the quantum. Credits are kept as debt repaid rather than spent again, so the bound is one-sided, and the brake is a bounded integrator, in which discretion is allowed and its sum is not. The check finds the account never below $-A_b$ in any case, at gains up to 2 and on the monthly calendar, with 0.054 % of output to spare at its closest. The Maastricht clip, when it is on, limits any controller's request to $B^w \le 0.03 - \left(r\,b - s - G_s - G_t\right)$, the budget that would keep the deficit within 3 % at today's revenue. The 60 % reference for the debt is drawn but not enforced.

10.5.5 The Treasury's Estimate

The MPC plans from an estimate $\hat{x}$, because the treasury does not know the private shock, the induced investment or the central bank's offset. Each month the estimate is stepped over T_c by fourth-order Runge-Kutta on the plant's own Equations (10.1) to (10.3), with the private shock $\hat{w}$ held, its own pipelines G_s, G_t and the debt b copied from the truth, and then corrected by the published gap, as shown in Equation (10.10).

$$\left[\hat{Y}, \hat{I}, \hat{m}, \hat{y}_m, \hat{w}\right] \leftarrow \left[\hat{Y}, \hat{I}, \hat{m}, \hat{y}_m, \hat{w}\right] + L\left(y_m + \varepsilon - \hat{y}_m\right) \tag{10.10}$$

where L is the steady-state Kalman gain of the block $[Y, I, m, y_m, w]$ linearized at potential and discretized over a month, with w a random walk, process weights 10^{-7}, 10^{-8}, 0, 0 and 2×10^{-5}, and a measurement weight of $\max(4 \times 10^{-6}, \sigma^2_{pub})$. These are teaching values. The gain is $L = (1.439, 0.748, -0.052, 0.608, 1.400)$ in the reference economy and $(0.891, 0.439, 0, 0.464, 0.655)$ in case 4, where the offset is off and the prints are noisy. The private shock w plays the role of the added disturbance state of Section 1.6.3, as a fuel bias does in the offset-free controller of Volume 9, Chapter 4. A perfect-state setting hands the MPC the true states instead, as a labeled comparison.

10.5.6 Eight Budgets as a Quadratic Program

At every budget, and whenever a setting has made its plan stale, the MPC linearizes the economy at $\hat{x}$ and the budget in force by central differences, with the private shock as an eighth state that it expects to fade, $\dot{w} = -w/\tau_w$ with $\tau_w = 2\,\text{yr}$. The linear model, with the affine term the linearization leaves, is converted to quarterly steps, $T_s = 0.25\,\text{yr}$, by an exact zero-order-hold discretization, and the MPC solves the quadratic program (QP) of Equation (10.11) with the framework's linear MPC, `core/linmpc.js`, which Section 1.6.3 describes.

$$\begin{aligned} \min\ & \textstyle\sum_{k=1}^{N} T_s \left[(\hat{Y}_k - 1)^2 + \lambda_B B_k^2 \right] + \nu \textstyle\sum_j (B_j - B_{j-1})^2 \\ & \quad + \rho_d\, \varsigma_d^2 + \rho_b\, \varsigma_b^2 \\ \text{s.t. } & B_{min} \le B_j \le B_{max}, \qquad |B_j - B_{j-1}| \le \Delta_B \\ & r_k b_k - s_k \le 0.03 + \varsigma_d, \qquad b_N \le b_{ceil} + \varsigma_b, \qquad \varsigma \ge 0 \end{aligned} \tag{10.11}$$

where $N = H/T_s = 32$, the budget is held over blocks of four quarters, one per annual budget, B_{-1} is the budget in force, $\lambda_B = 0.2$, $\nu = 2$, the deficit row is the linearized deficit at every quarter, and its slack price ρ_d is 400 for the strict setting, 4 for the soft one and absent when the limit is off, while the ceiling's ρ_b is 2000. The deficit limit is therefore soft in every setting, with a breach priced so high in the strict one that the plan avoids it whenever it can, which is how a limit is written as a cost [5]. On the annual calendar the program has 10 variables, eight budgets and two slacks, and 67 rows. It is solved by the dual active-set method of [18]. The plan's first budget is the request that the calendar then rounds and caps, and if a solve were to fail the budget in force would be kept. Case 8 needs 13 solves, one at the start and one each October, none fails, and the twelve October solves take at most two active-set iterations, 0.25 on average.

10.5.7 Economy, Belief and the Models Behind Them

Equations (10.1) to (10.4), with the shocks, the print errors and the premium, are the simulated economy. The treasury's model is the same equations without knowledge of the private shock, corrected by one published number a month, and linearized with a shock it expects to fade over two years and no foresight of the next one. The models both stand for are the multiplier-accelerator of [11] with Phillips's lags [1] and the estimated reaction functions and premiums of [6, 9, 10], whose coefficients are not copied here. The application estimates no model from data, and the PID, Bohn's rule and the brake carry no model of the economy.

10.6 Ten Cases on One Machine

Table 10.4 lists the cases. Each is loaded from the dropdown, which applies the defaults of Tables 10.2 and 10.3 with the changes its row names, and each run covers the years its row gives, on the default seed. A variant is the case with the named settings changed before Run, and the application gives the same run when the settings are carried through Reset instead, since Reset keeps the reader's settings (Section 1.6.1). A starting debt set in the Model dialog before Run takes effect from the start. Table 10.5 collects the outcomes on the true gap, which the treasury never sees. The quarterly count and the deficit years are those of the scorecard.

Table 10.4 Cases of the Simulator With the Crisis Challenge and the Reference

#	Change from the reference, run length	Teaching point
1	MANUAL, offset off, investment collapse at 0.25 yr, 10 yr	a budget on late figures
2	offset off, demand −2 % for good at 1 yr, 12 yr	stabilizers as droop
3	2 with the PID, K_c 1.5, no integral, monthly, lags 0.02 yr	Phillips 1954
4	2 with the PID, K_c 1.5, τ_I 2 yr, σ_{pub} 0.5 %, 24 yr	Phillips 1957
5	r_0 4 %, no premium, no shock, 30 yr	$r > g$
6	Bohn rule, φ 0.15, b_0 120 %, no shock, 20 yr	the fold
7	brake, investment boom at 1 yr, collapse at 6, 14 yr	a bounded integrator
8	MPC, financial crisis at 2 yr, 12 yr	a procyclical limit
9	pandemic at 1 yr, 12 yr	a decade of debt
C	financial crisis at 2 yr, stabilizers only, 12 yr	three limits at once
R	no shock, 10 yr	the equilibrium

Table 10.5 Lowest and Highest True Gap, Bad Quarters, Deficit Years and Final Debt Over Each Case's Own Run

Case and configuration	y_{min} (yr)	y_{max} (yr)	Qtr	Yr	b_{end}
1, no budget	−6.23 (2.02)	0.97 (5.42)	10	2	95.7
1, K_c 1, no integral	−5.48 (1.83)	4.71 (4.04)	21	2	100.2
1, 2 % at the first October	−3.80 (1.62)	3.83 (5.50)	32	2	94.7
2, as shipped	−5.22 (2.21)	0.00	43	11	112.9
2, s_{tr} 0	−9.90 (2.63)	0.00	39	8	109.7
2, offset on	−4.21 (1.92)	0.45 (3.79)	6	2	94.0
3, P only	−1.28 (1.31)	0.00	0	11	112.2
3, τ_I 2 yr	−1.27 (1.27)	0.00	0	11	111.6
4, as shipped	−5.88 (2.04)	5.08 (4.81)	59	22	148.1
4, K_c 0.5, τ_I 4 yr	−5.15 (2.00)	2.22 (3.98)	17	22	142.5
4, the same and τ_D 0.5 yr	−7.89 (2.15)	4.46 (19.96)	43	22	139.1
5, stabilizers	0.00	0.00	0	30	201.0
5, Bohn ρ_B 0.05	−0.55 (14.38)	0.00	0	30	150.0
5, Bohn ρ_B 0.2	−1.78 (5.50)	0.14 (18.23)	0	7	104.9
6, Bohn ρ_B 0.05	−2.29 (2.44)	0.00	10	20	lost 11.7
6, ρ_B 1, B_{min} −10 %	−6.34 (5.75)	0.00	31	20	lost 15.2
6, stabilizers	0.00	0.00	0	20	lost 10.5
7, brake	−6.08 (7.60)	5.42 (2.40)	25	3	91.0
7, Bohn rule	−5.09 (7.58)	5.79 (2.46)	22	3	90.2
7, stabilizers	−4.96 (7.58)	5.72 (2.46)	23	3	90.3
8, MPC, strict limit	−6.29 (3.69)	1.64 (7.04)	10	1	85.2
8, MPC, soft limit	−4.77 (3.46)	1.27 (5.67)	7	1	96.0
8, PID	−4.36 (3.38)	3.76 (5.17)	11	1	100.4
8, PID, Maastricht clip	−4.97 (3.50)	2.92 (6.35)	14	1	97.2
8, MPC, strict, offset off	−9.30 (4.48)	0.00	21	3	105.3
9, stabilizers	−6.31 (1.92)	5.30 (3.23)	9	1	99.3
9, Bohn rule	−6.36 (1.96)	4.35 (3.25)	9	1	95.8

Gap in % of potential at the year given, debt in %. "Lost" is the year the debt passes 250 %. Qtr: quarters outside ±2 %. Yr: calendar years over a 3 % deficit.

10.6.1 By Hand, and by Nobody

In case 1 the minister holds the budget while investment collapses from year 0.25. Left at zero, the gap bottoms at −6.23 % at year 2.02 and the recovery peaks at +0.97 % at year 5.42, with 10 bad quarters. The recognized gap passes −1 % only at year 0.91, after the first October. The rule of the Introduction, one point of stimulus per point of published gap, sets 0.5 % at that October, 2.5 % at the next, capped, and 2.9 % at the October of year 2, when the published gap reads −2.89 % and the true one −1.46 %. A year later the published gap is +3.09 %, the true one +4.25 %, and the cap lets the budget fall only to 0.9 %. Twenty-one quarters are outside the band, and the economy is still swinging at year 10, at −3.77 %. A minister who asks for 2 % at the first October and leaves it there gets the shallowest trough of the three, −3.80 %, and then a lasting boom with 32 bad quarters, because with the central bank off nothing ever takes the stimulus away.

Table 10.6 The Largest Gain Whose Case-4 Cycle Decays, by How Often the Budget Is Decided

Integral time τ_I	Continuous time	Every month	Each October
none	1.85	1.53	0.76
4 yr	1.55	1.31	0.70
2 yr	1.28	1.11	0.63

Recognition 0.25 yr, implementation 1 yr, no print error, no quantum, a 10 % cap. Engine columns judged on years 32 to 40 against 12 to 20.

Case 2 removes the decision altogether. Private demand falls 2 % for good at year 1, the gap reaches −5.22 % at year 2.21 and settles at −3.02 %, the droop offset of Section 10.4. Without the benefit response the trough is −9.90 % and the accelerator rings. The stabilizers are paid for with a deficit of 4.21 % at year 12 against 1.8 % before the shock, so the debt rises from 90 % to 112.9 % by year 12 and 162.4 % by year 30. With the central bank's offset on, the gap closes to −0.07 % at year 12, only 6 quarters are bad, and the debt ends at 94.0 %.

10.6.2 Phillips 1954 and 1957

Case 3 hands the same shortfall to proportional policy deciding every month with lags of a week. At $K_c = 1.5$ the gap falls to −1.28 % at year 1.31 and approaches the steady-state offset $-2/(0.6625 + 1.5) = -0.925$ %, which the check confirms within 0.003 points after twenty years. At year 12 the gap is −0.90 % and the budget 1.4 %. An integral time of two years removes the offset by year 12 with the budget at 2.0 %, the whole shortfall, which is Phillips's 1954 result. The integral is not free. The deficit stays at 4.41 % and the debt reaches 111.6 % by year 12, because with no central bank, holding output up against a lasting shortfall means paying for it every year.

Case 4 keeps those gains and restores the lags of a real budget, a quarter of recognition, an October calendar and a year of implementation, with a print error of 0.5 %. The loop never settles. The gap swings between −5.88 % and +5.08 %, with peaks at years 9.12, 13.81, 18.10 and 22.12, 4.33 years apart on average, and 59 of 96 quarters are bad, as shown in Figure 10.3. Linearized in continuous time, the same gains are stable without the lags, with a slowest eigenvalue of -0.319 per year, and unstable with them, at $+0.063 \pm 2.545i$, a growing cycle of 2.47 years, as the application's check confirms. The cycle the engine shows is longer, because budgets come once a year, and bounded, because the cap holds it. On the shipped prints 19 of the 24 budgets move the full 2 %. Without the print error 23 of 24 do, the budget cycles through 0, 2, 4 and 2 % with a period of exactly four years, and 58 quarters are bad. Raising the cap to 10 % lets the same loop grow until the budget touches both bounds and the gap swings from −14.76 % to +13.25 %, while a cap of 1 % holds the late swings within ±2.22 % and leaves 15 bad quarters. The per-decision cap is a rate limit that turns an unstable linear loop into a limit cycle whose size it sets.

> **TRY IT IN THE APP** **4 · Phillips 1957: The Same Gains With Lags**
>
> Load the case and press Run. Watch the output gap trend. With Phillips's gains and the lags of a real budget the gap never settles: it swings between −5.88 % and +5.08 %, the scorecard counts 59 of 96 quarters outside the band, and at year 24 the status line reads SPIRAL.

The gain the lags allow depends on how often the budget is decided, as shown in Table 10.6. Section 1.4.11 counts a sample time as about half a period of added dead time, so a budget held for a year adds about half a year of delay to a loop whose lags alone bound the proportional gain at 1.85. The continuous-time column is the boundary of the linearized loop, found by bisection on its eigenvalues. For proportional policy it is the ultimate gain of Section 1.4.7. The other two are found on the engine, with the print error and the quantum removed and the cap raised to 10 % so that none of them shapes the answer, by bisection on whether the gap's swing over years 32 to 40 is smaller than over years 12 to 20. Monthly decisions cost 13 to 17 percent of the continuous-time gain, and

an annual budget allows half to 57 percent of what monthly decisions allow, 0.76 against 1.53 for proportional policy and 0.63 against 1.11 with an integral time of two years. Case 4's gain of 1.5 is more than twice the annual limit, and the weaker policy of the case's second step, $K_c = 0.5$ with $\tau_I = 4\,\text{yr}$, lies inside it. That policy leaves 17 bad quarters, seven in the first slump and the rest in smaller swings that end by year 10.

🕮 **Pause.** Try Exercise 10.2 before reading on.

The derivative term of the third step helps only on clean data. On the noisy prints, $\tau_D = 0.5\,\text{yr}$ raises the bad quarters from 17 to 43 and the gap's RMS after year 6 from 1.03 % to 2.35 %, because each quarter's new print error becomes a jump in the measured rate and a budget swing at the next October. On seeds 1 to 10 the same two settings leave 9 to 19 and 23 to 46 bad quarters, so the ordering does not depend on the draw. Without the print error the same term lowers the RMS from 0.80 % to 0.42 % and the bad quarters from 11 to 8, since it recovers phase that the exponential lags take, and the linearized loop agrees, its slowest eigenvalue moving from -0.112 to -0.115. The expectation of the application's original design, that the derivative term would make the lagged loop worse, holds only through the data.

🕮 **Pause.** Try Exercise 10.3 before reading on.

10.6.3 When r Exceeds g, and Past the Fold

Case 5 changes nothing in the economy except the borrowing rate, now 4 % against growth of 2 %, with no premium and no shock. Output stays at potential, and the reference budget's primary deficit of 0.9 % now drives the debt from 90 % to 119.9 % at year 10, 156.4 % at 20, 201.0 % at 30 and 255.4 % at 40, the closed form $b(t) = (b_0 + \delta/(r-g))\,e^{(r-g)t} - \delta/(r-g)$ with $\delta = 0.009$ to within 10^{-13} at year 30. Holding 90 % still would take a surplus of 1.8 %. Bohn's rule at its shipped 0.05, above $r - g = 0.02$, holds the debt but far above its target, 150.0 % at year 30, 159.4 % at 40 and 175.5 % at 80, on its way to the proportional controller's equilibrium $(\delta + \rho_B b^*)/(\rho_B - (r-g)) = 180\,\%$. At 0.2 the debt settles at 104.9 %, against the predicted 105 %, with a primary surplus of 2.1 %. A proportional rule on the debt leaves an offset, as the stabilizers leave one on the gap.

Case 6 starts at 120 % with a premium of 0.15 points per point above 100 %, past the fold of 100.71 %. The rate is 5.62 % by year 2, reaches its cap of 11 % by year 8, and the shipped rule, cutting the budget by up to 5 %, loses market access at year 11.69. A coefficient of 1 with cuts of up to 10 % delays the loss to year 15.17 and takes the economy into a slump of −6.34 % on the way, and the stabilizers alone lose access at year 10.54. The application's sweep of coefficients 0.05, 0.2 and 1 against cut limits of 2, 5 and 10 % loses access in all nine runs, between years 11.3 and 15.2. This is the honest case. A cut shrinks output before it improves the balance, the smaller output raises the ratio, and the ratio raises the rate. From 102 % the shipped rule brings the debt back to 92.2 % in thirty years, and bisection on the engine places the highest start from which it recovers at 103.47 %, so the rule moves the threshold up by less than three points from the fold of the fixed budget. On the Model dialog's steps of one point the shipped rule recovers from 103 % and not from 104 %. Volume 5, Chapter 9 meets the same limit of authority in a pension fund whose deficit grows on its own while a capped contribution refills it.

> **TRY IT IN THE APP** **6 · The Fold: A Spiral No Surplus Stops**
>
> Load the case and run it. The debt starts at 120 %, past the fold, and Bohn's rule cuts the budget while the borrowing rate climbs to its cap of 11 % by year 8. Market access is lost at year 11.69. Then choose the stabilizers alone as the controller, press Reset and run again: access is lost sooner, at year 10.54.

10.6.4 Rules and an Optimizer Through a Crisis

Case 7 runs a boom from year 1 and a collapse from year 6 under the brake, whose account fills to +2.00 % at the October budget of year 2 as it cuts into the boom, and empties to −1.10 % at the

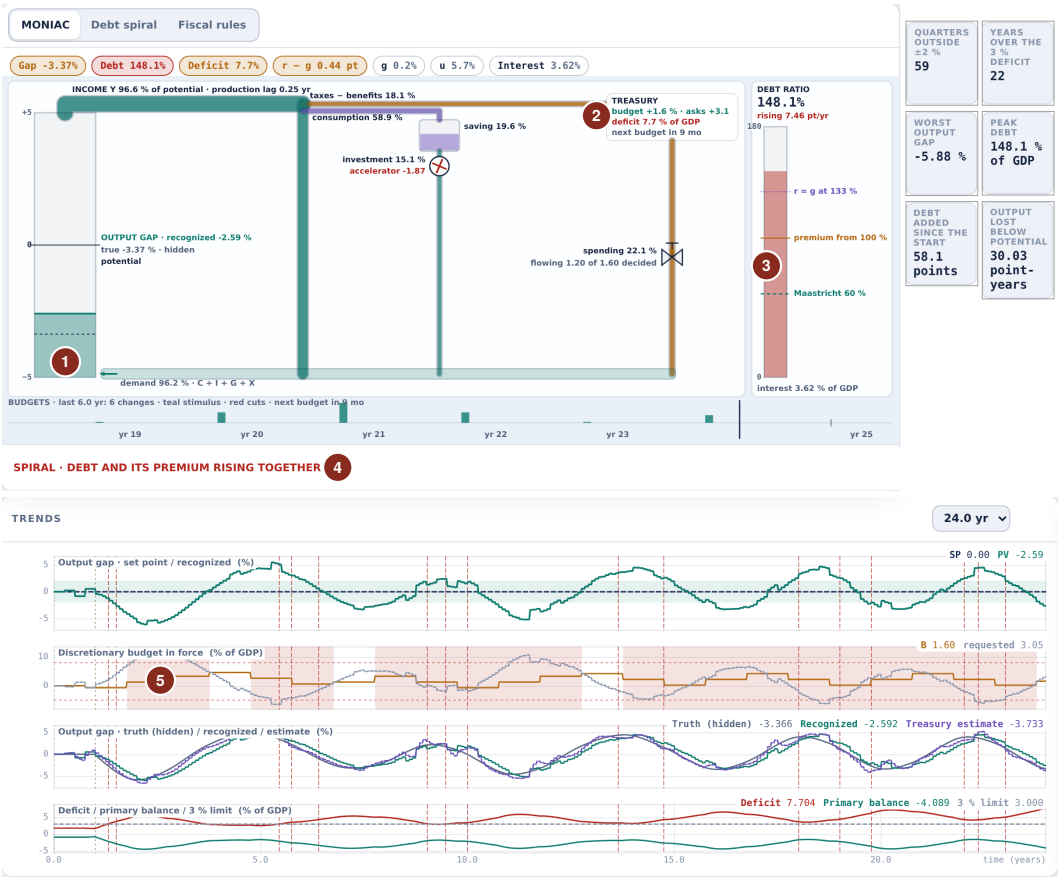


Figure 10.3 Case 4, Phillips's 1954 gains under the lags of 1957, at $t = 24.00$ yr in AUTO in the MONIAC view with a 24-year trend window, as loaded, a composite of the plant view, the trends and six scorecard tiles set in gray frames at the upper right. The badges read a gap of −3.37 %, debt 148.1 %, deficit 7.7 %, $r - g$ of 0.44 points, growth 0.2 %, unemployment 5.7 % and interest 3.62 % of output. The output tank shows the published gap of −2.59 % as its surface and the hidden true gap of −3.37 % as a dashed line, income is 96.6 % of potential, consumption 58.9 %, saving 19.6 %, taxes less benefits 18.1 %, investment 15.1 % with the accelerator taking 1.87 %, and spending 22.1 %, with 1.20 of the 1.60 points decided flowing. The treasury holds a budget of +1.6 % and its PID asks for +3.1 %, with the next budget in nine months. The debt column reads 148.1 %, rising 7.46 points a year past the premium threshold of 100 % and the level of 133 % at which r would equal g, and the status strip reads SPIRAL. The budget strip counts six changes in six years, each a full two points except the last. In the trends the published, true and estimated gaps cycle together, the budget moves in annual steps with the cap's shading over 19.0 of the 24 years by the run's record, and the deficit climbs past 7 %. The scorecard counts 59 quarters outside ±2 %, 22 deficit years, a worst gap of −5.88 %, a peak debt of 148.1 %, 58.1 points of debt added and 30.03 point-years of output lost. Callouts: (1) the output tank with the true gap dashed below the published surface, (2) the treasury's budget and request, (3) the debt column with its premium threshold, (4) the status line, (5) the budget in force with the cap's shading.

October of year 7 as it stimulates into the bust. BRAKE BINDING is first raised by the monthly run at year 7.83, the one after that budget, and the debt ends at 91.0 %. The cycle is no smaller. The brake leaves 25 bad quarters against 23 with the stabilizers alone and 22 under Bohn's rule, whose

budget stays between −0.3 % and +0.4 %. Its stimulus of 1.2 % decided at the October of year 4 was withdrawn at the October of year 5, so the withdrawal was still reaching demand as investment fell, and the bust's trough is −6.08 % against −4.96 % with no discretion. Neither rule can time a cycle through an annual budget, which is the lesson of case 4 in another form.

> ✦ DID YOU KNOW? **A Brake That Spends Into the Bust**
>
> From the simulator: in case 7 the brake leaves 25 bad quarters and a trough of −6.08 %, against 23 and −4.96 % with the stabilizers alone (Table 10.5). A stimulus decided one October and withdrawn the next was still being withdrawn when investment fell.

Case 8 is the challenge's financial crisis, a collapse and a rescue of 10 % of output at year 2, under the MPC. With the strict deficit limit the plan cuts the budget to −1.9 % at the October of year 2, when the stabilizers have already raised the deficit to 3.32 %, and to −3.9 % a year later, at a deficit of 4.75 %, a full cap's move in the middle of the slump. Both plans are bound by the deficit row of their first quarter, and even the strict price leaves it 0.14 and 0.89 points above the limit. The trough deepens to −6.29 % at year 3.69 and 10 quarters are bad, for the lowest final debt of the case, 85.2 %. With the soft limit the plan never cuts below −0.9 %, the trough is −4.77 % and 7 quarters are bad, with a debt of 96.0 %, better on both counts than the PID at its defaults, which leaves 11 bad quarters, a boom of +3.76 % and a debt of 100.4 %. The PID under the Maastricht clip is worse still, at 14 bad quarters, because the clip also cuts in the slump. With the central bank off, the strict-limit MPC drives the gap to −9.30 %. An optimizer does what its constraints say, and a deficit limit that binds in a slump is procyclical whoever enforces it.

Case 9 is a pandemic, a year of lockdown that removes 10.5 % of private demand while emergency transfers of 10 % are paid. The aid enters consumption at $c = 0.75$, offsetting 7.5 of the 10.5 points, and the rest takes the gap to −6.31 % at year 1.92 and the deficit to 16.7 %. The debt jumps from 90 % to 109.2 % by year 2, falls to 97.6 % at year 3 as output rebounds to +5.30 %, and is still 99.3 % at year 12. At $r - g = -1$ point a one-off debt fades by about one percent of itself a year, so nine points remain a decade later, or six under Bohn's rule.

10.6.5 The Crisis Challenge

The challenge repeats the crisis of case 8 with the stabilizers alone and asks for no more than six quarters outside ±2 % in twelve years, no more than two calendar years with a deficit above 3 %, and a debt at or below 100 % at year 12, judged on the truth. Table 10.7 compares nine configurations, and Figure 10.4 draws its bad quarters and final debts against the two limits. As shipped the gap falls to −4.77 % at year 3.46, total investment from 17.0 % to 13.3 % of output at year 3.08, and the debt peaks at 107.3 % at year 3.71, the rescue plus a smaller output, before ending at 99.5 %. Nine quarters are bad, six in the slump and three in the boom that follows, to +2.31 % at year 5.62, as the central bank's offset overshoots. The deficit limit and the debt target are met, so the stabilizers fail on the gap alone.

The suggested setting raises the tax stabilizer to 0.3, which acts at once, and adds the brake at 0.5 per point of gap, which leans against the slump at the Octobers of years 2 and 3 with 0.6 % and 0.9 % and then against the boom with cuts down to −1.1 %. It leaves five bad quarters, all in the slump, a trough of −4.14 %, a boom held to +1.88 %, one deficit year and a debt of 98.9 %, with the account between −1.50 % and +1.30 % against its band of ±1.55 %. Halving the step changes its final debt by 1.0×10^{-5} and neither verdict. The tax rate alone meets the challenge exactly at its limit of six quarters, and the brake alone does not, at seven. A brake gain of 1 with the same tax rate also meets it, with the smallest swings of the table, but over 24 years its cycle grows, to a swing between −1.82 % and +1.71 % in the last six years, while at 0.5 the swing has decayed to between −0.14 % and +0.37 %. Twelve years are too short to show that instability, which is case 4's lesson at a smaller gain. The original design proposed a debt target of 90 %. Of 68 settings of the MPC, Bohn's rule and

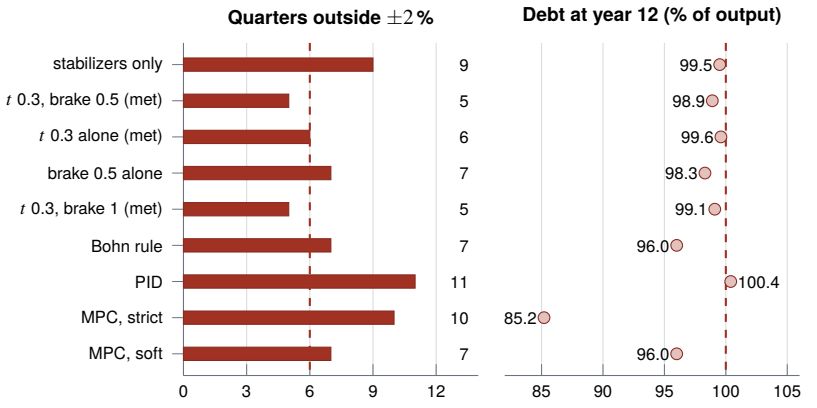


Figure 10.4 Two columns of Table 10.7, the crisis challenge over twelve years: the quarters with the true gap outside ±2 % (bars, values at the right) against the limit of six, and the debt at year 12 (dots) against the limit of 100 % (dashed lines). Notice that every configuration meets the debt limit except the PID, so the challenge is lost on the gap, and that the configuration with the lowest debt, the MPC with the strict deficit limit, has the second most bad quarters.

Table 10.7 The Crisis Challenge of Twelve Years in Nine Configurations

Configuration	Qtr	Yr	b_{12}	y_{min}, y_{max}	Target
stabilizers only, as shipped	9	1	99.5	−4.77, 2.31	not met
t 0.3 and brake K_{br} 0.5, suggested	5	1	98.9	−4.14, 1.88	met
t 0.3 alone	6	1	99.6	−4.48, 1.91	met
brake K_{br} 0.5 alone	7	1	98.3	−4.36, 2.13	not met
t 0.3 and brake K_{br} 1	5	1	99.1	−3.82, 1.30	met
Bohn rule, ρ_B 0.05	7	1	96.0	−5.19, 1.93	not met
PID, K_c 0.5, τ_I 8 yr	11	1	100.4	−4.36, 3.76	not met
MPC, strict deficit limit	10	1	85.2	−6.29, 1.64	not met
MPC, soft deficit limit	7	1	96.0	−4.77, 1.27	not met

Limits 6 quarters, 2 years and 100 % of output. Gap in %.

the brake measured on this case, none reaches it within the gap and deficit limits, and the lowest debt among them, 67.1 % under a Bohn coefficient of 0.5 with a target of 80 % and a tax rate of 0.3, comes with 24 bad quarters. Repaying a rescue of 10 % in a decade at $r - g = -1$ point needs surpluses that the gap band does not allow.

10.7 Technology in Practice

The treasury of Section 10.5 reads a published gap and an exact debt, and acts through a budget that passes a calendar, a quantum and a cap. Figure 10.5 draws the loop with the institutions of a real budget in place of the schematic's symbols.

What Measures It

Output is compiled, not measured by a device. National accounts are assembled from surveys and administrative records, published first about a month after a quarter ends and revised several times after that [19]. The gap needs potential output as well, a trend whose end-of-sample estimate is the larger source of error in real time [20]. In the model the recognition lag τ_{rec} of 0.25 yr and the quarterly print error σ_{pub} of Equation (10.3) stand for the late print and its revisions, and the debt is read exactly, as a treasury reads its own accounts.

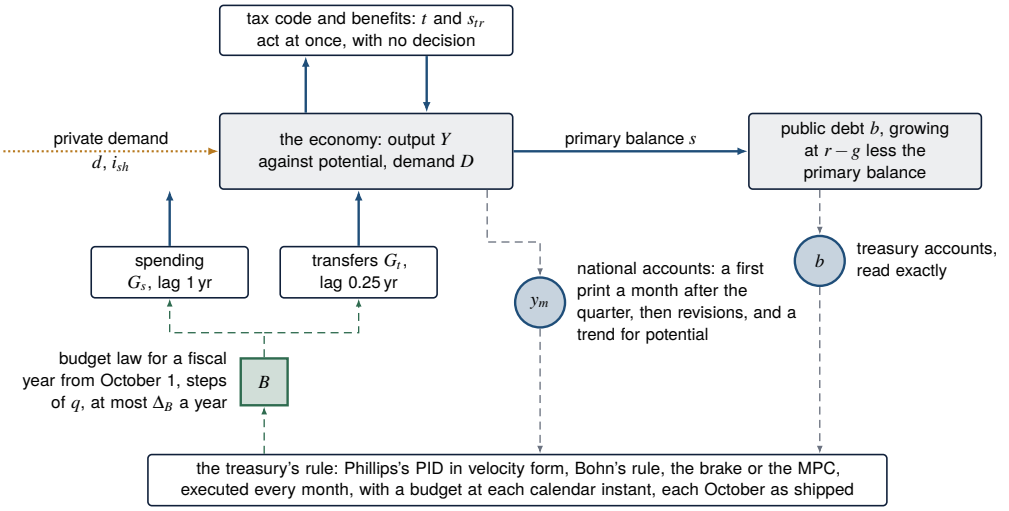


Figure 10.5 The budget loop with the institutions of a real budget in place of the schematic's symbols. National accounts and an estimate of potential output give the published gap y_m, the treasury's accounts give the debt b, and a rule sets the budget B, which reaches demand through a spending and a transfer pipeline. Notice the inner loop that needs no decision: the tax code and the benefits answer output at once, while the treasury acts once a year through two lags.

TECHNOLOGY Sensor: Output Read From Revised Accounts

No instrument reads output. Statistical agencies aggregate surveys and administrative records into national accounts. In the United States the first, or advance, estimate of a quarter's output appears about one month after the quarter ends, a second and a third estimate follow in the next months, and comprehensive revisions come later [19]. The gap also needs potential output, a trend. Revisions of real-time gap estimates have been found to be of the same order of magnitude as the gap itself and highly persistent, and the real-time estimates were severely biased around turning points of the cycle [20]. On a trend the failure is a published gap that later moves, most near a turning point.

What Moves It

The final element is the budget law. The model's annual calendar instant, each October, is the start of the fiscal year of the United States Treasury [21]. The quantum q and the cap Δ_B of Equation (10.7) make each budget a step of tenths of a point of output, at most two points from the last. The marginal tax rate t and the benefit response s_{tr} are the automatic stabilizers, which act without a decision.

TECHNOLOGY Final Element: A Budget Adopted Once a Year

The fiscal year of the United States Treasury begins on October 1 and ends on September 30 [21], so a discretionary budget is set once a year. The tax code acts without a decision. The federal income tax reduces the multiplier effects of demand shocks through the marginal taxation of income fluctuations, and such automatic stabilizers avoid the slow implementation that can make discretionary policy lag far behind events. Unemployment insurance answers the cycle too, but unemployment lags output, and the stabilizing effect of the taxes appeared to be several times that of the benefits [22]. Its failure is lateness, a stimulus that lands in the recovery, as in case 1.

Where the Controller Runs

No computer closes this loop. The budget is adopted for a fiscal year [21], so the decision period is one year, twelve times the control period T_c of one month at which the simulated treasury executes. A

central bank decides more often: the Federal Open Market Committee holds eight regularly scheduled meetings a year [23], between the monthly and the quarterly calendars of the app. In the simulator the PID of Equation (10.8) is the framework's velocity-form block, executed every month and allowed to change the budget only at a calendar instant, and Bohn's rule, the brake and the MPC take its place in the same slot.

> **IN PRACTICE A Ceiling Computed From the Cycle**
>
> The Swiss debt brake turns the cycle into a spending limit with one formula. Maximum expenditure equals revenue times a factor k, the ratio of real trend output to actual output, so a deficit is allowed when output is below trend and a surplus is required above it. Trend output comes from a modified Hodrick-Prescott filter on 24 annual points with a smoothness parameter of 100, and revenue is assumed to move one for one with output. The parameters were kept simple and transparent on purpose, so that the ceiling would not become a black box for policymakers and voters [15].

What Is Modernizing It

Two developments shorten the recognition lag. The first is nowcasting. A dynamic factor model cast in state-space form reads many series as they are released, and a Kalman filter updates its estimate of current growth with each release, weighting each by its timeliness and quality. One central bank's staff nowcast is updated daily when new data arrive and published every week [19]. The treasury's estimate of Equation (10.10) is the same kind of filter fed by one published number a month. The second is the estimate of potential output itself. Estimates made by central bank staff after the mid-1990s had considerably better revision properties than earlier ones, circumstantial evidence that the procedures for estimating potential output improved [24]. The predictive controller of Section 10.5 shows what a model-based plan adds to either estimate, and case 8 shows what its constraints take away.

10.8 Limitations

The economy is a teaching model with backward-looking expectations. Households spend out of current disposable income without anticipating the taxes that a debt implies, firms invest on current growth, and lenders price today's debt. A rule that the public believed, a brake or a debt target, would change how households save, how firms invest and what premium lenders charge. Every case is therefore exposed to the Lucas critique [13], and it measures a rule acting on households and lenders who never learn it, not on a public that anticipates it. The application has no inflation and so no anchored share of expectations, which the central bank of Chapter 11 carries.

Output is set by demand and potential grows at a fixed rate, the central bank's offset is an integral standing in for monetary policy, and the borrowing rate ignores it. The economy is real and closed, with no maturity structure, so the whole debt reprices at once. The premium is a fixed function of the debt, the fold is a static threshold rather than a run on the debt, and the market-access limit of 250 % is a convention. The coefficients, lags and thresholds are teaching choices, so the figures reported here describe no particular country.

The treasury has one discretionary instrument split between spending and transfers, with no tax-rate decisions. The observer has fixed teaching weights and expects every private shock to fade over two years, the MPC knows nothing of shocks to come, and its deficit limit is a priced soft constraint even in the setting labeled strict. The brake is simplified, with the stabilizers allowed in full, the bound enforced at every budget rather than corrected over several years, and credits kept as debt repaid, and the Maastricht rule clips only the deficit a budget plans at today's revenue.

OPEN CHALLENGE **Where Potential Output Is Today**

Every rule that leans on the gap needs today's potential output, the end of a trend that no one can yet see. A study of real-time estimates found that the bulk of their revisions came not from revised data but from the pervasive unreliability of end-of-sample estimates of the trend [20]. Estimates made by central bank staff after the mid-1990s held up considerably better in real time, an improvement that purely statistical detrending methods did not share [24]. Progress would be estimates of potential that hold when the budget is decided, not only after the revisions.

10.9 Conclusion

This chapter studied a simulator that joins Phillips's multiplier-accelerator, drawn as his hydraulic machine, to a treasury that decides once a year through a recognition lag, a quantum and a cap, and to a debt ratio whose borrowing rate rises past a threshold. The stabilizers cut the multiplier from 4 to 1.51 and leave an offset of −3.02 % after a lasting fall of 2 % in demand. Integral policy removes it at the cost of a debt that reaches 111.6 % in twelve years, the lags turn Phillips's 1954 gains into a cycle with 59 of 96 bad quarters, and an annual budget allows a proportional gain of 0.76 where monthly decisions allow 1.53. Above $r = g$ a debt grows from 90 % to 201 % in thirty years with no shock, and from 120 %, past a fold of 100.71 %, no rule in the sweep avoids losing market access. In the crisis challenge the stabilizers alone leave nine bad quarters and a stronger tax with the brake leaves five. Expectations look backward and every coefficient is a teaching value. A fuller model would add a public that learns the rule, a maturity structure for the debt, inflation, and a premium that responds to expectations of default rather than to the debt alone. The application makes the lag, the calendar and the debt of fiscal policy measurable on one machine.

KEY POINTS

- Taxes that fall and benefits that rise in a slump steady the economy at once, but alone they leave a gap of −3.02 % after a lasting fall of 2 % in demand.
- Integral policy removes that offset at the cost of a debt that reaches 111.6 % in twelve years.
- An annual budget allows a proportional gain of 0.76 where monthly decisions allow 1.53, and the lags turn Phillips's 1954 gains into a cycle with 59 of 96 bad quarters.
- Above $r = g$ a debt grows from 90 % to 201 % in thirty years with no shock, and past the fold of 100.71 % no rule in the sweep avoids losing market access.

Where else this applies. Volume 5, Chapter 9 refills a fund through a capped contribution decided a year late, against a deficit that grows on its own.

References

[1] A. W. Phillips, "Stabilisation policy in a closed economy," *The Economic Journal*, vol. 64, no. 254, pp. 290–323, 1954.

[2] ——, "Stabilisation policy and the time-forms of lagged responses," *The Economic Journal*, vol. 67, no. 266, pp. 265–277, 1957.

[3] ——, "Mechanical models in economic dynamics," *Economica*, vol. 17, no. 67, pp. 283–305, 1950.

[4] C. Bissell, "Historical perspectives – The Moniac A hydromechanical analog computer of the 1950s," *IEEE Control Systems*, vol. 27, no. 1, pp. 69–74, 2007.

[5] R. S. Pindyck, "Optimal policies for economic stabilization," *Econometrica*, vol. 41, no. 3, pp. 529–560, 1973.

[6] H. Bohn, "The behavior of U. S. public debt and deficits," *The Quarterly Journal of Economics*, vol. 113, no. 3, pp. 949–963, 1998.

[7] E. M. Leeper, "Equilibria under 'active' and 'passive' monetary and fiscal policies," *Journal of Monetary Economics*, vol. 27, no. 1, pp. 129–147, 1991.

[8] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, III, *Process Dynamics and Control*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.

[9] T. Laubach, "New evidence on the interest rate effects of budget deficits and debt," *Journal of the European Economic Association*, vol. 7, no. 4, pp. 858–885, 2009.

[10] A. R. Ghosh, J. I. Kim, E. G. Mendoza, J. D. Ostry, and M. S. Qureshi, "Fiscal fatigue, fiscal space and debt sustainability in advanced economies," *The Economic Journal*, vol. 123, no. 566, pp. F4–F30, 2013.

[11] P. A. Samuelson, "Interactions between the multiplier analysis and the principle of acceleration," *The Review of Economics and Statistics*, vol. 21, no. 2, 1939.

[12] N. Barr, "The history of the Phillips Machine," in *A. W. H. Phillips: Collected Works in Contemporary Perspective*. Cambridge University Press, 2000, pp. 89–114.

[13] R. E. Lucas, "Econometric policy evaluation: A critique," *Carnegie-Rochester Conference Series on Public Policy*, vol. 1, pp. 19–46, 1976.

[14] A. J. Auerbach and D. Feenberg, "The significance of federal taxes as automatic stabilizers," *Journal of Economic Perspectives*, vol. 14, no. 3, pp. 37–56, 2000.

[15] T. Beljean and A. Geier, "The Swiss debt brake — has it been a success?" *Swiss Journal of Economics and Statistics*, vol. 149, no. 2, pp. 115–135, 2013.

[16] O. Blanchard, "Public debt and low interest rates," *American Economic Review*, vol. 109, no. 4, pp. 1197–1229, 2019.

[17] O. J. Blanchard, "Public debt and low interest rates," National Bureau of Economic Research, Working Paper 25621, 2019. [Online]. Available: https://www.nber.org/papers/w25621

[18] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[19] B. Bok, D. Caratelli, D. Giannone, A. Sbordone, and A. Tambalotti, "Macroeconomic nowcasting and forecasting with big data," Federal Reserve Bank of New York, Staff Report 830, 2017. [Online]. Available: https://www.newyorkfed.org/medialibrary/media/research/staff_reports/sr830.pdf

[20] A. Orphanides and S. van Norden, "The reliability of output gap estimates in real time," Board of Governors of the Federal Reserve System, Finance and Economics Discussion Series 1999-38, 1999. [Online]. Available: https://www.federalreserve.gov/econres/feds/the-reliability-of-output-gap-estimates-in-real-time.htm

[21] "31 U.S.C. 1102, fiscal year," United States Code, 2023 edition, Title 31, Chapter 11, U.S. Government Publishing Office, 2023. [Online]. Available: https://www.govinfo.gov/content/pkg/USCODE-2023-title31/html/USCODE-2023-title31-subtitleII-chap11-sec1102.htm

[22] A. J. Auerbach and D. Feenberg, "The significance of federal taxes as automatic stabilizers," National Bureau of Economic Research, Working Paper 7662, 2000. [Online]. Available: https://www.nber.org/papers/w7662

[23] "Meeting calendars, statements, and minutes (2021–2027)," Board of Governors of the Federal Reserve System, Federal Open Market Committee, 2026, accessed 2026-11-20. [Online]. Available: https://www.federalreserve.gov/monetarypolicy/fomccalendars.htm

[24] R. M. Edge and J. B. Rudd, "Real-time properties of the Federal Reserve's output gap," Board of Governors of the Federal Reserve System, Finance and Economics Discussion Series 2012-86, 2012. [Online]. Available: https://www.federalreserve.gov/pubs/feds/2012/201286/201286pap.pdf

10.10 Exercises

10.1 Concept check · 2 min

In case 2 private demand falls by 2 % of output for good at year 1. The central bank's offset is off, and no budget is decided, so only the taxes and benefits respond. Where does the output gap settle?

(a) At zero, because the benefits keep rising until output is back at potential

(b) Near −3.0 %, the shortfall times the multiplier that the stabilizers leave

(c) Near −8 %, the shortfall times the multiplier $1/(1-c)$ of consumption

(d) It never settles, because the accelerator keeps the economy cycling on its own

10.2 Concept check · 2 min

Table 10.6 shows that, with the same recognition and implementation lags, the largest proportional gain whose cycle decays is 1.53 when the budget is decided every month and 0.76 when it is decided each October. Which explanation is right?

(a) The quantum of 0.1 % rounds away the small corrections that a high gain makes
(b) The cap of two points per decision stops a high gain from acting fast enough
(c) A budget held for a year adds about half a year of delay, which takes phase from the loop
(d) A print error drawn once a quarter is averaged better by twelve decisions than by one

10.3 Concept check · 2 min

In case 4 the weaker policy, $K_c = 0.5$ with $\tau_I = 4\,\text{yr}$, leaves 17 bad quarters on the published data. Adding $\tau_D = 0.5\,\text{yr}$ raises the count to 43. With the print error removed, the same derivative term lowers the count from 11 to 8. Which explanation fits both results?

(a) The derivative is computed only at each October budget, so its rate estimate is a year old
(b) The derivative acts on the error, so every budget decision gives the output a kick
(c) The derivative term replaces part of the integral action that the slow loop needs
(d) Each new quarterly print error is a jump in the measured rate, which the next budget follows

10.4 Theory and calculation · 6 min

The reference economy has $c = 0.75$, a marginal tax rate $t = 0.25$ and a benefit response $s_{tr} = 0.3$. Private demand falls by 2 % of output for good, and the central bank's offset is off.

(a) Find the steady multiplier from spending to output and the gap at which the economy settles with the stabilizers alone. *Checkpoint:* 1.509 and −3.02 %.

(b) Repeat with a marginal tax rate of 0.4, and with no benefit response at the shipped tax rate. *Checkpoint:* 1.290 and −2.58 %, then 2.286 and −4.57 %.

(c) Proportional policy with $K_c = 1.5$ adds a budget of K_c times the negative gap, with no lags, at the shipped stabilizers. Find the steady gap and the budget that holds it. *Checkpoint:* −0.925 % and 1.39 %.

10.5 Theory and calculation · 6 min

Case 5 raises the borrowing rate to $r = 4\,\%$ against growth of $g = 2\,\%$, with no premium and output at potential. The reference budget runs a primary deficit $\delta = 0.9\,\%$ of output, and the debt starts at $b_0 = 90\,\%$.

(a) Solve $\dot{b} = (r - g)\,b + \delta$ for the debt at years 10 and 30, and find the primary surplus that would hold the debt at 90 %. *Checkpoint:* 119.9 %, 201.0 % and 1.8 %.

(b) Bohn's rule sets the budget to $-\rho_B(b - b^*)$ with $b^* = 90\,\%$. Find the steady debt for $\rho_B = 0.05$ and for $\rho_B = 0.2$, and the primary surplus in the second case. *Checkpoint:* 180 %, then 105 % with a surplus of 2.1 %.

(c) State the condition on ρ_B for the rule to hold the debt at all. *Checkpoint:* $\rho_B > r - g = 0.02$.

10.6 Theory and calculation · 6 min

Above the threshold $b_{thr} = 100\,\%$ the borrowing rate carries a premium $\varphi\,(b - b_{thr})$, with $r_0 = 1\,\%$ and $g^* = 2\,\%$. The reference budget holds a primary balance of $s = -0.9\,\%$.

(a) Show that $b = 90\,\%$ is an equilibrium of this budget and that it is stable. *Checkpoint:* $h(0.9) = -0.009$ and $h' = -0.01$ per year.

(b) With the shipped slope $\varphi = 0.03$, find the second equilibrium and the debt's eigenvalue there. *Checkpoint:* 104.67 % and +0.023 per year.

(c) With the case-6 slope $\varphi = 0.15$, find the fold, and say what a debt started at 102 % does under the fixed budget. *Checkpoint:* 100.71 %. It grows.

10.7 App challenge · 8 min

Load scenario 1, The budget by hand. Investment collapses at year 0.25, and the central bank's offset is off. Use the default seed and run each case for 10 yr. For each run read the tiles Quarters outside ±2 % and Worst output gap, and find in the exported file the highest true gap, column `gap_true_pct`, and its time.

(a) Leave the mode at Manual and the Discretionary budget request at 0. *Checkpoint:* 10 quarters, −6.23 %, highest 0.97 % at year 5.42.

(b) Press the reload button. Before Run, switch to Auto, set Controller to Phillips PID on the gap, PID proportional gain to 1 and Integral time to 0. *Checkpoint:* 21 quarters, −5.48 %, highest 4.71 % at year 4.04.

Accept a gap within 0.02 % and a time within 0.05 yr. Say when the stimulus does most of its work.

10.8 App challenge · 8 min

Load scenario 4, Phillips 1957: the same gains with lags. The scenario ships with $K_c = 1.5$, $\tau_I = 2$ yr, a print error of 0.5 % and the annual calendar. Use the default seed and run each case for 24 yr, changing the settings before Run. Read the tiles Quarters outside ±2 % and Worst output gap.

(a) Run the scenario as shipped. *Checkpoint:* 59 quarters and −5.88 %.

(b) Set PID proportional gain to 0.5 and Integral time to 4. *Checkpoint:* 17 quarters and −5.15 %.

(c) Keep (b) and set Derivative time to 0.5. *Checkpoint:* 43 quarters and −7.89 %.

The counts repeat exactly at the default seed. Accept a gap within 0.02 %. Compare the gains of (a) and (b) with Table 10.6.

10.9 App challenge · 8 min

Load the challenge, A financial crisis. Investment collapses at year 2 and a bank rescue adds 10 % of output to the debt. The target is at most six quarters outside ±2 %, at most two years over a 3 % deficit and a debt at or below 100 % at year 12. Use the default seed, make each change before Run and run 12 yr. Read the two count tiles and the debt at year 12, column `debt_pct`.

(a) Run the challenge as shipped, with the stabilizers only. *Checkpoint:* 9 quarters, 1 year, 99.5 %.

(b) Set Controller to Swiss debt brake and Brake stimulus per point of gap to 0.5. *Checkpoint:* 7 quarters, 1 year, 98.3 %.

(c) Keep (b) and set Marginal tax rate to 0.30. *Checkpoint:* 5 quarters, 1 year, 98.9 %.

Accept a debt within 0.1 %. Say which runs meet the target and why the tax rate helps where the brake cannot.

10.10 Application area · 3 min

At the reference equilibrium the overall deficit is 1.8 % of output, below a limit of 3 %. Each point of negative gap adds $t + s_{tr} = 0.55$ points to the primary deficit through the stabilizers.

(a) Find, to first order, the gap at which the stabilizers alone take the deficit to the limit. *Checkpoint:* About −2.2 %.

(b) The gap band of the scorecard is ±2 %. Say what this means for a deficit limit enforced in a slump, and how case 8 shows it.

10.11 Application area · 4 min

A bank rescue adds 10 points of output to the debt, and afterwards the primary balance returns to the level that holds the debt still. Take $r - g = -1$ point, as in the reference economy.

(a) Find how much of the rescue is left in the debt ratio after ten years. *Checkpoint:* About 9.05 points.

(b) Find the constant extra primary surplus that would remove the whole rescue in ten years. *Checkpoint:* About 0.95 % of output a year.

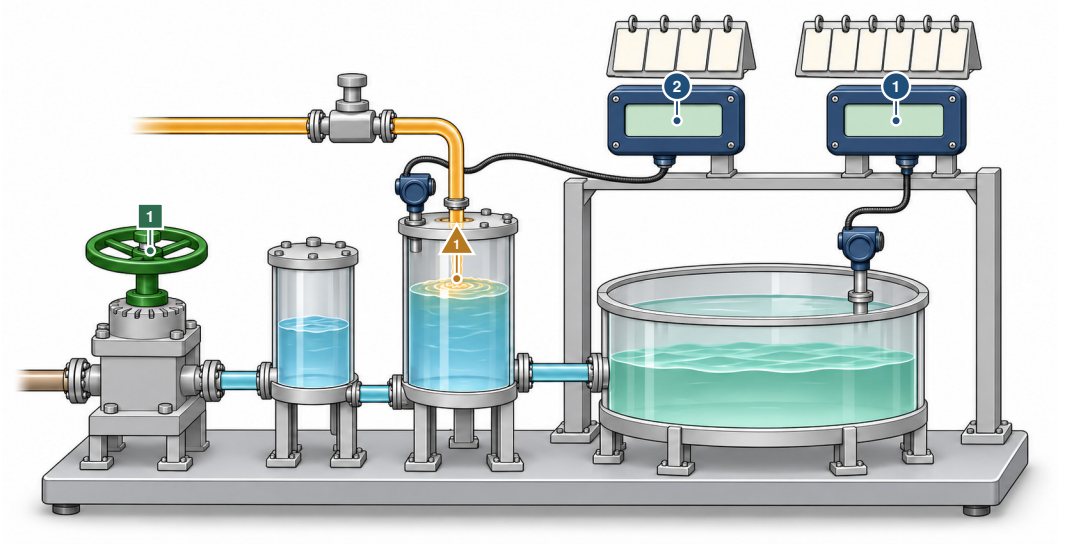
Name one feature of real public debt that the model omits and that would change these numbers.

10.12 Review the agent · 5 min

An AI assistant is asked to retune the Phillips PID of case 4 and replies as follows. “The cycle comes from too much gain. With an integral time of two years the stability limit is 1.28 in continuous time and 1.11 when the budget is decided every month. Keep $\tau_I = 2\,\text{yr}$ and lower K_c from 1.5 to 1.0, which leaves a 10 % margin below the monthly limit, and the cycle will decay.” Find the flaw in the proposal and say how it could be checked.

CHAPTER 11

Inflation Targeting as Sampled, Quantized Control of a Lagged Economy



The controller moves 1 the policy rate, moved at scheduled meetings in quarter-point steps

The controller reads 1 the price index, published weeks after each month

2 the reading of output, published after each quarter and revised

What disturbs it 1 a shock to spending or to prices

Notice the two narrow vessels between the wheel and the wide vessel: a move of the rate passes through both before prices answer, and each reading reaches its display only on its own calendar. Illustration, not to scale.

IN PLAIN TERMS **Steering Prices With One Interest Rate**

The main idea. A central bank tries to keep prices rising slowly and steadily by moving one interest rate. A higher rate slows borrowing and spending, and slower spending later slows the rise in prices.

The challenge. The rate reaches spending after about a year and prices after about two, and the figures that show prices come out weeks late. The bank meets only eight times a year and moves the rate in small fixed steps.

Why it matters. A bank that keeps pushing on the newest figure can turn a small boom into a deep slump.

THREE WAYS TO READ THIS CHAPTER

Explore	11 min	the first page, the history of the first inflation targets (Section 11.2), the fact on the target range (Section 11.5), the try-it segments of cases 2 and 4 and the fact on the floor (Section 11.6), and the key points
Study	45 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	15 min	the first page, the results (Section 11.6), Technology in Practice (Section 11.7), Limitations (Section 11.8) and the open challenge

The control question A central bank moves one rate eight times a year in quarter-point steps, on statistics published late, and the economy answers in output a year later and in prices two years later. Which way of deciding holds inflation near its target through those lags, and what do the calendar, the quantum and the lower bound cost it?

Learning objectives

- Derive the Taylor-principle boundary on the inflation response from the model's steady state, with and without an anchored public.
- Determine the action of the inflation PID from the sign of the process gain in the convention of Chapter 1.
- Explain why the transmission is three lags in series and not a dead time, and estimate the effective dead time a step test returns.
- Predict what the decision calendar, the rate quantum and the floor do to a demand boom and a financial crisis.
- Compare a proportional rule, the PID and forecast targeting on an unannounced fall of the neutral rate and on an oil shock.

Prerequisites Sections 1.4.2 (proportional offset), 1.4.3 (controller action), 1.4.4, 1.6.2, 1.4.9 and 1.4.11 (sample time and a quantized output). Chapter 9 and Chapter 10 decide on the same kind of calendar. Section 1.6.3 is needed only for forecast targeting.

The app https://apmonitor.com/apps/inflation-targeting/

See also Volume 5, Chapter 3 and Volume 5, Chapter 4 for control through a late measurement, and Volume 5, Chapter 9 for a slow stock refilled by a sampled, capped decision. Volume 9, Chapter 4 for an estimator that carries an unmeasured bias.

11.1 Introduction

A policy rate is changed at a meeting, and the economy it is meant to steer answers over the following two years. In the simulator of this chapter, a rise of 100 bp held for one quarter moves the output gap most 1.04 yr after the rise and inflation most after 2.40 yr, and the bank sees both only through a consumer price index (CPI) published two weeks after each month and a gross domestic product (GDP) estimate published a month after each quarter and revised for a year. A chair who raises the rate 25 bp at every meeting for as long as the inflation print exceeds 2.25 % takes it to 8.875 %, a year and a half after inflation peaked, and drives the output gap to −4.93 % and inflation to −1.82 %. The same demand boom under the Taylor rule peaks the rate at 5.625 % and the gap stays between +1.61 % and −0.90 %.

Economists have described this problem in the language of feedback control for seventy years. Stabilization policy was written in 1954 as proportional, integral and derivative correction of a production gap acting through lags [1], and in 1972 short-term stabilization was posed as a discrete-time linear quadratic tracking problem on a 28-state quarterly econometric model [2]. The rule proposed in 1993 sets the rate from the inflation gap and the output gap with fixed coefficients [3], which is a proportional controller on two measured errors. Inflation-forecast targeting chooses the rate so that the bank's own forecast of inflation meets the target over a policy horizon [4], which is receding-horizon optimal control. A control paper is the right frame because the domain already uses it, with one standing objection. Under rational expectations a discretionary optimum does not maximize the social objective, and there is then no way control theory can be applied to economic planning [5]. The simulator takes the other branch, with backward-looking expectations that can be integrated forward in time, and states that choice as a limitation.

The structure is general. A slow plant reached through a chain of lags, a measurement that is a late and revised average of the past, an actuator that moves only at scheduled instants and in fixed steps, a hard lower limit, and an operating point that drifts without being announced appear together in process plants whose quality is known only when a laboratory result returns, whose analyzers

report on a cycle, and whose operators change setpoints at shift meetings. The column behind a slow analyzer of Volume 5, Chapter 3 and the digester graded at its blowline in Volume 5, Chapter 4 are two of them. Each of those features has a textbook answer on its own, such as dead-time compensation, sampled-data design, anti-windup and integral action [6]. What a transfer function from the policy rate to inflation does not show is which vintage of which statistic the controller held when it decided. The belief this plant overturns is that a controller responding harder or more often to the newest measurement must control better. In this economy a stronger inflation response raises the integrated loss of a demand boom from 4.24 to 4.85, and deciding every week instead of eight times a year lowers it only from 3.64 to 3.46.

However, the policy-rule literature evaluates rules on estimated models, where the information set and the calendar are assumptions of the analysis [7, 8], so a student rarely has a plant on which the chair, the rule, a proportional-integral-derivative (PID) controller and an optimizer act on the same published data through the same quantized calendar. This chapter works with such a plant and measures on it what the lags, the data and the calendar do to each policy. The structure, the Taylor rule, forecast targeting and the real-time data problem are all taken from the literature cited below. The simulator joins a continuous-time economy to a statistics agency that keeps every vintage and to the framework's shared decision calendar, with each equation, constant, delay and limit written out. Nine cases and a challenge each isolate one mechanism, and each was measured on the running engine.

The chapter is organized as follows. Section 11.2 traces the problem's history, Section 11.3 writes out the economy and Section 11.4 its dynamic character. Section 11.5 gives the bank's four ways of deciding, and Section 11.6 reports the measured cases and the challenge. Section 11.7 sets the real statistics, rate and committee beside the model, Section 11.8 states the limitations, Section 11.9 summarizes the results, and the exercises follow in Section 11.10.

11.2 Stabilization Policy as Feedback, 1954 to Forecast Targeting

The control description of macroeconomic policy is older than the policy rules in use today. In 1954 a model of a closed economy was given a government that corrects the deviation of production from its desired level in proportion to the deviation, to its time integral and to its rate of change, with the correction reaching demand through lags [1]. Chapter 10 rebuilds that economy and its PID policy as a plant of its own. Four years later the same author fitted the relation between unemployment and the rate of change of money wage rates in the United Kingdom from 1861 to 1957 [9], the curve that later carried his name and that the Phillips-curve view of this application draws. How long monetary policy takes to act was by 1961 a question argued in its own right [10].

The 1970s brought optimal control to the problem and then the objection to it. A quadratic tracking formulation was solved analytically and applied to a 28-state quarterly econometric model, and the experiments were presented as a tool both for policy planning and for analyzing the dynamics of econometric models [2]. Within five years two arguments limited what such a design could claim. A fitted model's coefficients need not survive a change in the policy that produced the data [11], and discretionary policy, the best decision given the current situation, does not maximize an agreed social objective when the economy's agents are rational [5]. Both arguments are about the plant changing when the controller does, and both return in the limitations of this chapter.

Inflation targeting turned the objective into a published number, and the history box after this paragraph follows the first regimes and what they shared [7]. Forecast targeting made the bank's forecast the intermediate target [4]. A rate rule with fixed responses to inflation and the output gap was proposed in 1993 as a description of practice [3], and the efficient rules of a simple backward-looking model, those minimizing a weighted sum of output variance and inflation variance, turned out to be Taylor rules and to be equivalent to inflation targets pursued at different speeds [8]. The policy-rule literature of the same decade set the rules inside a common framework of optimal policy, credibility and imperfect information [12].

Table 11.1 Signals of the Economy and the Bank by Role, With Their Defaults

Symbol	Role	Default	Range
π_{pub}	CV, latest monthly CPI print	SP π^* 2 %	0–6 %
π, y	objectives, true inflation and gap, hidden from the bank	2 %, 0	—
y_{pub}	measured, latest quarterly gap print	0	vintages 1–3
i	MV, applied policy rate	2.875 %	0.125–20 %
i^w	request, from the chair's slider or a controller	2.875 %	0.125–19.875
d_D, d_S	DVs, demand and cost-push shocks	0, 0	events, ±6, ±3
r^*, g^*	DVs, true neutral rate and potential growth	0.875 %, 2	−2–4, 0–5
r_1, r_2	states, transmission stages	0.875 %	—
π^a, λ	states, adaptive expectations, anchored share	2 %, 0.7	0–1 for λ
m, P	states, potential misperception, log potential	0, 0	—
I_y, I_{yp}, I_π	states, integrals for period averages	0	—
u, g	outputs, unemployment and real GDP growth	4 %, 2	—
$\hat{x}$	bank's estimate of seven model states	truth at start	—

Shocks: d_D in percent of potential, d_S and g^* in percent a year.

HISTORY Inflation Targets as Published Numbers, 1998 to 2012

In early 1998 explicit inflation targeting was pursued in New Zealand, Canada, the United Kingdom, Sweden and Australia, and arguably also in Finland and Spain. The regimes shared three marks: a publicly announced numerical target, given as a range, a point or a point with a tolerance interval, a framework for decisions that compared an inflation forecast with the target, so that the forecast served as an intermediate target, and a higher than average degree of transparency and accountability [7]. In the United States the Federal Open Market Committee adopted, effective January 24, 2012, a statement judging that inflation of 2 percent, measured by the annual change in the price index for personal consumption expenditures, is most consistent over the longer run with its statutory mandate [13].

Two problems kept the rule from being a formula. A rule evaluated with the information available in real time prescribes rates that differ considerably from the same rule evaluated with revised data [14], and ex post revisions of the estimated output gap are of the same order of magnitude as the gap itself and highly persistent, mostly because the trend of output is unreliable at the end of the sample [15]. The second problem was the floor. By 2003 the overnight rate in Japan had been within 50 basis points of zero since October 1995, so that the usual remedy of a lower short-term rate was unavailable [16]. A dynamic model with moderate nominal rigidities was by then able to account for the observed inertia of inflation and persistence of output after a monetary shock [17], and the lags of the present model give the same qualitative picture, with output moving well before inflation.

11.3 An Economy of Stocks and Lags

The simulated economy is a continuous-time teaching analogue of the quarterly backward-looking model of [7], in years, drawn in Figure 11.1 as the stocks and flows it is. The policy rate i is the one in force, held between decisions by the bank of Section 11.5. Inflation π, the output gap y and unemployment u are in percent, rates of change in percent a year, and time in years. Table 11.1 classifies the signals.

11.3.1 Transmission and Prices

The nominal rate less expected inflation is the real rate, and it reaches spending through two first-order stages, as shown in Equation (11.1). These are lag states, not market rates.

$$\tau_1 \dot{r}_1 = i - \pi^e - r_1, \qquad \tau_2 \dot{r}_2 = r_1 - r_2 \tag{11.1}$$

where $\tau_1 = 0.25\,\text{yr}$ stands for credit and asset prices and $\tau_2 = 0.5\,\text{yr}$ for spending decisions. The output gap answers the second stage against the true neutral real rate, and the gap moves inflation, as

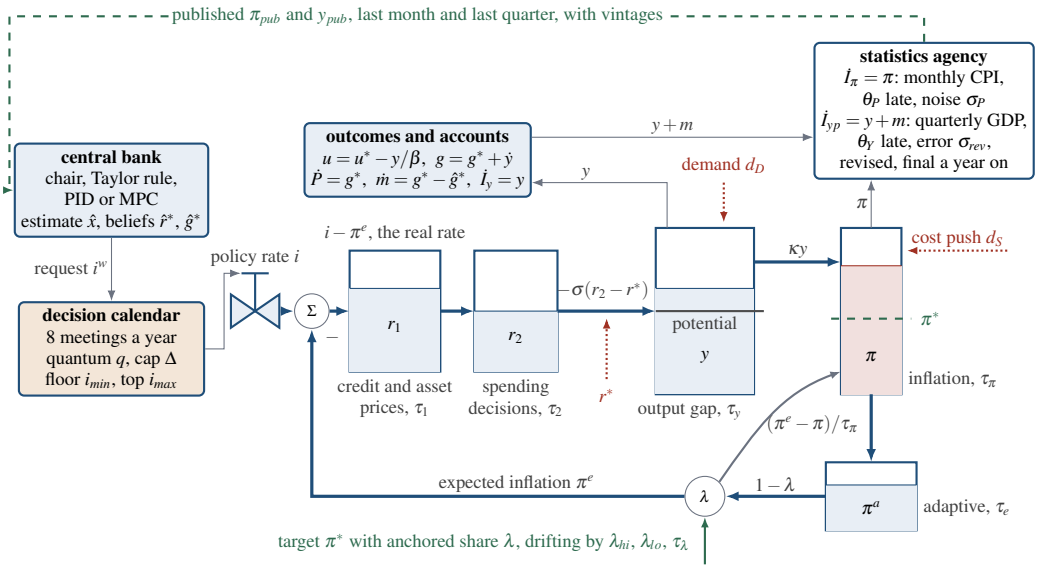


Figure 11.1 The economy as stocks and lags, the statistics agency, and the bank with its calendar. The real rate $i - \pi^e$ fills the credit stage r_1, which fills the spending stage r_2, and the second stage drains the output gap y against the true neutral rate r^*. The gap raises inflation π at the rate κy while inflation relaxes toward expected inflation π^e, a mix with share λ of the target π^* and the adaptive stock π^a. Densely dotted red arrows are disturbances, thick blue arrows are flows between stocks and thin gray arrows are signals. The agency integrates π and the gap the bank can measure, $y + m$, and returns monthly and quarterly averages late along the dashed green path, where the bank's request i^w passes the calendar's quantum q, cap Δ and bounds before it reaches the valve.

shown in Equation (11.2).

$$\begin{aligned} \tau_y \dot{y} &= -y - \sigma(r_2 - r^*) + d_D \\ \dot{\pi} &= \kappa y + (\pi^e - \pi)/\tau_\pi + d_S \end{aligned} \tag{11.2}$$

where $\tau_y = 0.75\,\text{yr}$, $\sigma = 1$ point of gap per point of real rate, $r^* = 0.875\,\%$, $\kappa = 0.5$ percent a year per point of gap and $\tau_\pi = 2\,\text{yr}$. The demand disturbance d_D is in percent of potential and the cost push d_S in percent a year. At a steady gap inflation settles at $\pi = \pi^e + \kappa\tau_\pi y$, a short-run Phillips curve whose slope against unemployment is $-\kappa\tau_\pi\beta = -2$ with the Okun coefficient below.

11.3.2 Expectations and Credibility

Expected inflation is a share λ anchored on the announced target and an adaptive remainder, as shown in Equation (11.3).

$$\begin{gathered} \pi^e = \lambda\pi^* + (1-\lambda)\pi^a, \qquad \tau_e \dot{\pi}^a = \pi - \pi^a \\ \tau_\lambda \dot{\lambda} = \begin{cases} \lambda_{hi} - \lambda & |\pi - \pi^*| \le 1 \\ 0 & 1 < |\pi - \pi^*| < 3 \\ \lambda_{lo} - \lambda & |\pi - \pi^*| \ge 3 \end{cases} \end{gathered} \tag{11.3}$$

where $\tau_e = 1.5\,\text{yr}$, and the drift of the second line acts only when the learner switches it on, with $\lambda_{hi} = 0.9$, $\lambda_{lo} = 0.3$ and $\tau_\lambda = 6\,\text{yr}$. The share is clipped to $[0,1]$ inside the derivative and after every step. With $\lambda = 0$ a gap held at one point raises inflation at the constant rate $\kappa\tau_\pi/(\tau_\pi + \tau_e) = 0.286$ points a year, the accelerationist curve of the 1970s, which the application's check confirms to 5×10^{-9} after fifteen years. With $\lambda = 1$ inflation returns to target after an oil shock with the rate held at neutral, which the check confirms within 0.01 points after fifteen years.

11.3.3 Outcomes and Accounts

Unemployment, growth and the accounts that the statistics need are given by Equation (11.4).

$$\begin{aligned} u &= u^* - y/\beta, \qquad g = g^* + \dot{y} \\ \dot{P} &= g^*, \qquad \dot{m} = g^* - \hat{g}^* \\ \dot{I}_y &= y, \qquad \dot{I}_{yp} = y + m, \qquad \dot{I}_\pi = \pi \end{aligned} \tag{11.4}$$

where $u^* = 4\%$, $\beta = 2$ points of gap per point of unemployment, $g^* = 2$ percent a year, P is the log of potential output in percent with the GDP index $100\exp[(P+y)/100]$, and m is how far the bank's potential has drifted from the truth when its belief $\hat{g}^*$ differs from g^*, so that the gap it can measure is $y+m$, the Orphanides problem [15]. Both identities hold in the exported file, Okun's law to 5×10^{-9} on every row and the growth identity to 0.0015 points against a central difference of the GDP index, away from the instants where a shock starts or ends. The integrals give exact period averages, interpolated to the period boundary inside the step. Quarterly growth, annualized, is four times the sum of the quarter's change in log potential and in average gap, two negative quarters in a row raise RECESSION, and calendar-year growth is the change of log potential over the year plus the change of the year's average gap.

11.3.4 What the Agency Publishes

The bank never sees the economy. Each month's average inflation, from I_π, is published as the CPI print of Equation (11.5) two weeks after the month ends and is never revised.

$$\pi_{\text{pub},j} = 12\int_{(j-1)/12}^{j/12} \pi\,\mathrm{d}t + \sigma_P\,\nu_j \tag{11.5}$$

where $\sigma_P = 0.2$ points and ν_j is a standard normal deviate. Each quarter's average of $y+m$, from I_{yp}, receives a first print one month after the quarter, modeled on the United States advance estimate, with an error $\varepsilon \sim N(0, 0.5^2)$, a second print two months later that keeps 0.4ε, and a final print a year after the first with no error. The published gap is the newest quarter's latest vintage. A misleading first print can be scheduled, two points too low and revised later. Every deviate comes from the engine's seeded generator in a fixed order, one per CPI print and one per first GDP print, so a reset replays every release. In 15 years the agency publishes 179 CPI prints and 173 GDP prints. At $t = 0$ the bank holds a print of last month and last quarter at the starting point, as though the economy had been there before.

11.3.5 Shocks, Start and Integration

Three event shocks have declared shapes in the time t' since they started. The oil shock adds $d_S = 3$ percent a year for a year and then decays with a 0.5 yr time constant, and the challenge fires it at 6. The demand boom adds $d_D = 2$ for two years and decays with the same time constant. The financial crisis starts at $d_D = -6$ and recovers in a straight line over three years. Two sliders add a persistent d_D of up to ±6 and d_S of up to ±3, a de-anchoring event sets $\lambda = 0.4$, and the true r^* and g^* are sliders hidden from the bank. A case starts in equilibrium at its target unless it declares otherwise, with $r_1 = r_2 = r^* - y_0/\sigma + \bar{d}_D/\sigma$ for a starting gap y_0 and a persistent demand $\bar{d}_D$, $\pi = \pi^a = \pi^*$, and the first rate on the grid at $r_2 + \pi^e = 2.875\%$. Classical fourth-order Runge-Kutta integrates the eleven states on a fixed step of a week, $\Delta t = 1/52$ year, the controller executes every week before the step, and the historian logs every week. The equilibrium holds with zero derivatives and no drift at all over ten noise-free years, and halving the step leaves the challenge's verdicts unchanged and its peak unemployment within 10^{-4} points. The default seed is 20261004, and the engine of Section 1.6.1 runs the economy at up to two simulated years a second. Table 11.2 gathers the constants.

Table 11.2 Model Constants, Marked M When the Show Model Dialog Sets Them, A When an Advanced Control Does and C When Only the Source Does

Symbol	Meaning	Value	Range	Class	Symbol	Meaning	Value	Range	Class
τ_1	credit stage lag	0.25 yr	0.05–1	M	τ_2	spending stage lag	0.5 yr	0.05–1.5	M
τ_y	output-gap lag	0.75 yr	0.1–2	M	σ	interest sensitivity of spending	1.0	0.2–2	M
κ	Phillips slope, %/yr per point of gap	0.5	0.1–1.5	M	τ_π	pull of inflation toward π^e	2 yr	0.5–5	M
τ_e	adaptive expectations lag	1.5 yr	0.25–5	M	λ_{hi}, λ_{lo}	anchored share near and far from target	0.9, 0.3	0.5–1, 0–0.6	M
τ_λ	credibility drift lag	6 yr	1–20	M	u^*, β	natural unemployment, Okun coefficient	4 %, 2	2–8, 1–3.5	M
θ_Y	first GDP print after the quarter	1/12 yr	0–26 wk	M	σ_{rev}	first-print error	0.5	0–2	M
θ_P	CPI print after the month	2 wk	0–13 wk	M	σ_P	CPI noise	0.2	0–1.5	M
r^*	true neutral real rate	0.875 %	−2–4	A	g^*	true potential growth	2 %/yr	0–5	A
—	second print, share of error kept	1/6 yr, 0.4	—	C	—	final print after the first	1 yr	—	C
—	oil shock, hold, decay	3 %/yr, 1, 0.5 yr	6 in challenge	C	—	boom, hold, decay	2, 2, 0.5 yr	—	C
—	crisis and linear recovery	−6, 3 yr	—	C	—	drift bands on $\lvert\pi - \pi^*\rvert$	1, 3 points	—	C
—	STAGFLATION	$\pi > \pi^* + 2$, $y < -1$	—	C	—	OVERHEATING, DE-ANCHORED	$y > 2$, $\lambda < 0.5$	—	C
Δt	integration step, control period	1/52 yr	—	C	—	log rate, seed	52 per yr, 20261004	—	C

11.3.6 Assumptions

Expectations look backward. The anchored share and the adaptive stock of Equation (11.3) make the model causal, so an engine can step it forward. A public that learns the bank's rule would change these relationships, which is the critique of [11], and the model therefore shows what a rule does to this economy, not what it would do to one whose public anticipates it.

Teaching values. Every coefficient, lag, shock size and threshold is a declared teaching value. The lags were chosen so that a pulse lands its troughs in design windows (Section 11.4), and estimated coefficients come from models such as [7] and [17], not from this one.

A local, linear economy. The investment-saving (IS) curve of Equation (11.2), the Phillips curve and Okun's law are linear near the operating point, so a gap near −7 %, which a chair moving 50 bp a meeting reaches in case 1 (−6.93 %), is outside the range they describe.

One instrument, one closed economy. There is no exchange rate, no financial sector beyond the two stages, no fiscal policy, no forward guidance and no asset purchases.

Constructed statistics. The CPI and GDP prints are period averages of the model's own states with a simplified revision schedule, not reconstructions of any agency's procedure. The anchored share is read by the bank as an exact survey.

Deterministic shocks. Only the releases are random. The crisis is a demand shock, not a credit crunch with dynamics of its own.

11.4 How the Economy Answers a Move

11.4.1 Three Lags in Series, Not a Dead Time

Nothing in the economy forgets a move, and nothing waits. A point held from $t = 0$ on the noise-free reference has moved the first stage 0.63 points after a quarter but the second only 0.15, and the gap is −0.09 % after half a year, −0.35 % after a year and −0.76 % after two, while inflation has moved only −0.27 points by then. The response starts at once and arrives late, the signature of three lags in series rather than a transport delay. A first-order-plus-dead-time fit of such a response returns an effective dead time made of lags (Section 1.4.4). Released after a quarter, the same rise moves the gap most at 1.04 yr by −0.13 % and inflation most at 2.40 yr by −0.07 points, quarters 4.2 and 9.6. Held a year, the troughs come at quarters 6.1 and 11.3, by −0.50 % and −0.26 points, as shown in Figure 11.2(a). Each rise is set before Run, which puts it in force from $t = 0$, so every time is measured from the rise.

🕮 **Pause.** Try Exercise 11.2 before reading on.

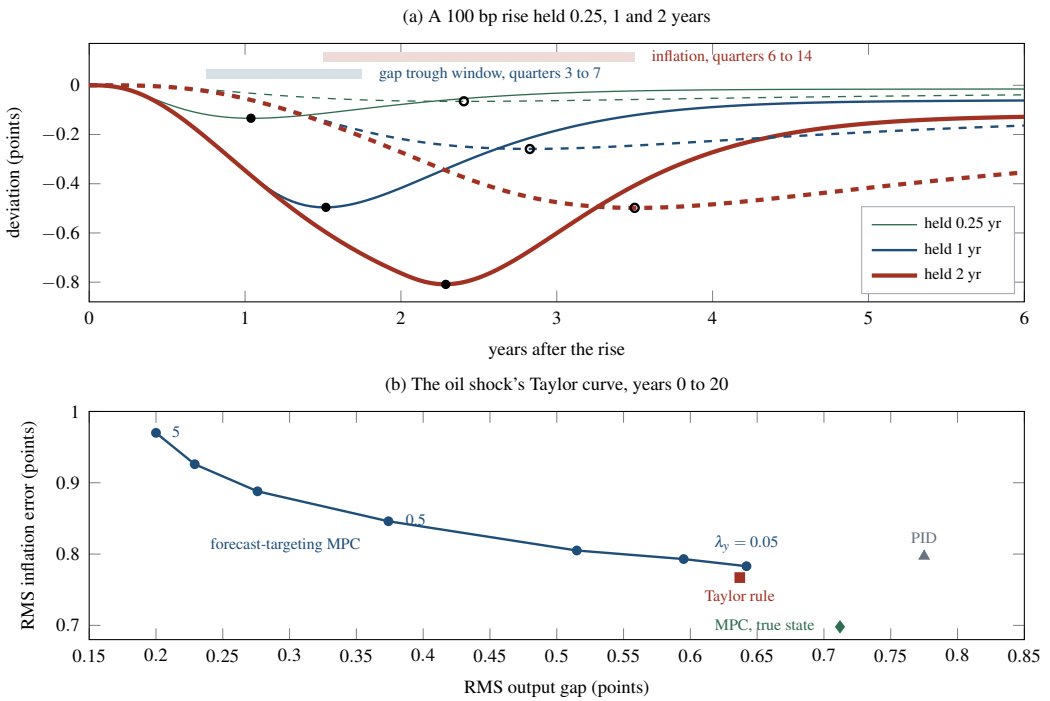


Figure 11.2 Two measurements from the application's engine. (a) The output gap (solid) and inflation (dashed) after a 100 bp rise held 0.25 years (thin lines), 1 year (medium) and 2 years (thick) on the noise-free reference with weekly decisions and no quantum, in force from $t = 0$, with each gap trough marked by a filled dot, each inflation trough by an open circle and the design windows shaded above. (b) The oil shock of case 5 over 20 years, root-mean-square (RMS) inflation error against RMS output gap for the model predictive controller (MPC) at output weights from 0.05 to 5 (line with dots), the Taylor rule (square), the PID (triangle) and the MPC given the true state (diamond).

The application's plan placed the troughs of a rise held two years inside quarters 3 to 7 for the gap and 6 to 14 for inflation. The model cannot do that, and the check was restated rather than the model bent. The gap keeps falling for as long as the real rate is held up, so a rise held two years has its gap trough after the release, at quarter 9.2, with inflation's at quarter 14.0. The windows are therefore asserted on rises held a quarter and a year, which land inside both, and the two-year rise is checked for a gap trough between quarters 8 and 10. The windows themselves are design targets of the application and not estimates.

11.4.2 The Taylor Principle With an Anchor

The Taylor rule of Section 11.5 is stable only if the real rate rises when inflation does. Setting the derivatives of Equations (11.1) to (11.3) to zero with the rule acting on true values and $\hat{r}^* = r^*$ gives $\pi^a = \pi$, $y = \lambda x/(\kappa\tau_\pi)$ and $y = -\sigma[(a_\pi - 1 + \lambda)x + a_y y]$ for an inflation deviation $x = \pi - \pi^*$, so a deviation can persist without a shock only when the bracket of Equation (11.6) vanishes.

$$\left[a_\pi - 1 + \lambda\left(1 + \frac{1 + \sigma a_y}{\kappa\sigma\tau_\pi}\right)\right]x = 0 \tag{11.6}$$

where a_π is the rule's total inflation response and a_y its gap response. The bracket is proportional to the determinant of the closed loop's state matrix, so a real root crosses zero where it vanishes. For the linearized loop without publication delays or quantum the slowest root is $+0.0148$ per year at $a_\pi = 0.9$ with no anchor and -0.0148 at 1.1, so a deviation feeds itself below the boundary,

Table 11.3 Bank Settings, Defaults and Ranges. A Marks an Advanced Control

Symbol	Setting	Default	Range
Policy and calendar			
π^*	inflation target	2 %	0–6
—	controller in AUTO	Taylor rule	3 policies
—	calendar	8 meetings a year	4 calendars
Δ, q	largest move, quantum (A)	0.5, 0.25	0.25–2, 0–1
i_{min}, i_{max}	lower bound (A), top	0.125 %, 20 %	−2–2, fixed
$\hat{r}^*$, $\hat{g}^*$	bank's neutral rate and growth (A)	0.875 %, 2	−2–4, 0–5
Taylor rule			
a_π, a_y	inflation and gap responses	1.5, 0.5	0–3, 0–2
ρ	partial adjustment (A)	0	0–0.9
PID			
K_c, τ_I	gain, integral time	1.5, 8 yr	0.1–5, 0–20
τ_D	derivative time (A)	0	0–2 yr
K_y	gain on the published gap	0.5	0–2
—	floor tracking (A)	on	on, off
Forecast-targeting MPC			
λ_y, ν	weight on the gap, move penalty	0.5, 0.3	0.05–5, 0.05–3
H	horizon (A)	3 yr	1–6
—	information (A)	estimator	estimator, truth
Public expectations			
λ_0	anchored share at the start	0.7	0–1
—	credibility drift	off	off, on

slowly. The boundary is $a_\pi = 1$ with no anchor, the classic Taylor principle, and 0.25 with 30 percent anchored at $a_y = 0.5$, because an anchored public pulls inflation back by itself. Between the two, Equation (11.6) puts it at 0.75 for an anchored share of 0.1 and 0.5 for 0.2. A forty-year sweep on case 3's boom confirms both. With no anchor, $a_\pi = 0.9$ leaves inflation 1.00, 1.06 and 1.43 points over target at years 10, 20 and 40, while 1.1 leaves 0.79, 0.59 and 0.47. With 30 percent anchored, 0.2 diverges, from 1.02 to 1.30 points between years 10 and 40, while 0.3 converges, from 0.90 to 0.69. The plan expected 0.8 to diverge at an anchored share of 0.3. It converges there, to 0.04 points at year 40, so the case runs with no anchor.

11.4.3 A Grid That Holds the Neutral Rate

The rate lives on the grid of target-range midpoints, $0.125 + 0.25n$ percent. The neutral real rate is 0.875 % rather than a round 1 % so that the neutral nominal rate, 2.875 %, lies on that grid and the equilibrium holds exactly. With a 50 bp quantum the nearest rates are 2.625 % and 3.125 %, neither of which holds the economy still, and the Taylor rule settles into a relay cycle whose size the quantum sets, not the gain. From year 20 to 30 of case 6 the rate alternates between them, rising every 3.6 years, with the gap within ±0.15 % and inflation between 1.96 % and 2.04 %. An inflation response of 3 instead of 1.5 leaves the gap within ±0.16 %. Chapter 1 treats the period and the step of a sampled output in Section 1.4.11. The starting rate is rounded to the quantum in force at start-up, so a one-point quantum starts the rate at 3.125 % and leaves a gap cycle from –0.24 % to +0.27 %.

🕮 **Pause.** Try Exercise 11.3 before reading on.

11.5 How the Bank Decides

The controller executes every week. Each execution reads the releases that came out since the last one and corrects the bank's estimate, asks the selected policy for a request i^w, and asks the calendar whether this execution is a decision. The chair and the three automatic policies share that last step, so they differ only in how they form the request. Table 11.3 lists the settings.

11.5.1 The Calendar and the Shape of a Decision

The calendar and the shaping of each decision come from the framework's shared decision-calendar module, `core/cadence.js`, which also moves the contribution of Volume 5, Chapter 9 on a calendar with a quantum and a cap. Eight meetings fall at the fractions 0.075, 0.205, 0.340, 0.455, 0.575, 0.715, 0.840 and 0.955 of each year, spaced as the Federal Open Market Committee's schedules usually are but a teaching calendar, not any year's dates. Monthly and quarterly calendars are regular intervals, and the continuous one makes every weekly execution a decision. An execution is a decision when a calendar instant falls at or before it and after the previous decision, so a meeting between two weekly executions is taken at the first one after it. The eight meetings therefore fall 6 to 8 weeks apart, monthly decisions 4 or 5 weeks apart and quarterly ones every 13 weeks. The module also keeps a log of the decisions, which the meeting strip of the Economy view draws as one block for each quantum step, above its line for a rise and below it for a cut. At a decision the request is shaped as shown in Equation (11.7).

$$\begin{aligned} \tilde{\imath} &= b + q\,\mathrm{round}\big((i^{w} - b)/q\big), \qquad \Delta_q = q\max\big(1, \lfloor \Delta/q \rfloor\big) \\ i_k &= \mathrm{clip}_{[i_{min},\, i_{max}]}\Big(i_{k-1} + \mathrm{sat}_{\Delta_q}\big(\tilde{\imath} - i_{k-1}\big)\Big) \end{aligned} \tag{11.7}$$

where $b = 0.125$ is the grid's base, rounding takes ties away from zero, the bounds are first pulled in to the grid, sat limits a move to $\pm\Delta_q$, and a request that is not finite holds the rate. The default $q = 0.25$, $\Delta = 0.5$, $i_{min} = 0.125\,\%$ and $i_{max} = 20\,\%$ give the familiar 25 bp steps, at most 50 bp a meeting. The rate is then held to the next decision. The application's check finds every decision of four calendars on the grid, inside the cap and the bounds and on its calendar instant, and the floor uncrossed in every case under every policy.

> ✦ DID YOU KNOW? **A Target Range of 0 to 1/4 Percent, 2008**
>
> On December 16, 2008 the Federal Open Market Committee set a target range for the federal funds rate of 0 to 1/4 percent [18]. The model's floor of 0.125 % is the midpoint of that range, and its grid of midpoints keeps ranges a quarter point wide.

11.5.2 The Chair

In MANUAL the request is the chair's slider, from 0.125 % to 19.875 % in 25 bp steps, and the slider is a request, which the next meeting shapes exactly as it shapes a controller's. In AUTO the slider follows the controller's request, not the rate after the quantum, the cap and the floor, so a switch to MANUAL leaves the chair asking for what the controller asked for. A switch to AUTO, or to another controller, holds the rate in force at the first decision after it, and every policy transfers without a jump (Section 1.4.10).

11.5.3 The Taylor Rule

The rule is a proportional controller on the latest prints, as shown in Equation (11.8).

$$\begin{aligned} i_T &= \hat{r}^{*} + \pi^{*} + a_{\pi}(\pi_{\mathrm{pub}} - \pi^{*}) + a_{y} y_{\mathrm{pub}} \\ i^{w} &= \rho\, i + (1 - \rho)\, i_T \end{aligned} \tag{11.8}$$

where $a_{\pi} = 1.5$ is the total response to a point of inflation, Taylor's coefficient of one plus one half [3], $a_y = 0.5$, and the partial adjustment ρ is zero by default. The intercept is the bank's belief $\hat{r}^{*} = 0.875\,\%$, which an unannounced move of r^{*} makes wrong.

11.5.4 The PID

The PID is the framework's velocity-form PID of Section 1.6.2. It runs every week on the CPI print, and the published gap is added outside it, as shown in Equation (11.9).

$$\begin{aligned} e_k &= \pi_{\text{pub},k} - \pi^*, \qquad P_k = K_c e_k \\ D_k &= K_c \tau_D (\pi_{\text{pub},k} - \pi_{\text{pub},k-1})/\Delta t \\ S_k &= S_{k-1} + (P_k - P_{k-1}) + (D_k - D_{k-1}) \\ \tilde{u}_k &= \hat{r}^* + \pi^* + S_k + I_{k-1}, \quad u_k = \text{clip}_{[\underline{u},\bar{u}]}(\tilde{u}_k) \\ I_k &= I_{k-1} + \left[K_c e_k/\tau_I + (u_k - \tilde{u}_k)/\tau_I\right]\Delta t \\ i^w &= u_k + K_y y_{\text{pub}} \end{aligned} \tag{11.9}$$

where $K_c = 1.5$, $\tau_I = 8\,\text{yr}$, $\tau_D = 0$ and $K_y = 0.5$. These are Equations (1.23) to (1.26) with the bias $u_b = \hat{r}^* + \pi^*$, $\tilde{u}_k = v_k$ and a tracking time equal to τ_I, so the running sum is never clipped and the back-calculation acts on the integral only.

The error $e_k = \pi_{\text{pub},k} - \pi^*$ is the framework's $a(\text{SP} - \text{PV})$ with $a = -1$. A higher rate lowers inflation, so the process gain is negative, and the controller must raise its output as the PV rises. Inflation above target raises the rate. In the convention of Section 1.4.3 this is a direct-acting controller, and the loop gain aK_cK_p is positive. The app's guide to the PID also calls it direct acting. Named by its push against inflation, the same loop would be called reverse acting, so the error equation, not the name, fixes the sign.

Pause. Try Exercise 11.1 before reading on.

With floor tracking on, $\underline{u} = i_{min} - K_y y_{\text{pub}}$ and $\bar{u} = i_{max} - K_y y_{\text{pub}}$, so that the request is held inside the rate's own bounds and the integral stops at the floor. With it off, $\underline{u} = -70\,\%$, which removes the lower clip while the upper one still back-calculates. The derivative is taken on the measurement with no filter, so a nonzero τ_D differentiates the CPI's noise, and $\tau_I = 0$ removes the integral. The meeting's quantum and cap are not fed back to the integral, which keeps integrating in the weeks between meetings while the rate is held. Whenever the PID is not the policy in force, in MANUAL or with another controller selected, it tracks the rate in force less $K_y y_{\text{pub}}$. With its integral off and $K_c = 1.5$, $K_y = 0.5$, the PID is the Taylor rule, and the application's check finds the two runs of case 4 identical to 10^{-9} at every step.

11.5.5 The Bank's Estimate

The model predictive controller (MPC) reads a state estimate $\hat{x} = [r_1, r_2, y, \pi, \pi^a, b_D, b_S]$ of the bank's own linear model, which is Equations (11.1) to (11.3) with $\hat{r}^*$ in place of r^*, the bias b_D in place of d_D and b_S in place of d_S, $\dot{b}_D = 0$, $\dot{b}_S = -b_S/\tau_b$ with $\tau_b = 1\,\text{yr}$, and the anchored share read each week as an exact survey and frozen over a forecast. The demand bias absorbs whatever the bank has wrong about the neutral rate, so its implied estimate is $\hat{r}^* + b_D/\sigma$. The offset-free controller of Volume 9, Chapter 4 carries an unmeasured fuel bias in its estimator in the same way. The estimate is propagated every week by fourth-order Runge-Kutta with the rate in force, and the bank keeps its own monthly and quarterly averages of $\hat{\pi}$ and $\hat{y}$. Each release corrects it at the middle of the period it describes and carries the correction to now, as shown in Equation (11.10).

$$\begin{aligned} &\hat{x} \leftarrow \hat{x} + e^{A(t-t_m)} L \nu \\ &L = \begin{cases} (0,0,0,0.5,0.25,0,0.4)^\top & \text{CPI print} \\ (0,0,0.8,0,0,1,0)^\top & \text{GDP print} \end{cases} \end{aligned} \tag{11.10}$$

where A is the bank model's state matrix, t_m is the middle of the month or quarter, and the innovation ν is the print less the bank's own average for a CPI print or a first GDP print, and the change from the previous print for a revision. The gains are teaching values, not a Kalman design. A perfect-state setting gives the MPC the true states and the true shocks as its biases, as a labeled comparison.

11.5.6 Forecast Targeting as Model Predictive Control

At every decision, and whenever a setting or a switch has made its last plan stale, the MPC discretizes the bank model exactly for a zero-order hold over one decision interval, $T_s = 1/8$ year on the default calendar and never less than a month, and solves the quadratic program (QP) of Equation (11.11) with the framework's linear MPC, `core/linmpc.js` (Section 1.6.3).

$$\begin{aligned} \min_{u_0,\ldots,u_9} \; & \textstyle\sum_{k=1}^{N} T_s\left[(\hat{\pi}_k - \pi^*)^2 + \lambda_y \hat{y}_k^2\right] \\ & + \nu \textstyle\sum_{j=0}^{9} (u_j - u_{j-1})^2 \\ \text{s.t.} \; & i_{min} \le u_j \le i_{max}, \qquad |u_0 - i| \le \Delta \\ & |u_j - u_{j-1}| \le \Delta\, p\, n_{j-1}, \qquad j = 1,\ldots,9 \end{aligned} \tag{11.11}$$

where $N = H/T_s = 24$ steps, $u_{-1} = i$, the ten blocks hold $n_j = 1,1,1,1,2,2,3,4,6,3$ steps, p is the number of decisions a step spans, one on the default calendar and $52/12$ on the continuous one, $\lambda_y = 0.5$ and $\nu = 0.3$. The program has 10 variables and 40 rows, and a ridge of $10^{-9}(1 + d_{max})$, with d_{max} the largest diagonal entry of the Hessian, keeps it strictly convex. The dual active-set method of [19] solves it, the first move becomes the request and the calendar shapes it, and a failed solve holds the rate and is counted. Over the 161 solves of case 8 it needs 1.18 active-set iterations on average and 12 at most, with none failing. The forecast freezes the anchored share and knows nothing of shocks to come.

With a one-step horizon the program has a closed form. If b_π is the inflation entry of the discrete input matrix and $\hat{\pi}_1$ the one-step forecast with the rate held, the move is

$$\Delta i = \frac{T_s\,|b_\pi|}{\nu + T_s b_\pi^2}\,(\hat{\pi}_1 - \pi^*) \tag{11.12}$$

a velocity-form integral move on the forecast error. The application checks that the shared PID run as a pure integral controller with that gain gives the same move to 10^{-12} and that the QP agrees to the ridge, on a horizon below the slider's one year that only the check can set. Over one meeting the rate barely reaches inflation, $b_\pi = -4.47 \times 10^{-5}$, so with the check's $\nu = 1$ and a forecast of 2.96 % the move is 5.35×10^{-6} points and both policies hold the rate. Forecast targeting earns its lead only over a horizon long enough for the lags to act.

11.5.7 Economy, Belief and Benchmark

Equations (11.1) to (11.5), with the shocks, the drift of the anchored share and the release noise, are the simulated economy. The bank's model is their linear copy with the believed neutral rate, two biases, a frozen anchored share and no foresight, corrected only by the releases. The model both stand for is an estimated quarterly economy such as [7] or a simple backward-looking model such as [8], whose coefficients were estimated or derived and not copied here. No model is identified from data inside the application, and the PID and the Taylor rule use no model at all.

11.6 Measured Runs of the Economy

Table 11.4 lists the cases. Each starts from the defaults of Tables 11.2 and 11.3, selected from the dropdown, with the change named in its row, and runs for its own length at the default seed (Section 1.6.4). A variant is the case with one setting changed before Run, and the figures were produced by the engine alone. Table 11.5 collects the outcomes. Table 11.7 lists what each case requires before the application credits it. A case is credited once its listed actions have been taken and the economy has reached the listed state, and the challenge counts only on its own model and disturbances, so changing either voids an attempt. The integrated loss is $\int[(\pi - \pi^*)^2 + \lambda_y y^2]\,\mathrm{d}t$ on the true states with the MPC's weight, so it is compared only within one case and one run length.

Table 11.4 The Nine Cases, the Challenge and the Reference

#	Change from the reference, run length	Teaching point
1	chair in MANUAL, boom at 0.25 yr, 8 yr	the overshoot the lags make
2	Taylor rule, the same boom, 10 yr	a proportional rule
3	$a_\pi = 0.8$, $\lambda_0 = 0$, boom, 40 yr	the Taylor principle
4	PID, r^* falls a point at 1 yr, 20 yr	integral action
5	MPC, oil shock at 1 yr, 20 yr	an infeasible frontier
6	Taylor rule, boom at 1 yr, noise-free data, 10 yr	cadence and quantum
7	PID with $\tau_I = 2$ yr, crisis at 1 yr, 12 yr	the floor and windup
8	MPC, boom, oil and crisis at 1, 5 and 9 yr, 20 yr	optimizer against PID
9	$\pi = \pi^a = 10\,\%$, $y = 2.4$, $\lambda = 0.3$ drifting, 20 yr	the sacrifice ratio
C	oil at 6 %/yr at 1 and 5 yr, $\lambda_0 = 0.4$ drifting, 15 yr	a 1970s decade
R	noise-free data, no shock, 5 yr	the equilibrium

Table 11.5 Final and Peak Inflation, Lowest Gap, Peak Unemployment, Worst Calendar-Year Growth and Loss Over Each Case's Own Run

Case and configuration	π_{end}	π_{max}	y_{min}	u_{max}	g_{min}	Loss
1, rate held at 2.875 %	2.67	3.24	0.00	4.00	1.19	9.84
1, ±25 bp while $\lvert\pi_{\text{pub}} - 2\rvert > 0.25$	−1.82	3.05	−4.93	6.47	−0.20	63.67
2, Taylor rule	2.06	2.95	−0.90	4.45	0.44	4.24
2, $a_\pi = 2.5$	1.95	2.90	−1.33	4.66	0.23	4.85
2, $a_y = 1$	2.03	2.88	−1.10	4.55	0.39	3.93
2, MPC	2.06	2.85	−1.66	4.83	0.13	5.19
3, $a_\pi = 0.8$	4.77	4.77	−0.23	4.12	0.78	118.95
3, $a_\pi = 1.5$	2.08	3.10	−0.72	4.36	0.55	7.90
4, Taylor rule	1.52	2.00	−0.61	4.31	1.64	4.60
4, PID	1.86	2.00	−0.61	4.31	1.64	1.78
4, PID, $\tau_I = 4$ yr	1.98	2.00	−0.61	4.31	1.64	1.05
4, MPC	2.01	2.15	−0.58	4.29	1.68	0.68
5, MPC, $\lambda_y = 0.05$	1.97	4.75	−2.20	5.10	0.61	12.68
5, MPC, $\lambda_y = 0.5$	1.99	4.77	−1.26	4.63	1.21	15.73
5, MPC, $\lambda_y = 5$	1.99	4.81	−0.49	4.24	1.61	22.82
5, Taylor rule	1.96	4.72	−2.23	5.12	0.65	15.84
6, Taylor, weekly	2.02	2.85	−0.80	4.40	0.83	3.46
6, Taylor, 8 a year	2.02	2.87	−0.86	4.43	0.77	3.64
6, Taylor, quarterly	2.02	2.90	−0.97	4.48	0.65	4.03
7, PID, floor tracking	2.04	2.74	−3.25	5.63	−0.24	17.43
7, floor tracking off	2.08	3.23	−3.25	5.63	−0.24	22.84
7, $i_{min} = -2\,\%$	1.99	3.20	−3.25	5.63	−0.24	22.58
7, $\pi^* = 1\,\%$	1.12	1.52	−3.25	5.63	−0.24	17.34
8, MPC	2.02	4.30	−3.49	5.75	0.43	28.93
8, PID	1.91	4.43	−3.86	5.93	0.70	41.56
8, Taylor rule	1.96	4.58	−3.32	5.66	0.80	32.57
8, MPC, true state	1.91	4.38	−3.48	5.74	0.10	24.26
9, Taylor rule	1.99	10.00	−3.75	5.88	−0.83	239.23
9, $a_\pi = 2$, $a_y = 0$	1.96	10.00	−5.79	6.90	−1.02	235.94
9, MPC, $\lambda_y = 2$	2.10	10.00	−1.98	4.99	−0.20	295.82

Inflation, gap and unemployment in %, growth in %/yr, all on the true states.

11.6.1 The Chair and the Rule

In case 1 a fiscal stimulus adds two points of demand for two years from year 0.25. Left at 2.875 %, the boom takes the gap to +1.89 % at year 2.3 and inflation to 3.24 % at year 3.3. The first CPI print above 2.25 % is published at year 0.38, and it is noise on a month whose true average was 2.00 %, while the true gap is already +0.33 %. A chair who raises 25 bp at every meeting while the print is above 2.25 %, and cuts while it is below 1.75 %, keeps raising for a year and a half after inflation peaked at 3.05 % at year 2.8, reaches 8.875 % at the meeting of year 4.3, and takes the gap to −4.93 % at year 6.1, unemployment to 6.47 % and inflation to −1.82 % at year 8, with a recession on the way. Steps of 50 bp make it worse, a rate of 11.875 % and a gap of −6.93 %. That chair is an integral

controller on a plant with two years of lag.

The Taylor rule on the same boom in case 2 raises the rate to 5.625 % at year 2.5, holds inflation to 2.95 % and the gap between +1.61 % and −0.90 %, and scores a ten-year loss of 4.24. A stronger inflation response mostly makes a bigger move at the wrong time. At $a_\pi = 2.5$ the inflation peak falls only to 2.90 % while the gap falls to −1.33 % and the loss rises to 4.85. A stronger gap response helps, 3.93 at $a_y = 1$ with inflation peaking at 2.88 %, because on a demand shock the gap moves a year before inflation and is the earlier warning of the same disturbance. Dropping the gap term costs 5.49, a partial adjustment of 0.7 smooths the rate to 41 moves instead of 66 for a loss of 5.17, and the MPC at its shipped weight scores 5.19 by pushing the gap to −1.66 %.

> **TRY IT IN THE APP 2 · The Taylor Rule on the Same Boom**
>
> Load the case and press Run. Watch the applied policy rate on the trend. The rule raises it in quarter-point steps to a peak of 5.625 % at year 2.5, inflation peaks at 2.95 %, and after ten years the scorecard gives an integrated loss of 4.24.

11.6.2 The Principle, the Bias and the Frontier

In case 3 no one is anchored and the rule answers each point of inflation with 0.8. The real rate then sits on average 0.19 points below the true neutral rate from year 10 on, the gap stays near +0.19 %, and inflation climbs without end, 2.92 % at year 5, 3.17 % at 10, 3.48 % at 20 and 4.77 % at 40. At 1.5 the same boom dies out, 2.47 % at year 10 and 2.08 % at 40. The climb is slow because the gap that drives it is small, and its direction is the lesson.

In case 4 the true neutral rate falls from 0.875 % to −0.125 % at year 1 and the bank is not told. The Taylor rule, a proportional controller with the wrong intercept, can run a lower rate only while inflation stays below target, and it settles at 1.52 % at year 20. The PID's integral moves the intercept down by 0.84 points by year 20 and ends at 1.86 %, and an integral time of four years moves it 1.05 points and ends at 1.98 %. The integral time that brings inflation within 0.05 points of the target by year 20 therefore lies between those two, since eight years leave 0.14 points and four leave 0.02. The MPC, whose estimator carries a demand bias, ends at 2.01 % with a loss of 0.68 against the rule's 4.60. The integral finds the neutral rate that real-time data misstate [14].

> **TRY IT IN THE APP 4 · The PID Finds a Neutral Rate That Moved**
>
> Load the case and run twenty years: under the PID inflation ends at 1.86 %. Then set the controller to the Taylor rule, press Reset and run again. Without an integral the rule settles at 1.52 %, below the target for good, because a proportional controller with the wrong intercept needs an error to hold a lower rate.

In case 5 an oil shock adds 3 percent a year to inflation for a year. At the shipped weight the MPC lets inflation reach 4.77 % at year 2.4 and the gap fall to −1.26 % at year 3.6. Across the slider the peak inflation error stays between 2.75 and 2.81 points while the worst gap ranges from −2.20 % to −0.49 %, and even the MPC given the true state, at a weight of 0.05, peaks 2.59 points over target. No output weight, rule or information keeps inflation and the gap both within a point, which the application's check confirms over seven weights, the true state, the Taylor rule and the PID. The weight trades the RMS gap, from 0.64 to 0.20 points, against the RMS inflation error, from 0.78 to 0.97, as shown in Figure 11.2(b). The Taylor rule sits at 0.64 and 0.77, a little inside the MPC's low-weight end, because the MPC's curve is the frontier of its own family on an estimated state with a frozen anchor, not the efficient frontier of the economy [8]. The true state moves it to 0.71 and 0.70.

11.6.3 Cadence and the Floor

Case 6 repeats the boom with noise-free releases, so that runs differ only in their calendar. Over ten years the Taylor rule's loss is 3.46 deciding every week, with 22 moves in 520 decisions, 3.53 monthly, 3.64 at eight meetings a year, with 19 moves in 80, and 4.03 quarterly, and inflation peaks

at 2.85 %, 2.86 %, 2.87 % and 2.90 %. The lag dominates the sampling. The MPC moves harder and gains more from frequent decisions, 3.40 weekly against 4.23 at eight meetings and 5.44 quarterly. The relay cycle of a coarse quantum is the subject of Section 11.4.

In case 7 a financial crisis takes six points off demand at year 1. The PID, given a two-year integral time so that windup shows, reaches the floor at year 2.1 and spends 2.25 years there. The gap falls to −3.25 % at year 2.0 and inflation to 0.34 % at year 3.1. The trough is the lag's, not the floor's. With the floor removed, $i_{min} = -2$ %, it is the same −3.25 %, because the crisis does its damage before any cut reaches spending. With floor tracking off the integral winds up, the PID asks for −2.79 % at year 3.5, holds the floor until year 5.1 instead of 4.5, and the recovery overshoots into OVERHEATING, a gap of +2.19 % at year 5.75 and inflation of 3.23 % at year 7.6, raising the loss from 17.43 to 22.84. A target of 1 % lines the economy out at a neutral rate a point nearer the floor, so the rate spends 3.71 years at the floor and inflation is negative for 2.71 years, down to −0.74 %, while the gap's trough stays −3.25 %.

> **DID YOU KNOW? Removing the Floor Does Not Lower the Loss**
>
> From the simulator: with the lower bound moved from 0.125 % to −2 % in case 7, the gap's trough stays at −3.25 %. The PID with floor tracking at the real floor scores a loss of 17.43, the run with the floor at −2 % scores 22.58 (Table 11.5).

11.6.4 The Optimizer and Credibility

Case 8 fires a boom at year 1, an oil shock at 5 and a crisis at 9 with the same prints in every run. The MPC's twenty-year loss is 28.93 against the PID's 41.56 and the Taylor rule's 32.57, with lower peaks of unemployment (5.75 % against 5.93 %) and inflation (4.30 % against 4.43 %) than the PID. Its worst calendar year is worse, 0.43 percent growth in the year ending at 10 against the PID's 0.70 in the year ending at 5, and it has two recessions, at years 3.75 and 9.50, to the PID's none, because it brings the gap down faster and growth is the gap's rate of change. Given the true state its loss falls to 24.26, a larger gain than its whole lead over the Taylor rule, so in this economy information is worth more than the optimizer.

Case 9 starts at 10 % inflation held steady by a gap of +2.4 % against an anchor of 0.3 that drifts with performance. The Taylor rule raises the rate to 14.125 % at year 2.0, the gap falls to −3.75 % at year 3.6, and inflation is within half a point of the target for good from year 12.9, by which time half the public is anchored again, 0.78 by year 20. The disinflation costs 13.8 gap point-years, a sacrifice ratio of 1.73 per point. A faster rule, $a_\pi = 2$ and $a_y = 0$, gets there at year 5.4 for a ratio of 2.43 and unemployment of 6.90 %, and the MPC at $\lambda_y = 2$ takes 15.7 years at a ratio of 1.35. A faster disinflation costs more output in this model, and whether speed buys credibility faster in an economy that learns is the question the backward model cannot answer [11].

11.6.5 A 1970s Decade

The challenge fires two oil shocks of 6 percent a year for a year, at years 1 and 5, with 40 percent of the public anchored and the share drifting, eight meetings a year and 25 bp steps. It asks for inflation within half a point of 2 % throughout years 12 to 15, no calendar year of growth below −2 % and unemployment never above 8 %, judged on the true states. Table 11.6 compares six configurations, and Figure 11.4 charts two of its columns. As shipped, the Taylor rule lets inflation reach 8.26 % at year 6.4 and expected inflation 4.94 %, and STAGFLATION shows for 5.4 years. In the slump after the second shock its gap term eases while expectations are still high. The real rate is zero at year 10 with inflation at 3.23 %, inflation is up to 1.10 points off target in the last three years, and the anchored share, 0.48 at year 2, has fallen to 0.38 by year 8 and is still 0.38 at year 12. The growth and unemployment limits are met, with a worst year of 0.10 % and unemployment of 5.85 %, so the rule fails on the late inflation alone.



Figure 11.3 The two-oil-shock challenge at $t = 15.00$ yr under forecast-targeting MPC with an output weight of 0.2. The figure is a composite of the Economy view, eight scorecard tiles set at its right and the Trends panel with a 20-year window. The header and the control panel are cut. The badges read inflation 2.05 %, gap −0.15 %, rate 3.375 %, unemployment 4.1 %, growth 1.7 %, real rate 1.33 %, anchored share 0.70 and a last move 1.1 quarters ago. (1) The valve holds 3.375 % while the MPC asks 3.35 %, with the next decision in four weeks. (2) The two pipes, tinted against the hidden 0.88 % neutral rate, carry 1.33 % into the first stage, 1.38 % out of it and 1.28 % out of the second. (3) The published gap reads 0.56 % against a true −0.15 %, and (4) the thermometer's CPI print 1.93 % against a true 2.05 % and an expected 2.0. (5) The meeting strip counts seven moves in the last two years, four up and three down. In the trends the published CPI and the rate rise twice after the shocks at years 1 and 5 (dotted marks), and the dashed marks are, by the run's record, DE-ANCHORED at the start and STAGFLATION at years 2.77 and 5.10. The scorecard gives a loss of 152.82, peak inflation 8.18 %, peak unemployment 6.14 %, a worst calendar year of 0.25 % growth, no time at the floor, no recession, 121 solves with no failure and the challenge met.

Table 11.6 The Challenge of 15 Years in Six Configurations

Configuration	$\lvert\pi - \pi^*\rvert_{12-15}$	g_{min}	u_{max}	π_{max}	Target
as shipped, Taylor rule	1.10	0.10	5.85	8.26	not met
MPC, $\lambda_y = 0.2$, suggested	0.16	0.25	6.14	8.18	met
MPC, $\lambda_y = 0.5$	0.92	0.46	5.77	8.70	not met
Taylor rule, $a_y = 0$	0.17	0.06	6.06	7.54	met
Taylor rule, $a_\pi = 2.5$	0.46	−0.11	6.48	7.57	met
PID	1.99	−0.01	6.75	7.55	not met

Limits 0.5 points, −2 %/yr and 8 %. Deviation in points, growth in %/yr.

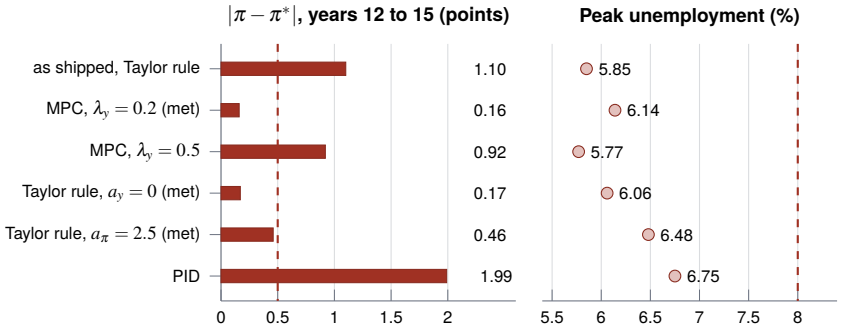


Figure 11.4 Two columns of Table 11.6, the 1970s challenge over 15 years: the largest inflation error over years 12 to 15 (bars, values at the right) against the limit of 0.5 points, and the peak unemployment (dots) against the limit of 8 % (dashed lines). Notice that every configuration that meets the target peaks higher in unemployment than the shipped rule and the MPC at its shipped weight, and that no configuration comes near the unemployment limit.

The suggested MPC, at an output weight of 0.2, keeps the real rate up until its forecast is at target, 0.63 % at year 10 against the rule's zero, and holds inflation within 0.16 points over years 12 to 15 (Table 11.6), with unemployment peaking at 6.14 % and a worst year of 0.25 %, as the scorecard of Figure 11.3 shows. Inflation returns within a point of the target early enough for the anchor to drift back, from 0.38 at year 8 to 0.56 at year 12 and 0.70 at year 15, against the rule's 0.55 at the end. A Taylor rule with no gap response also passes (0.17), as does an inflation response of 2.5 (0.46, at unemployment of 6.48 % and a year of –0.11 %), while the MPC at its shipped weight of 0.5 does not (0.92) and neither does the PID (1.99). The verdict is not a property of one seed. On the default seed and seeds 1 to 4 the shipped rule misses by 1.06 to 1.13 points and the suggested MPC passes with 0.10 to 0.23, and over seeds 1 to 20 the rule fails every time (1.06 to 1.28) and the MPC passes every time (0.10 to 0.37). No configuration of Table 11.6 comes near the growth or unemployment limit, the worst year being –0.11 % and the highest unemployment 6.75 %. Judged against a stricter unemployment limit of 6 %, none of the six configurations meets the challenge. Both lessons of the decade are about the weight on output, which is a tuning parameter here.

11.7 Technology in Practice

The bank of Section 11.5 reads two published statistics and moves one administered rate on a calendar. This section sets the real statistics, the rate and the committee beside the model.

What Measures It

Inflation and output are compiled from surveys and records, not read by a device. In the United States the consumer price index of a month is released in the middle of the next month, and the first estimate of a quarter's output about one month after the quarter, followed by a second and a third estimate and later comprehensive revisions [20]. The model's CPI print two weeks after the month, with noise σ_P and no revision, and its three GDP vintages of Section 11.3 stand for these releases. The published gap also needs potential output, whose end-of-sample estimate is the larger source of error in real time (Section 11.2). The model gives the print noise but no bias, so the fault described below is one the simulated bank never meets.

Table 11.7 Scenarios and Completion Requirements

#	Scenario	Kind	Required to complete
1	The chair	Experiment	In MANUAL, request a rate above 3.125 %, keep chairing to year 8, and confirm that the first rise crossed both transmission stages before the gap turned.
2	Taylor rule	Experiment	Ten years with the rule as shipped ($a_\pi = 1.5$), then ten years with $a_\pi = 2.5$ after Reset.
3	Taylor principle	Demonstration	With no anchor, forty years at $a_\pi < 1$ that end with inflation more than 2 points over target, then forty years at $a_\pi = 1.5$ after Reset that end within 0.3 points of it.
4	Neutral rate	Experiment	Twenty years of the PID with integral action after the hidden fall in r^*, then twenty years of the Taylor rule after Reset.
5	Oil shock	Demonstration	Twenty years of the MPC, then twenty more after Reset with $\lambda_y \leq 0.1$ or $\lambda_y \geq 2$, and confirm that no run keeps inflation and the gap both within a point.
6	Cadence	Experiment	Ten years at eight meetings a year with the 25 bp quantum, ten years on the weekly or the quarterly calendar after Reset, then thirty years with a quantum of at least 50 bp.
7	The floor	Experiment	With floor tracking on, run until the rate reaches the floor and on to year 12, then twelve years with floor tracking off and twelve with a target of at most 1 %, each after Reset.
8	MPC and PID	Experiment	Twenty years of the MPC through the three shocks, then twenty years of the PID after Reset.
9	Disinflation	Experiment	After year 1, hold inflation within 0.5 points of the target for three years with the shipped rule, then again with $a_\pi = 2$ and $a_y = 0$ after Reset.
C	1970s challenge	Challenge	Over the full 15 years, inflation within 0.5 points of the target from year 12 to 15, no calendar year of growth below −2 % and unemployment never above 8 %, on the case as given.[a]
R	Reference	Exploration	Three years of the equilibrium in AUTO, then in MANUAL a request of at least 3.375 % held until the gap falls below −0.2 %.

[a] As shipped, the Taylor rule misses by 1.10 points (worst year +0.10 %, peak unemployment 5.85 %). The suggested MPC at $\lambda_y = 0.2$ passes with 0.16 points, a margin of 0.34 on the band, a worst year of +0.25 % and unemployment of 6.14 %, revealed before Run or carried through Reset alike.

> **TECHNOLOGY** **Sensor: The Consumer Price Index**
>
> No instrument reads inflation. A statistical agency collects prices and weights them with expenditure estimates from a consumer expenditure survey [21]. In the United States the index of a month is released in the middle of the next month, 18 days after the month ends [20]. Its typical fault is a bias, not noise. A review in 2001 estimated that the index overstated the change in the cost of living by about 0.6 percentage point a year, with a confidence interval from 0.1 to 1.2, roughly half of it from quality change and new items, against an estimate of about 1.1 points made in 1996 [21]. On a trend a bias does not show: the print looks steady while the truth sits below it.

What Moves It

The final element is an administered short rate. A committee sets a target range for an overnight rate, and administered rates, among them the rate paid on reserve balances, support that range [22]. The quantum q of 25 bp and the grid of midpoints of Equation (11.7) copy a range a quarter point wide, the cap Δ limits a meeting's move, and the floor i_{min} stands for the effective lower bound. Figure 11.5 draws the loop with these institutions in place of the schematic's symbols.

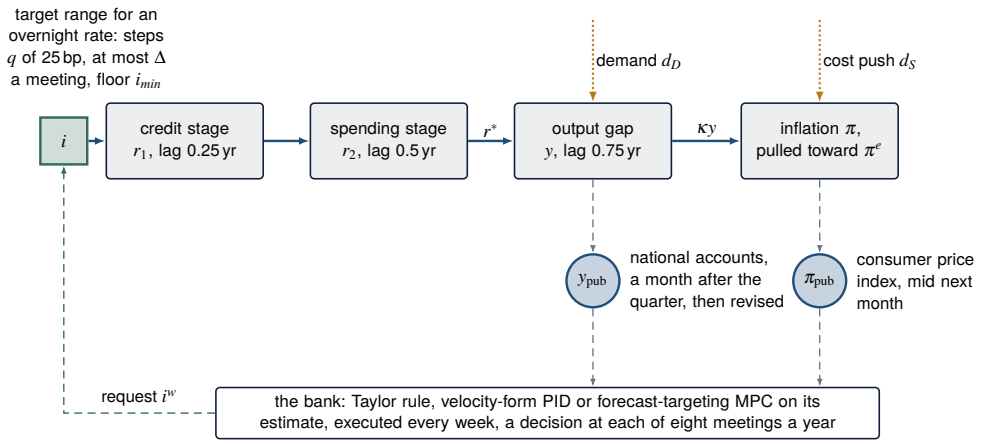


Figure 11.5 The policy loop with the institutions of a real central bank in place of the schematic's symbols. A price index and the national accounts give the published inflation π_{pub} and gap y_{pub}, the bank's policy forms a request each week, and only a meeting turns it into a step of the target range for the rate i. Notice the three lags in series between the rate and the gap, and that both readings leave the economy after its last stages and return weeks late.

> **TECHNOLOGY** **Final Element: A Target Range for an Overnight Rate**
>
> A central bank moves a rate it administers. The Federal Open Market Committee's primary means of adjusting the stance of policy is a change in the target range for the federal funds rate [13]. The Board sets the rate paid on reserve balances, an important tool, and the implementation note issued with each decision gives the settings of the tools that support the target range. The posted rate is updated each business day with the next business day's rate [22]. The element saturates: the same statement names a funds rate constrained by its effective lower bound [13], and central banks have judged that bound to be close to zero [23].

Where the Controller Runs

No computer closes this loop. The Federal Open Market Committee holds eight regularly scheduled meetings a year and other meetings as needed [24]. The simulated bank executes every week, $\Delta t = 1/52$ year, and its default calendar makes one execution in six to eight weeks a decision, so the meeting, not the weekly execution, is the period that counts. In the application the Taylor rule, the velocity-form PID of Equation (11.9) and the MPC fill the same slot ahead of the framework's shared calendar module, `core/cadence.js`, which rounds, caps and holds each request.

> **IN PRACTICE** **Rules as Benchmarks, Not Autopilots**
>
> The Federal Open Market Committee regularly consults the prescriptions of several rules with other information and does not follow them mechanically. Its published set holds the Taylor rule, a balanced-approach rule with twice the gap coefficient, a rule adjusted for the effective lower bound, an inertial rule that keeps 0.85 of the previous rate, and a first-difference rule that sets only the change of the rate and needs no estimate of the neutral rate or of potential output. Following a rule strictly would first require a choice of inflation measure, activity measure and neutral rate [23]. The partial adjustment ρ of Equation (11.8) has the inertial form, and a first-difference rule is an integral controller in velocity form.

What Is Modernizing It

Three developments sharpen what the bank reads. The first is nowcasting. A dynamic factor model in state-space form reads many series as they are released, a Kalman filter updates its estimate with each

release, and one central bank's staff nowcast is updated daily when new data arrive and published weekly [20]. The second is nowcasting of prices alone: a model built on relatively few variables nowcasts headline and core inflation in the consumer price index and in the price index for personal consumption expenditures, with accuracy that improves as information accumulates over a month or a quarter [25]. The third is a filter for the unobserved intercept. The natural rate of interest, potential output and trend growth can be estimated jointly by a Kalman filter [26]. The bank's estimator of Equation (11.10) is a filter of the same kind with fixed teaching gains, whose demand bias stands in for an error in the neutral rate, and forecast targeting by MPC is the chapter's own model-based plan on that estimate.

11.8 Limitations

The economy is a teaching model. Its expectations look backward, so the Lucas critique applies with full force [11], and the objection that control theory does not apply to planning under rational expectations [5] is avoided by assumption, not answered. The lags, the slope of the Phillips curve and every threshold are teaching values, the design windows for the troughs are the application's own, and no number here is an estimate for any country. The output gap and Okun's law are local and linear.

The bank's estimator has fixed teaching gains rather than a Kalman design, reads the anchored share as an exact survey, and is corrected only by the two releases. The MPC freezes the anchored share over its horizon, knows nothing of shocks to come, and uses one decision interval as its step with blocked moves later in the horizon. The policy rate is the only instrument, so a crisis that needs a rate below the floor has no other answer. The statistics are constructed averages of the model's own states.

The inflation target, the quantum and the floor also set the starting state. A change made before Run restarts the case with it, as Reset does (Section 1.6.1), so the application gives one run whichever way a setting is chosen.

> **OPEN CHALLENGE** **Where Is the Neutral Rate Now?**
>
> Every rule with an intercept needs the neutral real rate, which no one observes. A study that estimated it with potential output and trend growth by a Kalman filter found substantial variation over four decades in the United States, an average standard error of 1 to 2-1/2 percentage points, and a significant deterioration of stabilization when policymakers mismeasure the rate, which timely updating of the estimates can mitigate [26]. The fall of one point in case 4 lies inside that error. Progress would be estimates precise enough in real time to set the intercept, or rules that, like the first-difference rule, do not need one [23].

11.9 Conclusion

This chapter's simulator joins a continuous economy of three lags to a statistics agency that publishes late averages with revisions and to a bank that may move one rate on a calendar, in quarter-point steps, above a floor.

A rise held a quarter peaks in output after 1.04 years and in inflation after 2.40, and a chair who chases the print takes the rate to 8.875 % and the gap to −4.93 %. The Taylor principle's boundary falls from 1.0 with no anchor to 0.25 with 30 percent anchored, the PID's integral finds an unannounced fall in the neutral rate where the rule settles at 1.52 %, and no policy holds both inflation and the gap within a point through an oil shock. Deciding every week lowers a boom's loss only from 3.64 to 3.46, the floor's trough is the lag's at −3.25 %, and in the 1970s challenge the shipped rule misses by 1.10 points where forecast targeting at an output weight of 0.2 holds 0.16.

The expectations are backward-looking and every coefficient is a teaching value. A fuller model would add a learning public, a second instrument such as asset purchases or forward guidance, and an estimator designed from the release statistics. The simulator makes the lag, the vintage and the calendar of monetary policy measurable in the same loop.

KEY POINTS

- The economy answers a rate move in output after about a year and in prices after about two, so a chair who keeps raising while the print is high takes the rate to 8.875 % and the gap to −4.93 %.
- The Taylor principle's boundary falls from 1.0 with no anchor to 0.25 with 30 percent of the public anchored.
- The PID's integral finds an unannounced fall in the neutral rate where the proportional rule settles at 1.52 %.
- Deciding every week lowers a boom's loss only from 3.64 to 3.46, because the lag dominates the sampling.

Where else this applies. Volume 5, Chapter 9 moves a contribution through the same decision calendar, quantum and cap, and Volume 5, Chapter 3 controls a column through a late measurement.

References

[1] A. W. Phillips, "Stabilisation policy in a closed economy," *The Economic Journal*, vol. 64, no. 254, pp. 290–323, 1954.

[2] R. Pindyck, "An application of the linear quadratic tracking problem to economic stabilization policy," *IEEE Transactions on Automatic Control*, vol. 17, no. 3, pp. 287–300, 1972.

[3] J. B. Taylor, "Discretion versus policy rules in practice," *Carnegie-Rochester Conference Series on Public Policy*, vol. 39, pp. 195–214, 1993.

[4] L. E. O. Svensson, "Inflation forecast targeting: Implementing and monitoring inflation targets," *European Economic Review*, vol. 41, no. 6, pp. 1111–1146, 1997.

[5] F. E. Kydland and E. C. Prescott, "Rules rather than discretion: The inconsistency of optimal plans," *Journal of Political Economy*, vol. 85, no. 3, pp. 473–491, 1977.

[6] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, III, *Process Dynamics and Control*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.

[7] G. Rudebusch and L. E. Svensson, "Policy rules for inflation targeting," National Bureau of Economic Research, Cambridge, MA, NBER Working Paper w6512, Apr. 1998.

[8] L. Ball, "Efficient rules for monetary policy," *International Finance*, vol. 2, no. 1, pp. 63–83, 1999.

[9] A. W. Phillips, "The relation between unemployment and the rate of change of money wage rates in the United Kingdom, 1861–1957," *Economica*, vol. 25, no. 100, pp. 283–299, 1958.

[10] M. Friedman, "The lag in effect of monetary policy," *Journal of Political Economy*, vol. 69, no. 5, pp. 447–466, 1961.

[11] R. E. Lucas, "Econometric policy evaluation: A critique," *Carnegie-Rochester Conference Series on Public Policy*, vol. 1, pp. 19–46, 1976.

[12] R. Clarida, J. Galí, and M. Gertler, "The science of monetary policy: A new Keynesian perspective," *Journal of Economic Literature*, vol. 37, no. 4, pp. 1661–1707, 1999.

[13] "Statement on longer-run goals and monetary policy strategy," Board of Governors of the Federal Reserve System, Federal Open Market Committee. Adopted effective January 24, 2012; as reaffirmed effective January 27, 2026, 2026, accessed 2026-11-20. [Online]. Available: https://www.federalreserve.gov/monetarypolicy/files/FOMC_LongerRunGoals.pdf

[14] A. Orphanides, "Monetary policy rules based on real-time data," *American Economic Review*, vol. 91, no. 4, pp. 964–985, 2001.

[15] A. Orphanides and S. van Norden, "The unreliability of output-gap estimates in real time," *Review of Economics and Statistics*, vol. 84, no. 4, pp. 569–583, 2002.

[16] G. B. Eggertsson and M. Woodford, "Zero bound on interest rates and optimal monetary policy," *Brookings Papers on Economic Activity*, vol. 2003, no. 1, pp. 139–233, 2003.

[17] L. J. Christiano, M. Eichenbaum, and C. L. Evans, "Nominal rigidities and the dynamic effects of a shock to monetary policy," *Journal of Political Economy*, vol. 113, no. 1, pp. 1–45, 2005.

[18] "FOMC statement, december 16, 2008," Board of Governors of the Federal Reserve System, press release, 2008, accessed 2026-11-20. [Online]. Available: https://www.federalreserve.gov/newsevents/pressreleases/monetary20081216b.htm

[19] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[20] B. Bok, D. Caratelli, D. Giannone, A. Sbordone, and A. Tambalotti, "Macroeconomic nowcasting and forecasting with big data," Federal Reserve Bank of New York, Staff Report 830, 2017. [Online]. Available: https://www.newyorkfed.org/medialibrary/media/research/staff_reports/sr830.pdf

[21] D. E. Lebow and J. B. Rudd, "Measurement error in the consumer price index: Where do we stand?" Board of Governors of the Federal Reserve System, Finance and Economics Discussion Series 2001-61, 2001. [Online]. Available: https://www.federalreserve.gov/pubs/feds/2001/200161/200161pap.pdf

[22] "Interest on reserve balances," Board of Governors of the Federal Reserve System, 2026, accessed 2026-11-20. [Online]. Available: https://www.federalreserve.gov/monetarypolicy/reserve-balances.htm

[23] "Policy rules and how policymakers use them," Board of Governors of the Federal Reserve System, Monetary Policy Principles and Practice, 2026, accessed 2026-11-20. [Online]. Available: https://www.federalreserve.gov/monetarypolicy/policy-rules-and-how-policymakers-use-them.htm

[24] "Meeting calendars, statements, and minutes (2021–2027)," Board of Governors of the Federal Reserve System, Federal Open Market Committee, 2026, accessed 2026-11-20. [Online]. Available: https://www.federalreserve.gov/monetarypolicy/fomccalendars.htm

[25] E. S. Knotek, II and S. Zaman, "Nowcasting U.S. headline and core inflation," Federal Reserve Bank of Cleveland, Working Paper 14-03, 2014.

[26] T. Laubach and J. C. Williams, "Measuring the natural rate of interest," Board of Governors of the Federal Reserve System, Finance and Economics Discussion Series 2001-56, 2001. [Online]. Available: https://www.federalreserve.gov/pubs/feds/2001/200156/200156pap.pdf

11.10 Exercises

11.1 Concept check · 2 min

The PID of Equation (11.9) forms its error as $e_k = \pi_{\text{pub},k} - \pi^*$. In the terms of Section 1.4.3, which statement is right?

(a) Direct acting with $a = -1$, because a higher rate lowers inflation and the rate must rise with the PV

(b) Reverse acting with $a = +1$, because the bank pushes against inflation that is above the target

(c) Reverse acting with $a = -1$, because a higher rate raises inflation through the output gap

(d) Direct acting with $a = +1$, because the published CPI and the rate are both in percent

11.2 Concept check · 2 min

On the noise-free reference, a rate held one point up from $t = 0$ has moved the output gap by –0.09 % after half a year and by –0.35 % after a year. What does the reading at half a year show?

(a) The rate reaches the gap after a transport delay of about a year, and the early reading is noise

(b) The transmission is a chain of lags in series, so the response starts at once and arrives late

(c) The two-week delay of the CPI print holds back the bank, so the gap answers only to the old rate

(d) The rise waits for the next of eight meetings, so half a year covers only three or four decisions

11.3 Concept check · 2 min

In case 6, with a rate quantum of 0.5 points and the Taylor rule on eight meetings a year, the rate alternates between 2.625 and 3.125 % from year 20 to 30. An inflation response of 3 instead of 1.5 leaves the gap cycle almost unchanged. What sets the size of the cycle?

(a) The inflation response, because a proportional rule with a high gain is close to instability

(b) The noise of the CPI print, which pushes the request across the rounding boundary at random

(c) The quantum, because neither grid rate next to the neutral 2.875 % holds the economy still

(d) The integral of the rule, which winds up between meetings while the rate is held

11.4 Theory and calculation · 7 min

At a steady state with the Taylor rule acting on true values and $\hat{r}^* = r^*$, the model gives $\pi^a = \pi$, $y = \lambda x/(\kappa\tau_\pi)$ and $y = -\sigma[(a_\pi - 1 + \lambda)x + a_y y]$ for an inflation deviation $x = \pi - \pi^*$. Use $\sigma = 1$, $\kappa = 0.5$ and $\tau_\pi = 2\,\text{yr}$.

(a) Eliminate y to recover Equation (11.6), and evaluate the factor $1 + (1 + \sigma a_y)/(\kappa\sigma\tau_\pi)$ at $a_y = 0.5$. *Checkpoint:* 2.5, so the boundary is $a_\pi = 1 - 2.5\lambda$.

(b) Give the boundary on a_π at anchored shares of 0, 0.1, 0.2 and 0.3. *Checkpoint:* 1.0, 0.75, 0.5 and 0.25.

(c) At $\lambda = 0.3$, find the boundary for $a_y = 0$ and for $a_y = 1$, and explain why a stronger gap response lowers it. *Checkpoint:* 0.4 and 0.1.

11.5 Theory and calculation · 6 min

Use Equations (11.2) and (11.3) with $\kappa = 0.5$, $\tau_\pi = 2\,\text{yr}$, $\tau_e = 1.5\,\text{yr}$, the credibility drift off and no shocks.

(a) With $\lambda = 0$ and the gap held at one point, let $c = \pi - \pi^a$. Show that c settles and find the constant rate at which inflation then rises. *Checkpoint:* $c = 0.43$ points, a rise of 0.286 points a year.

(b) Case 9 starts with $\pi = \pi^a = 10\,\%$, $\lambda = 0.3$ and $\pi^* = 2\,\%$. Find the expected inflation and the gap that hold inflation still. *Checkpoint:* $\pi^e = 7.6\,\%$ and $y = 2.4\,\%$.

(c) With $u^* = 4\,\%$, $\beta = 2$ and $r^* = 0.875\,\%$, find the starting unemployment and the real rate r_2 that holds that gap, and the nominal rate $r_2 + \pi^e$ before it is rounded to the grid. *Checkpoint:* 2.8 %, −1.525 % and 6.075 %.

11.6 Theory and calculation · 5 min

Use the shaping of Equation (11.7) with $b = 0.125$, $i_{min} = 0.125\,\%$ and $i_{max} = 20\,\%$.

(a) The rate in force is 2.875 %, $q = 0.25$, $\Delta = 0.5$, and the controller asks for 3.70 % at three meetings in a row. Give the rate after each meeting. *Checkpoint:* 3.375, 3.625 and 3.625 %.

(b) With $q = 0.5$ and the largest move set to $\Delta = 0.25$, what is the largest move a meeting can make? *Checkpoint:* 0.5 points.

(c) In the application's one-step check, $T_s = 1/8$ year, $\nu = 1$, $b_\pi = -4.47 \times 10^{-5}$ and the forecast is 2.958 %. Find the move of Equation (11.12), and the forecast error that would be needed for the 0.25 grid to move the rate at all. *Checkpoint:* 5.35×10^{-6} points, and an error of about 22 000 points.

11.7 App challenge · 8 min

Load 2 · The Taylor rule on the same boom and run the full ten years at the default seed.

(a) Read the Integrated policy loss and Peak inflation tiles. *Checkpoint:* 4.24 and 2.95 %, within 0.01.

(b) Set Total inflation response to 2.5, press Reset and run ten years again. Read the same tiles and the lowest true gap, column y of the exported file. *Checkpoint:* 4.85, 2.90 % and −1.33 %, within 0.01.

(c) Set Total inflation response back to 1.5 and Output-gap response to 1, press Reset and run again. *Checkpoint:* 3.93 and 2.88 %, within 0.01.

Explain why the stronger gap response helps on this boom and the stronger inflation response does not.

11.8 App challenge · 8 min

Load 4 · The PID finds a neutral rate that moved. The true neutral rate falls a point at year 1 and the bank is not told.

(a) Run twenty years at the default seed. Read the last value of `pi` in the exported file and how far the PID card says the integral has moved the intercept. *Checkpoint:* 1.86 % and −0.84 points, within 0.01.

(b) Set Integral time (0 disables) to 0, press Reset and run twenty years. Then set Controller to Taylor rule, press Reset and run again. Read inflation at year 20 and the Integrated policy loss for both runs. *Checkpoint:* 1.52 % and 4.60 in both runs, within 0.01.

(c) Set Controller to PID + output gap and Integral time to 4, press Reset and run twenty years. *Checkpoint:* 1.98 %, −1.05 points and a loss of 1.05, within 0.01.

Explain the result of part (b) with Section 1.4.2.

11.9 App challenge · 9 min

Load 7 · The floor: a financial crisis, in which the PID has a two-year integral time.

(a) Run twelve years at the default seed. Read Time at rate floor, Peak inflation and Integrated policy loss. *Checkpoint:* 2.25 yr, 2.74 % and 17.43, within 0.01.

(b) Switch to Advanced, set PID floor tracking to Off, press Reset and run twelve years. Read the same tiles, and find the lowest request in column `want` of the exported file and its time. *Checkpoint:* 3.00 yr, 3.23 % and 22.84, –2.79 % at year 3.5, within 0.02.

(c) The lowest true gap is –3.25 % at year 2.0 in both runs. Explain why switching off floor tracking leaves the trough alone and still costs 5.4 in loss. *Checkpoint:* The trough comes before any cut reaches spending.

11.10 Application area · 4 min

The true potential growth falls from 2 to 1.5 % a year at $t = 0$, and the bank keeps believing $\hat{g}^* = 2\,\%$ a year.

(a) Use Equation (11.4) to find m after four years, and say by how much the Taylor rule with $a_y = 0.5$ then sets the rate differently from the same rule fed the true gap. *Checkpoint:* $m = -2$ points, a rate one point lower.

(b) Name a process measurement that fails in the same way, and the feedback in this chapter that removes the resulting bias. *Checkpoint:* A drifting analyzer or inferential estimate, and integral action on an unbiased measurement.

11.11 Application area · 3 min

In case 7 the rate cannot go below 0.125 %.

(a) With a target of 1 % instead of 2, the rate spends 3.71 years at the floor instead of 2.25. Explain why from the neutral nominal rate. *Checkpoint:* 1.875 % instead of 2.875 %, a point less room to cut.

(b) Name the two instruments the chapter says the model lacks, and the plant practice that the higher target corresponds to. *Checkpoint:* Asset purchases and forward guidance. Running farther from an actuator limit.

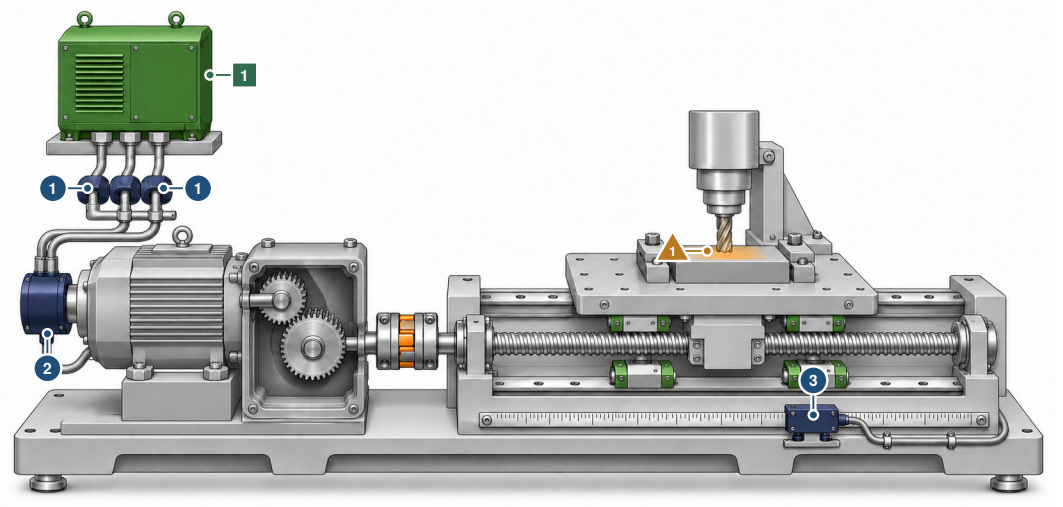
11.12 Review the agent · 5 min

An AI assistant proposes a model for an inner loop from the policy rate to the output gap: “The gap answers a rise held a quarter most after 1.04 years, so the path is a first-order-plus-dead-time process with $K_p = -1$ point of gap per point of rate, $\theta_p = 1.04\,\text{yr}$ and $\tau_p = \tau_y = 0.75\,\text{yr}$. SIMC with $\tau_c = \theta_p$

then gives $|K_c| = 0.36$ and $\tau_I = 0.75\,\text{yr}$." Find the flaw, give a better estimate of the dead time and time constant, and say how you would check it in the app.

CHAPTER 12

Commissioning a Servo Axis Whose Coupling Resonance Caps the Speed Loop



The controller moves 1 inverter that sets the motor's current

The controller reads 1 current sensors on the motor leads 2 encoder on the motor
3 position read on the load side (an encoder on the screw in the model)

What disturbs it 1 cutting force on the table

Notice that the motor's encoder and the load-side reading sit on opposite sides of the gears and the springy coupling, so the motor can read on target while the table is not. Illustration, not to scale.

IN PLAIN TERMS **Tuning a Machine Axis Around a Springy Joint**

The main idea. A motor turns a long screw that slides the table of a machine. A controller sets the motor's current, its speed and the table's position, one loop inside the next, and it is tuned from the inside out.

The challenge. The motor reaches the table through gears with a small gap and a coupling that twists like a stiff spring. A sensor on the motor cannot see the gap or the twist, and a speed loop pushed too hard makes the motor and the table ring against each other.

Why it matters. A machine that cuts metal or places parts needs axes that stop where they are told.

THREE WAYS TO READ THIS CHAPTER

Explore	12 min	the first page, the history of speed set by voltage (Section 12.2), the fact on error motions (Section 12.3), the try-it segments of cases 3 and 6 and the fact on the hunting integral (Section 12.5) and the key points
Study	65 min	the whole chapter with its three pause points and the twelve exercises at its end
Practice	25 min	the first page, the results (Section 12.5), Technology in Practice (Section 12.6), Limitations (Section 12.7) and the open challenges

The control question A servo motor drives a machine table through a flexible coupling and a gear pair with a gap. What caps the gain of its speed loop, what can a motor encoder not see, and what does a relay autotuner measure on such an axis?

Learning objectives

- Compute the anti-resonance, resonance and locked-load frequencies of the two-mass axis from its inertias and stiffness.
- Compute the voltage a motor needs at speed and the base speed at which the inverter's voltage circle runs out.
- Read the gain and phase margins of a speed loop that crosses unity more than once, and use a notch filter to raise its usable gain.
- Explain why Ziegler–Nichols gains from a relay test tune a resonant axis to its coupling.
- Budget the final error of a position loop closed on the motor encoder, and compare feedforward, a load encoder and an explicit MPC on one move.

Prerequisites Sections 1.6.2 (velocity form), 1.4.7 (relay test), 1.4.9 (windup), 1.4.10 (transfer), 1.4.11 (sample time) and 1.4.12 (margins). Section 1.6.3 is needed only for the explicit MPC. Chapter 2 introduces the algorithm on a simpler plant.

The app https://apmonitor.com/apps/servo-axis/

See also Volume 3, Chapter 2 for cascade structure and Volume 3, Chapter 9 for an inner current loop and an explicit MPC. Volume 7, Chapter 12 for a lightly damped mode under a moving drive and Volume 7, Chapter 13 for the cutting forces. Volume 20, Chapter 10 for trajectory feedforward on an arm and Volume 20, Chapter 14 for joint servos and a cable-driven wrist.

12.1 Introduction

A machine-tool feed axis converts the torque of a motor into the travel of a table through a coupling, a screw and a nut, and the design, modeling and control of such feed drives have a review literature of their own [1]. Its drive closes a current loop inside a speed loop inside a position loop, the structure that direct-current drives established [2]. The motor is the well-behaved part of the chain. High-performance servo drives are often limited by mechanical load resonance instead [3], and in drives whose speed-loop bandwidth and resonance both lie above 100 Hz a flexible coupling narrows the range of stable speed gains [4].

On the axis of this chapter a 1 A current step at rest rises from 10 to 90 % in 0.30 ms, while the motor and a table of five times its inertia swing against each other on the coupling at 399.9 Hz. Three expectations that a block diagram of the drive encourages fail on it. A higher speed gain does not buy a stiffer axis until the motor's torque runs out, because the speed loop sings once its gain passes 0.55 A s/rad, at 413.8 Hz for a gain of 0.7, while its q-axis current swings only between −15.9 and 17.5 A, less than 56 % of the drive's 31.7 A peak. A position loop closed on a 20-bit motor encoder, 9.54 nm of table per count, does not place the table to that resolution. With 0.5° of backlash in the gears and a 300 N cut, the table of the graded case stops 56.6 counts of the 16-bit screw encoder (8.6 µm) from its target, 55.0 of them half the gap (45.5) and the twist of the coupling under the cut (9.5). An autotuner does not find the loop's limit and then tune to it. The relay experiment finds a limit, at 432.4 Hz, but the limit belongs to the coupling, and the Ziegler–Nichols gains it implies overshoot a 600 rpm step by 82.2 % against 29.7 % for the commissioned ones.

The axis is one member of a general class. A drive that measures and actuates one inertia while a lightly damped mode joins it to another meets a speed-loop limit set by the mode rather than by its actuator. The damping of a PI speed loop on a two-inertia system depends strongly on the ratio of the load's inertia to the motor's [5]. A prototype rolling-mill drive with three inertias on shafts of low stiffness, resonating at 17.4 and 51.3 Hz like an actual mill, reached a high speed bandwidth with an observer-based two-degree-of-freedom controller, compared in the same study with the conventional PI one [6], and a servo motor whose coupling resonates at 160 Hz needed a modified

speed controller to extend its range of stable gains [4]. The same loops enclose a gap in the gears, among the nonlinearities that limit speed and position control the most [7], and friction, which with integral action makes a position servo hunt [8]. A transfer function from current to speed with a rigid load carries none of these.

However, the methods for resonant loads were compared on physical drives, a 100 Hz resonance in one study [3] and 160 Hz in another [4], where the true position of the table, the twist of the shaft and the state of the gear teeth cannot be shown beside what the encoders report, and no move can be run twice with one setting changed. This chapter works with a simulated axis that can be tuned from the current loop outward, one step at a time, with every step judged against where the table truly is. The dq model, the two-mass model, the notch, the relay experiment and the profile feedforward follow the literature reviewed in Section 12.2. The axis can be rebuilt from this chapter alone. Its drive reads only its phase currents, its bus voltage and two encoders, runs its three loops at three rates in one controller, and draws the frequency response of its own speed loop from the numbers the simulation runs, within 1.6 % in magnitude and 0.3° in phase of a simulated sine sweep between 200 and 500 Hz. Nine cases and a graded move and return each isolate one step of commissioning, and one of them is honestly infeasible. A tenth case sets an explicit model predictive controller (MPC) in place of the position and speed loops, computed offline as the piecewise affine solution of a constrained linear quadratic regulator [9], and measures it against the commissioned cascade on the same move and cut, losses included.

The chapter is organized as follows. Section 12.2 traces how the servo axis acquired its cascade of current, speed and position loops. Section 12.3 states the model from the windings to the table: the motor in its rotor frame, the two masses joined by a spring and a gap, the friction, the thermal image, the encoders and the motor-side transfer function. Section 12.4 describes the drive as implemented, with its three loops, its filters, its profile and feedforward, its relay autotuner and inertia identification, the linear loop behind its Bode view and the explicit MPC. Section 12.5 reports the measurements case by case, from torque applied by hand to the graded move and return, and Section 12.5.10 compares the explicit MPC with the cascade. Section 12.6 sets the axis among the sensors, drives and controllers of practice, Section 12.7 states what the model leaves out, Section 12.8 summarizes the results, and the exercises follow in Section 12.9.

12.2 How the Servo Axis Acquired Its Cascade

Electric motion control began with the direct-current motor, and the history box below tells how, in 1896, its speed came to be set by voltage rather than by resistance [10]. Closing a loop around such a drive asked for a measurement of its speed as well as its position, and by 1952 a tachometer on the shaft supplied the rate signal of a high-performance servomechanism, with designs that traded signal level for low inertia, low noise and low friction [11]. Once static converters fed the armature, the direct-current drive was controlled as the cascade that this simulator keeps, an armature-current loop inside a speed loop inside a position loop, each faster than the loop around it [2]. Volume 3, Chapter 2 compares a cascade with a single loop and with feedforward on a tank level, where the same rule of a faster inner loop applies.

> ⧖ HISTORY **Speed by Voltage, 1896**
>
> On 18 November 1896 H. Ward Leonard read a paper before the American Institute of Electrical Engineers on controlling the speed of a motor by the voltage generated in its armature circuit rather than by a resistance in it. Elevators, printing presses, cranes and the turrets of warships all needed a motor held at any speed from rest to full. In his scheme an engine turned a generator at a practically constant speed, and a rheostat in the generator's field set the voltage fed to the motor. At one tenth of full speed under full torque, throwing off the whole load changed his example motor's speed by 9 revolutions per minute, against 415 under resistance control, nearly 50 times as much. Reversing the rheostat sent current back to the line as the motor braked [10]. The drive of this chapter likewise acts on its motor only through the voltages it applies.

Alternating-current machines did not fit that structure at first. Their dynamic interactions are far more complex than those of direct-current machines and some of the important quantities cannot be measured, and the difficulty was overcome about twenty years before 1991 by field orientation, which controls the machine in moving frames of reference tied to its flux and needs on-line signal processing that only microprocessors made economical [12]. In less than two decades field orientation went from research studies to the accepted method of controlling AC machines for high dynamic performance [12]. Vector control of the permanent-magnet synchronous motor had been modeled and simulated with its inverter and speed controller by 1989 [13], and in 1994 sinusoidal permanent-magnet machines were reviewed as offering extremely smooth torque and extended high-speed operating ranges [14]. Setting the d- and q-axis currents according to the rotor speed, within the voltage and current that the inverter can supply, had been shown in 1990 to expand the operating limits of such motors [15], and the field weakening of case 9 is a simple form of that idea.

With the motor under control, the mechanics became the limit. On a two-inertia system a conventional PI speed loop has a damping that depends strongly on the inertia ratio, and derivative feedback of the motor speed improves it when the ratio is small [5]. Rolling mills were the heavy-industry version of the problem, with resonances of 17.4 and 51.3 Hz on a laboratory mill built to the inertia ratios of an actual one [6], and servo drives met it again above 100 Hz [4]. A comparison of seven methods of resonant-load control on a lightly damped 100 Hz resonance evaluated a notch filter, a bi-quad filter, a low-pass filter, rigid-body observers, observer filtering, active resonance damping and center-of-mass control for their command response and dynamic stiffness [3]. The gap in a gear train and the friction of machine slides each acquired a survey of models, analysis tools and compensation [7, 8].

Automatic tuning arrived from the regulator side. A relay in place of the controller drives a loop into a limit cycle at the frequency where its phase reaches $-180°$, and in 1984 the amplitude and period of that cycle became the basis of tuning simple regulators to specifications on phase and amplitude margins [16]. The ultimate gain and period it measures are the inputs of the Ziegler–Nichols rule [17], which the application applies, as Section 1.4.7 describes for a process loop. Trajectory planning for point-to-point moves came to be designed together with feedforward, bounding the jerk so that feedback is left only what the model does not predict [18]. The application assembles these pieces on one axis, where each can be tried against a measurement that no physical drive offers, the table's true position.

12.3 The Axis From Windings to Table

The simulated axis is a permanent-magnet servo motor on a 48 V drive that turns a leadscrew of 10 mm lead through a flexible coupling and a 1:1 gear pair, which folds the motor beside the screw as on many machine-tool axes, as shown in Figure 12.1. Everything mechanical is referred to the motor shaft. Seven states are integrated, the d- and q-axis currents i_d and i_q, the angle θ_m and speed ω_m of the rotor, the angle θ_l and speed ω_l of the screw, and the drive's thermal image h of the winding. Table 12.1 sorts the signals by their roles in the three loops.

> **+ DID YOU KNOW? Eighteen Error Motions on Three Axes**
>
> A typical machine tool has three linear axes, each with six translational and angular error motions, 18 in all. They are major contributors to the error between the tool and the workpiece, and they grow as guideways and recirculating balls pit, wear and crack [19]. The axis of this chapter keeps one motion, the travel itself.

12.3.1 The Motor in Its Rotor Frame

The motor has surface magnets, so its d- and q-axis inductances are equal, and in the frame of the rotor its windings and torque obey Equation (12.1), in the form used for permanent-magnet

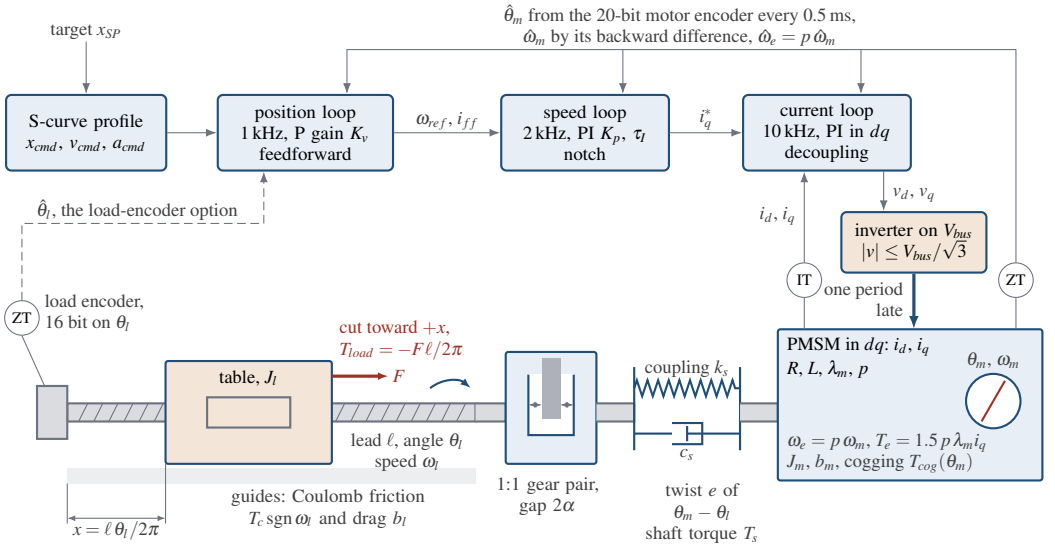


Figure 12.1 One servo axis and its drive, not to scale. The table, of inertia J_l referred to the motor, rides a leadscrew of lead ℓ at position x on guides with Coulomb friction T_c and drag b_l, and a cut F loads the screw with T_{load}. A 1:1 gear pair with a gap 2α and a coupling of stiffness k_s and damping c_s pass the shaft torque T_s of the engaged twist e to the motor, whose rotor of inertia J_m and drag b_m turns through θ_m at ω_m under the torque T_e of its q-axis current and the cogging T_{cog}. The drive reads the phase currents (IT), the 20-bit motor encoder and, as an option, the 16-bit load encoder (ZT), and closes the position, speed and current loops at 1, 2 and 10 kHz. The inverter applies v_d and v_q one 100 µs period late, inside the circle $V_{bus}/\sqrt{3}$.

Table 12.1 Signals of the Axis, Their Roles and Starting Values

Signal	Role	At start	Range
x	CV, position loop, the table	0	−100 to 300 mm
ω_m	CV, speed loop, a state	0	reference ±4500 rpm
i_d, i_q	CVs, current loop, states	0	limit 31.67 A
v_d, v_q	MVs, current loop	0	$\lvert v\rvert \le 27.71$ V
i_q^*	MV, speed loop	0	$\pm i_{lim}$
ω_{ref}	MV, position loop	0	±4500 rpm
θ_m, θ_l, ω_l	states, rotor and screw	0	—
h	state, I^2t image	0	1 folds back
F	DV, cut on the table	0	±1000 N
F_r, f_r	DV, ripple of the cut	0, 163 Hz	0–500 N, 10–1000 Hz
—	DV, workpiece inertia change	0	−80 to 300 %
—	DV, bus sag, friction rise	0, 0	0–50, 0–300 %
—	DV, encoder glitch	—	4096 counts once
$\hat\theta_m$, $\hat\theta_l$	measured, encoders	20, 16 bit	every 0.5 ms
i_d, i_q, V_{bus}	measured, sensors	ideal	every 0.1 ms

synchronous motor drives [13].

$$\begin{aligned} L\dot i_d &= v_d - R\,i_d + \omega_e L\,i_q \\ L\dot i_q &= v_q - R\,i_q - \omega_e\,(L\,i_d + \lambda_m), \qquad \omega_e = p\,\omega_m \\ T_e &= \tfrac{3}{2}\,p\,\lambda_m\,i_q = K_t\,i_q \end{aligned} \tag{12.1}$$

where the transform is amplitude invariant, so i_q is the peak phase current that makes torque, R and L are the resistance and inductance of a phase, λ_m the magnets' flux linkage and $p = 4$ the pole pairs. With $\lambda_m = 0.02\,\text{Wb}$ the torque constant is $K_t = 0.12\,\text{N m/A}$, the rated 1.27 N m takes 10.58 A and the peak 3.8 N m takes 31.67 A, and the back-EMF constant $p\lambda_m$ is 0.08 V s/rad. The inverter is averaged over each 100 µs PWM period and applies the voltages the current loop computed one period earlier, which the controller keeps inside the circle of Equation (12.2).

$$\sqrt{v_d^2 + v_q^2} \le V_{max} = V_{bus}/\sqrt{3} \tag{12.2}$$

The 48 V bus gives $V_{max} = 27.71\,\text{V}$. In steady state with $i_d = 0$ the voltage needed is

$$\sqrt{(Ri_q + \omega_e \lambda_m)^2 + (\omega_e L i_q)^2},$$

so without field weakening the motor runs out of voltage at 3308 rpm with no load, at 2419 rpm with its rated current and at 1008 rpm with its peak, and its nominal rating of 1.27 N m at 3000 rpm would need 33.20 V.

12.3.2 Two Masses, a Spring and a Gap

The rotor and the load, referred to the motor, obey Equation (12.3), with $\dot\theta_m = \omega_m$ and $\dot\theta_l = \omega_l$.

$$\begin{aligned} J_m\,\dot\omega_m &= T_e - T_{cog}(\theta_m) - b_m\,\omega_m - T_s \\ J_l\,\dot\omega_l &= T_s - b_l\,\omega_l - T_c\,\operatorname{sgn}\omega_l - T_{load} \\ T_{cog} &= A_{cog}\sin(n_{cog}\theta_m), \qquad T_{load} = -F\ell/2\pi \end{aligned} \tag{12.3}$$

where A_{cog} is 2 % of the rated torque, 0.0254 N m, at $n_{cog} = 24$ cycles per revolution, the cut F is positive toward $+x$, and the table stands at $x = \ell\theta_l/2\pi$. The compliances of the coupling, the gear mesh and the screw are lumped into one spring and one damper in series with the gap of the gears, which is modeled as a dead zone in the twist, as shown in Equation (12.4).

$$\begin{aligned} e &= \begin{cases} \Delta\theta - \alpha\operatorname{sgn}\Delta\theta, & |\Delta\theta| > \alpha \\ 0, & |\Delta\theta| \le \alpha \end{cases}, \qquad \Delta\theta = \theta_m - \theta_l \\ T_s &= k_s\,e + c_s\,(\omega_m - \omega_l)\min\bigl(1,\ |e|/e_r\bigr), \qquad T_s\,e \ge 0 \end{aligned} \tag{12.4}$$

where 2α is the whole gap and $e_r = 20\,\mu\text{rad}$ fades the damping in as the teeth meet, so the torque is continuous at engagement. The last condition sets T_s to zero whenever the damper would make a separating flank pull, since teeth push and never pull. With no gap the shaft is the plain spring and damper, and the default gear pair is preloaded. At the rated torque the spring of $k_s = 526\,\text{N m/rad}$ twists by 2.414 mrad, 3.84 µm of table travel. A dead zone is the simplest of the backlash models in use for control [7].

12.3.3 Friction, Heat and the Encoders

The guides' Coulomb friction, $T_c = 0.08\,\text{N m}$, is handled as a hybrid. The sliding direction is latched for each integration step so that the right-hand side is smooth inside it, and after the step the screw is stopped where its speed crosses zero and held still while the shaft and the cut pull on it by no more than T_c, $|T_s - T_{load}| \le T_c$. There is no static friction above the Coulomb level and no Stribeck dip. The drive protects the winding with the thermal image of Equation (12.5).

$$\tau_{th}\,\dot h = (i_d^2 + i_q^2)/I_{rated}^2 - h \tag{12.5}$$

where $\tau_{th} = 10\,\text{s}$. When h reaches 1 the current limit folds back from the peak to the rated current until h has fallen to 0.9, and at 31.6 A with the screw held the image trips after 1.191 s. Every 0.5 ms the drive samples both encoders as $\hat\theta = q\,\text{round}(\theta/q)$, with $q_m = 2\pi/2^{20}$ on the rotor, 9.54 nm of table per count, and $q_l = 2\pi/2^{16}$ on the screw, 0.1526 µm per count, so that a 1° gap is 182.0 load counts. The drive never reads ω_m, ω_l, e or x themselves.

Table 12.2 Constants of the Model. V Marks a Show Model Slider, H a Hidden Parameter, A an Advanced Control and C a Value Fixed in the Code

Symbol	Meaning	Value	Class
Motor and drive			
R	phase resistance	0.5 Ω	V
L	phase inductance, $L_d = L_q$	1.0 mH	V
λ_m	magnet flux linkage	0.020 Wb	V
p	pole pairs	4	H
J_m	rotor inertia	1.0 kg cm^2	V
b_m	bearing drag	1.0×10^{-4} N m s/rad	H
T_{rated}	continuous torque, the I^{2}t base	1.27 N m	V
T_{peak}	peak torque, the current limit	3.8 N m	V
V_{bus}	bus voltage	48 V	V
τ_{th}	time constant of the thermal image	10 s	V
Mechanics, referred to the motor			
J_l	table, screw and workpiece	5.0 kg cm^2	A
k_s	coupling and mesh stiffness	526 N m/rad	A
c_s	coupling and mesh damping	0.020 N m s/rad	V
2α	gear backlash, the whole gap	0, to 2°	A
b_l	screw and guide drag	2.0×10^{-4} N m s/rad	H
T_c	guide Coulomb friction	0.080 N m	V
ℓ	leadscrew lead	10 mm per rev	H
A_{cog}	cogging amplitude	2 % of rated	A
n_{cog}	cogging cycles	24 per rev	H
e_r	damping fade-in at contact	20 µrad	C
Instruments and detectors			
N_m, N_l	motor and load encoders	20, 16 bit per rev	H
—	singing detector, band and Q	420 Hz, 1	C
—	detector on and off, 10 ms RMS	0.30, 0.15 N m	C
—	encoder glitch	4096 motor counts	C

12.3.4 The Motor-Side Transfer Function

Without the gap, the friction and the drags, the response of the motor speed to the motor torque is the two-inertia transfer function of Equation (12.6) [5].

$$G_m(s) = \frac{\omega_m}{T_e} = \frac{J_l s^2 + c_s s + k_s}{s\left[J_m J_l s^2 + (J_m + J_l)(c_s s + k_s)\right]} \tag{12.6}$$

Its zeros lie at the anti-resonance $f_{ar} = \sqrt{k_s/J_l}/2\pi = 163.2$ Hz, where the load rings against a motor that stands still, and its poles at the resonance $f_r = \sqrt{k_s(1/J_m + 1/J_l)}/2\pi = 399.9$ Hz, where motor and load swing against each other with a damping ratio of $c_s/2\sqrt{k_s J_m J_l/(J_m + J_l)} = 0.048$. Against the rigid-body line $1/(J_m + J_l)s$ the damped response dips to 0.047 at 163.15 Hz and peaks at 52.3 times at 401.15 Hz. With the load held still the motor alone would ring at $\sqrt{k_s/J_m}/2\pi = 365.0$ Hz, the locked-load frequency. Above the resonance the motor feels only its own inertia, so the gain of the loop there is $(J_m + J_l)/J_m = 6$ times what a rigid axis would give.

12.3.5 Starting Point and Numerics

The constants are those of Table 12.2. Every case starts at rest at $x = 0$ with no current, the teeth centered in their gap, the screw held by its guides and a cold winding, $h = 0$. The seven states advance by the framework's fourth-order Runge–Kutta method (Section 1.6.1) on a fixed 100 µs step, 25 steps a period of the resonance, and the friction latch, the scorecard and the singing detector act after each step, never inside the derivative function. The singing detector passes the shaft torque through two band-pass sections at 420 Hz with $Q = 1$ and reports RESONANCE while their RMS over about 10 ms exceeds 0.30 N m, until it falls below 0.15 N m. The historian logs at 2 kHz, so that the resonance shows in the trends as more than a smear. Halving the physics step, with the loops kept at their rates, moves the frequency at which case 3 sings from 413.733 to 413.708 Hz and

the position at which case 6 settles by 0.001 of a load count, and the graded case run twice gives bit-identical states. The model has no random terms, so quantization is its only noise and every case replays exactly, which is why the exercises of this chapter need no seed. All 57 of the application's verification checks pass.

12.3.6 Assumptions

One torsional mode. The coupling, the mesh and the screw are one spring, so the axis has one resonance where a machine has several, among them the screw's axial mode, its bearings and the frame, and the next one up often limits a notched loop.

An averaged inverter. The voltages arrive one period late and clipped to the circle, with no switching ripple, dead time or diode drops, so above the no-load speed without field weakening the current simply follows the back-EMF.

An ideal surface-magnet motor. The inductances are equal and constant, the back-EMF is sinusoidal, iron saturates nowhere and loses nothing, and the thermal image stands in for a thermal model.

Coulomb friction only. The guides hold the table whenever it is pulled by less than T_c, with no static peak above it, no Stribeck dip and no pre-sliding spring, so the last micrometers of a move are a staircase of sticks and slips rather than the creep of a real guide.

Perfect quantizers. The encoders quantize and do nothing else, the current sensors are exact, and nothing is noisy, whereas real encoders carry interpolation error and a linear scale grows with temperature.

Teaching values. The motor, the mechanics and every threshold, the 20 mm following-error window, the 0.30 N m singing threshold, the two counts and the 6 dB of the graded case, are teaching values rather than a drive maker's data, and a real axis takes them from its drive's commissioning guide and its machine's accuracy class.

12.4 The Drive as Implemented

The drive reads its phase currents and its bus voltage every 100 µs and its two encoders every 0.5 ms, and nothing else. One controller runs every physics step. It first hands the inverter the voltages computed on the step before, then on every fifth step samples the encoders, on every tenth step runs the position loop, on every fifth the speed loop, and on every step the current loop, which computes the voltages for the next period. Table 12.3 gives each setting with its default and range.

12.4.1 One Shared Law at Three Rates

All three loops run the velocity-form PID of the framework, which Section 1.6.2 gives in general, with the clamp on the emitted output. With no derivative action and no bias it reduces to Equation (12.7).

$$\begin{aligned} p_k &= K_c\,(r_k - y_k), \qquad S_k = S_{k-1} + (p_k - p_{k-1}) \\ \hat{u}_k &= S_k + I_{k-1}, \qquad u_k = \mathrm{clamp}(\hat{u}_k,\ u_{min},\ u_{max}) \\ I_k &= I_{k-1} + \left[(K_c/\tau_I)(r_k - y_k) + (u_k - \hat{u}_k)/\tau_I\right] T \end{aligned} \tag{12.7}$$

where T is the loop's own period. In the notation of Equations (1.23) to (1.26), p_k is the proportional term P_k and $\hat{u}_k$ is the output before limiting v_k. The running sum S_k is never clamped, so a change of gain reaches the whole proportional term at once. Only the integral is wound back against what was applied (Section 1.4.9), with a tracking time equal to τ_I, and a loop without integral action, such as the default position loop, back-calculates nothing. Each loop has its own manual station, the q-axis voltage for the current loop, the current command for the speed loop and the speed reference, the jog, for the position loop. A loop in manual tracks its station by solving Equation (12.7) for I, every loop outside it tracks too because it cannot act through an inner loop that is not listening, and the header's MANUAL and AUTO switch throws the speed and position loops together, copying their outputs

Table 12.3 Drive Settings, Defaults and Ranges

Symbol	Setting	Default	Range
Position loop, 1 kHz			
—	position and speed control	cascade	or MPC
—	MPC evaluation, preview	lookup, on	online, off
K_v	position gain	30 s^{-1}	5–200
$\tau_{I,x}$	position integral time	off	0–500 ms
—	position feedback	motor encoder	or load
ϕ	profile feedforward	off	on, off
$\hat{J}$	inertia it assumes	6.0 kg cm^2	0.5–25
$\hat{T}_c$	friction it assumes	0.080 N m	0–0.4
e_{max}	following-error window	20 mm	0.5–50
$2\hat{\alpha}$	backlash compensation, its gap	off, 0	0–2°
—	dwell between queued moves	50 ms	0–500
v_{max}	profile speed	300 mm/s	10–700
a_{max}	profile acceleration	5 m/s^2	0.5–10
j_{max}	profile jerk	200 m/s^3	20–2000
Speed loop, 2 kHz			
K_p	speed gain	0.30 A s/rad	0.02–3
τ_I	speed integral time	20 ms	1–100
—	notch filter	off	on, off
f_n	notch center	400 Hz	100–900
Δf	notch width	200 Hz	20–400
d	notch depth	−30 dB	−40 to −3
τ_f	current-command low-pass	off	0–2 ms
—	speed feedback	motor encoder	or load
d_r	relay amplitude	2 A	0.2–10
Current loop, 10 kHz			
$K_{p,i}$	current gain	3 V/A	0.5–10
$\tau_{I,i}$	current integral time	2 ms	0.2–10
—	field weakening	off	on, off

into their stations once the run has started, so every transfer is bumpless (Section 1.4.10), and AUTO returns them to the case's own arrangement.

12.4.2 The Current Loop and the Voltage Circle

The current loop runs a PI on each axis with the cross-coupling and the back-EMF fed forward from the measured currents and the encoder speed, as shown in Equation (12.8).

$$\begin{aligned} v_d &= \mathrm{PI}_d(i_d^* - i_d) - \hat{\omega}_e L i_q \\ v_q &= \mathrm{PI}_q(i_q^* - i_q) + \hat{\omega}_e (L i_d + \lambda_m) \end{aligned} \tag{12.8}$$

with $|v_d| \le V_{max}$ and $|v_q| \le \sqrt{V_{max}^2 - v_d^2}$, where $\hat{\omega}_e = p\,\hat{\omega}_m$ and each PI part is Equation (12.7) with $K_c = 3\,\mathrm{V/A}$ and $\tau_I = L/R = 2\,\mathrm{ms}$, a pole-zero cancellation, its clamp shifted by the feedforward so that the total voltage stays inside the circle. The d axis is served first and may take the whole circle, and each PI back-calculates against what was applied. The linear analysis puts the small-signal bandwidth of the loop at 926 Hz, and a 1 A step at rest rises from 10 to 90 % in 0.30 ms. With field weakening on, the d-axis reference is the root nearest zero of Equation (12.9) that is not positive.

$$\begin{aligned} &\left(R i_q + \hat{\omega}_e (L i_d^* + \lambda_m)\right)^2 \\ &\quad + \left(R i_d^* - \hat{\omega}_e L i_q\right)^2 = (0.95\, V_{max})^2 \end{aligned} \tag{12.9}$$

Below the base speed $i_d^* = 0$, the reference is bounded by $-0.8\,I_{peak} = -25.3\,\mathrm{A}$, and where no i_d^* reaches the circle the one that comes closest is used. At the nominal 3000 rpm with the rated current it asks for −8.65 A. Without field weakening $i_d^* = 0$.

12.4.3 The Speed Loop, Its Filters and Its Limit

The speed is the backward difference of the motor encoder over one speed period, $\hat{\omega}_m = (\hat{\theta}_{m,k} - \hat{\theta}_{m,k-1})/T_\omega$ with $T_\omega = 0.5\,\text{ms}$, an average over the last period. The PI of Equation (12.7), $K_p = 0.30\,\text{A s/rad}$ and $\tau_I = 20\,\text{ms}$, acts on $\omega_{ref} - \hat{\omega}_m$, and the torque feedforward i_{ff} of the position loop is added to its output inside the limit $\pm i_{lim}$, $i_{lim} = \sqrt{I_{max}^2 - i_d^{*2}}$, where I_{max} is the peak current, or the rated current after an I^2t trip. The command then passes a first-order low-pass, exact for a held input and off by default, and the notch of Equation (12.10), and is clamped to $\pm i_{lim}$ again. The integral is wound back against the PI's own clamp, before the filters, so a filtered command that only this second clamp cuts is not fed back to it.

$$N(s) = \frac{s^2 + 2\zeta_z\omega_0 s + \omega_0^2}{s^2 + 2\zeta_p\omega_0 s + \omega_0^2} \tag{12.10}$$

where $\omega_0 = 2\pi f_n$, $\zeta_p = \Delta f/2f_n$ sets the width and $\zeta_z/\zeta_p = 10^{d/20}$ the depth at the center. The notch is discretized by the bilinear transform prewarped at its center, $s \to K(1-z^{-1})/(1+z^{-1})$ with $K = \omega_0/\tan(\omega_0 T_\omega/2)$, so that it sits exactly where it is asked to, and it runs at 2 kHz in transposed direct form II. It filters the command at all times, so switching it on in a running axis uses a filter that is already settled. The relay autotuner can stand in for the PI, and the inertia identification can add a chirp to ω_{ref}.

12.4.4 The Position Loop, the Profile and Feedforward

A move is a rest-to-rest S-curve of seven segments whose jerk is $\pm j_{max}$, solved exactly before it starts and lowered in its peaks when the distance is too short to reach them, and evaluated in closed form, so that x_{cmd} lands on the target to the last bit and never passes it. A target asked for during a move waits for that move to end and for the dwell, 50 ms by default, after it. A 100 mm move at the defaults takes 418.0 ms. The position loop gives the speed reference and the torque feedforward of Equation (12.11).

$$\begin{aligned} \omega_{ref} &= K_v\,\frac{x_{cmd} - \hat{x}}{\ell/2\pi} + \phi\,\frac{v_{cmd}}{\ell/2\pi} \\ i_{ff} &= \phi\,\frac{\hat{J}\,a_{cmd}\,2\pi/\ell + \hat{T}_c\,\operatorname{sgn} v_{cmd}}{K_t} \end{aligned} \tag{12.11}$$

where $\hat{x}$ is the motor encoder's position or the load encoder's, ϕ is the feedforward switch, and ω_{ref} is limited to $\pm 4500\,\text{rpm}$. Without feedforward a proportional loop needs a following error of v/K_v to move at the speed v, 10 mm at 300 mm/s and $30\,\text{s}^{-1}$. If $|x_{cmd} - \hat{x}|$ ever exceeds the window e_{max}, the drive faults and switches its torque off until Reset. Backlash compensation, for the motor-encoder loop only, moves the motor's aim by $\hat{\alpha}$ toward the last commanded direction at 20 rad/s, so that each reversal is a quick step of the assumed gap, as a numerical control's backlash compensation does.

12.4.5 The Drive's Own Tools

The relay autotuner replaces the speed PI by the relay of Equation (12.12), about the current that was holding the axis.

$$\begin{aligned} i_{cmd} &= i_0 \pm d_r, & K_u &= |U_1|/|E_1| \\ K_p &= 0.45\,K_u, & \tau_I &= T_u/1.2 \end{aligned} \tag{12.12}$$

where the sign follows the speed error with a hysteresis of two encoder counts of speed, 0.024 rad/s, the period T_u is the mean of eight cycles counted after four are skipped, and U_1 and E_1 are the fundamentals at $1/T_u$ of the relay's output and of the speed error over the counted cycles. The classic estimate $4d_r/\pi a$, Equation (1.16) with no switching band, reads the oscillation's peaks a, but at 2 kHz a cycle near 430 Hz has four or five samples, and the sampled peaks make it 9.4 % high in case 7 (0.701 against 0.641 A s/rad). The gains are then set by the Ziegler–Nichols PI rule [17] of Table 1.2.

The inertia identification sweeps the speed reference with a chirp from 3 to 30 Hz of $\pm 25\,\mathrm{rad/s}$ for 0.5 s, filters the commanded torque $K_t i_q^*$ and the encoder speed through the same 60 Hz low-pass, and fits $T = J\dot{\omega} + b\,\omega + T_c \operatorname{sgn}\omega$ by least squares, $\dot{\omega}$ being the difference of the filtered speed. The fitted J is the rigid-body inertia and goes to $\hat{J}$. The friction and drag fitted beside it do not, because the sweep cannot separate them (Section 12.5).

12.4.6 The Loop the Bode View Draws

The application computes the open speed loop, broken at the PI, from the same parameters the simulation runs, as shown in Equation (12.13).

$$L(j\omega) = C(z)\,F_{lp}(z)\,N(z)\,H(j\omega)^2\,T_i(j\omega)\,K_t\,\tilde{G}_m(j\omega) \tag{12.13}$$

where $z = e^{j\omega T_\omega}$, $H(j\omega) = (1 - e^{-j\omega T_\omega})/j\omega T_\omega$, $C(z) = K_p + (K_p/\tau_I)T_\omega z^{-1}/(1 - z^{-1})$ is the PI as Equation (12.7) runs it, H appears twice for the held current command and the averaged speed estimate, T_i is the closed current loop at 10 kHz with its period of delay, and $\tilde{G}_m$ is the plant of Equation (12.6) with the drags b_m and b_l and with the damping the back-EMF adds through the current loop, since the back-EMF is fed forward from the 2 kHz speed estimate and does not cancel at the mode. The margins are those of Equation (1.19), read with care because the loop can cross unity more than once. The crossover reported is the first frequency where $|L(j\omega)| = 1$, the phase margin is the worst over every crossover, and stability is judged by counting the crossings of the negative real axis to the left of -1. The gain margin is the ratio of the ultimate gain of the proportional loop with the filters to K_p, which differs from the margin of the PI loop by no more than 0.11 dB in any configuration quoted in this chapter. Each hold acts much as the added dead time of Section 1.4.11. At 400 Hz the hold, the speed estimate, the current loop and the PWM delay together cost 116.9° of phase, against 86.4° at 300 Hz. Against a simulated sine sweep of the plant at 300 rpm, the model's magnitude agrees within 1.56 % and its phase within 0.29° from 200 to 500 Hz and within 1.0 % from 20 to 150 Hz, but at the anti-resonance the simulated axis responds seven times more than the model's deep zero, and the error grows to 4.0 % at 600 Hz and 9.3 % at 700 Hz, because aliasing near the 1 kHz Nyquist frequency is left out, and its stability limit for case 3, a gain of 0.592, lies between the 0.55 at which the simulated axis is quiet and the 0.60 at which it sings.

12.4.7 Three Models

The simulation model is Equations (12.1) to (12.5). Equation (12.13) is the linear model behind the Bode view and the margins, derived from the parameters rather than identified, and blind to the gap, the Coulomb friction, the cogging, the quantization and aliasing. The controller's internal models are the decoupling of Equation (12.8), with the true L, λ_m and p, the feedforward's $\hat{J}$ and $\hat{T}_c$, the notch's center, which is 400 Hz against the plant's 401.15 Hz, and the compensation's assumed gap. Two models are identified while the axis runs, the relay's ultimate point and the chirp's inertia, and the comparison of each with the linear model is part of what the cases teach. The explicit MPC of Section 12.4.8 adds one more internal model, the two-mass axis at the declared machine.

12.4.8 Explicit Model Predictive Control

PI cascades, state-space control and model-based predictive control have been compared on drives with elastically coupled loads [20], and the application offers the predictive alternative as a second structure for the two outer loops, beside the cascade, never as the default and selected only in case 10. Section 1.6.3 describes the MPC option of the apps in general. This one differs in that its law is computed offline and looked up, and in that it replaces two loops of a cascade at once. Volume 3, Chapter 9 stores an explicit MPC as a lookup table in the same way, beside a converter cascade with an inner current loop. Wherever the position loop would act, with every loop in automatic, no fault and no relay test or chirp under way, one decision on each tick of the position loop sets the torque command u for the next millisecond, and the speed PI, its low-pass and its notch are bypassed. The current loop with its voltage circle and field weakening stays underneath, and the current limit with

its I^2t foldback and the following-error window stay above. With the position loop in manual, as in cases 1 to 4 and 7, the speed loop runs as shipped under either structure.

The controller's model is the two-mass axis of Equation (12.3) without the gap, the cogging and the Coulomb term, written at the motor shaft in the states $x = (\theta_l, \omega_l, \varepsilon, \dot{\varepsilon}, \tau)$, where $\varepsilon = \theta_m - \theta_l$ is the twist and τ the torque that the closed current loop delivers, as shown in Equation (12.14).

$$\begin{aligned} J_l\,\dot{\omega}_l &= k_s\varepsilon + c_s\dot{\varepsilon} - b_l\,\omega_l - d \\ J_m(\dot{\omega}_l + \ddot{\varepsilon}) &= \tau - k_s\varepsilon - c_s\dot{\varepsilon} - b_m(\omega_l + \dot{\varepsilon}) \\ \tau_c\,\dot{\tau} &= u - \tau \end{aligned} \tag{12.14}$$

where d is the load torque on the screw, the cut and the guides' friction together. Each constant takes the declared value of Table 12.2 ($J_m = 1.0\,\mathrm{kg\,cm^2}$, $c_s = 0.020\,\mathrm{N\,m\,s/rad}$, b_m and b_l) or the declared machine ($J_l = 5.0\,\mathrm{kg\,cm^2}$, $k_s = 526\,\mathrm{N\,m/rad}$), frozen when the application loads, so a machine or motor that a reader changes is a mismatch for the controller and never a new model. The lag $\tau_c = 0.25\,\mathrm{ms}$ stands in for the closed current loop. It was fitted to that loop's response to a 3 A step at rest, whose mean over the first millisecond is 0.756 of the step. A delay of 0.15 ms followed by a lag of 0.154 ms fits the step best, the single lag with the same mean over the ten samples is 0.30 ms, and at 1 ms the adopted lag predicts 0.982 of the step against a measured 1.000. The model keeps the resonance at 399.9 Hz and the anti-resonance at 163.2 Hz. An exact zero-order hold over $T_s = 1\,\mathrm{ms}$ gives $x_{k+1} = \Phi x_k + \Gamma u_k + E\,d_k$. Torque becomes current through the K_t that the shipped feedforward of Equation (12.11) uses, so a changed flux linkage moves the torque conversion of both structures alike.

A decision plans four torque moves of one millisecond each and applies only the first, as Equation (12.15) states.

$$\begin{aligned} &\min_{u_0,\ldots,u_3} \sum_{k=1}^{3} \left[q_x\,\tilde{\theta}_{l,k}^2 + q_\varepsilon\,\tilde{\varepsilon}_k^2 + q_\tau\,\tilde{\tau}_k^2 \right] + \tilde{x}_4^T P \tilde{x}_4 + r \sum_{j=0}^{3} u_j^2 \\ &\text{subject to } x_{k+1} = \Phi x_k + \Gamma u_k + E\,d_k, \quad x_0 = \hat{x} \\ &\qquad -U \le u_j \le U, \qquad U = K_t\,i_{lim} \end{aligned} \tag{12.15}$$

where $\tilde{x}_k = x_k - x_k^*$ is the predicted error against the reference state k periods ahead, $\hat{x}$ is the estimate, and P solves the discrete algebraic Riccati equation for Φ, Γ, $\mathrm{diag}(q_x, 0, q_\varepsilon, 0, q_\tau)$ and r, the cost-to-go that closes the horizon in the usual stabilizing construction [21]. Bryson's rule on 10 µm of table (6.283 mrad of screw), on the twist that holds the rated 1.27 N m and on that torque itself gives $q_x = 2.533 \times 10^4\,\mathrm{rad^{-2}}$, $q_\varepsilon = 1.715 \times 10^5\,\mathrm{rad^{-2}}$ and $q_\tau = 0.620\,(\mathrm{N\,m})^{-2}$, and the command is priced at 10^{-4} of the torque, $r = 6.2 \times 10^{-5}$ in the same unit. With no bound active the first move is the LQR's, $u = -K\tilde{x}_0$, with gains of 81.59 N m/rad on the table's angle, 0.3108 N m s/rad on its speed, −134.1 N m/rad on the twist, 0.055 54 N m s/rad on its rate and 0.2009 on the torque, which closes the rigid-body motion at 80.9 Hz with a damping ratio of 0.712, damps the coupling's mode to 0.200 at 461.8 Hz against its passive 0.048, and leaves the lag's pole at 2130 Hz. The weights were fixed before any closed-loop run.

The reference comes from the move profile evaluated k periods ahead, as shown in Equation (12.16), with $\bar{v}$, $\bar{a}$ and $\bar{j}$ the profile's speed, acceleration and jerk at the screw in radians.

$$\begin{aligned} \theta_{l,k}^* &= 2\pi x_{cmd}/\ell, \qquad \omega_{l,k}^* = \bar{v}, \qquad \dot{\varepsilon}_k^* = (J_l\bar{j} + b_l\bar{a})/k_s \\ \varepsilon_k^* &= (J_l\bar{a} + b_l\bar{v} + d_k - c_s\dot{\varepsilon}_k^*)/k_s \\ \tau_k^* &= (J_m + J_l)\bar{a} + (b_m + b_l)\bar{v} + d_k \end{aligned} \tag{12.16}$$

where $d_k = \hat{d} + T_c\,\mathrm{sgn}\,\bar{v}$ adds the declared friction in the profile's direction to the estimated load torque $\hat{d}$, which is the target-shift form of offset-free predictive control [22]. The plan therefore puts

in the twist that an acceleration needs before the table can follow. The preview reaches the move in progress 1 to 4 ms ahead, while the cascade's feedforward reads the present profile, and a queued move enters neither before it starts. With the preview off the present command is held with no motion or friction ahead, the knowledge of the cascade without feedforward.

Only the torque box constrains the plan, and its rows are hard. Its bound U is the current limit as the speed loop clamps it (the peak current, folded back to the rated one by the I^2t image, less the share of the d axis) converted by K_t, and it enters the problem as a parameter. There is no speed row, because the profile bounds the speed, and no voltage row, because the circle stays with the current loop. Collecting the estimate, the four references, the four load torques, U and a constant into a parameter ϑ of 31 entries turns Equation (12.15) into $\min \frac{1}{2} z^T G z + z^T F \vartheta$ over the moves z subject to $C_q z \geq b$, and an active set $\mathscr{A}$ of rows gives the law of Equation (12.17).

$$\begin{aligned} z_u &= -G^{-1} F \vartheta, \qquad & W_{\mathscr{A}} &= \left(C_{\mathscr{A}} G^{-1} C_{\mathscr{A}}^T \right)^{-1} \\ \lambda_{\mathscr{A}} &= W_{\mathscr{A}} \left(b_{\mathscr{A}} - C_{\mathscr{A}} z_u \right), \qquad & z &= z_u + G^{-1} C_{\mathscr{A}}^T \lambda_{\mathscr{A}} \end{aligned} \tag{12.17}$$

The law is optimal wherever $\lambda_{\mathscr{A}} \geq 0$ and every row holds. With four moves in a box each move is free, at its floor or at its ceiling, so there are $3^4 = 81$ candidate sets, and because $G^{-1}F$ has rank 4 the unconstrained moves cover all of $\mathbb{R}^4$ and every one of the 81 sets is optimal on a region with an interior. The table of 81 regions is thus the complete partition, nothing is truncated, and a finite parameter always finds its region. The search scales z_u by U and takes its tolerances relative to $\max(1, \|z_u\|_\infty / U)$, because a large following error makes z_u thousands of newton meters, where fixed tolerances miss. In 46,909 decisions over 56 closed-loop runs, every case of the application, the worst cases and the comparison of Section 12.5.10, 30 of the 81 regions occur, and 38 once 5,085 states at region boundaries and 20,000 states sampled from the physical range are added. Against the online solver that the apps share, on the same program, the lookup's moves differ by at most 2.5×10^{-13} of $\max(1, U)$ in closed loop and 7.9×10^{-11} at the boundaries, with no miss. In Node.js a lookup takes a median of 0.18 µs and a 99th percentile of 2.06 µs, against 13.4 and 27.2 µs for the online solve. Played in Chrome at real time, the top speed of 1000 decisions per second, the whole controller takes 1.06 and 0.76 % of one core in cases 10 and 6 under the lookup, 3.23 and 3.27 % under the online solve and 0.66 and 0.84 % under the cascade, at 60 frames per second, while other computations loaded the same host. The online solve is offered as a switch beside the table and gives the same moves.

A steady Kalman filter at the encoders' 2 kHz estimates x and $\hat{d}$ from the motor encoder, the measured q-axis current as torque, and the load encoder when a loop reads it, with a seventh state then for the load encoder's slow offset from the model's load angle (the gap or the screw's own wind-up), so that the fast motion is estimated through the collocated motor encoder. It predicts with the declared model from the torque commanded over the last 0.5 ms, assuming the quantization over $\sqrt{12}$ and 0.01 N m on the torque as measurement noise, and the cogging amplitude on the rotor, 0.02 N m on the load, 0.005 N m per read on $\hat{d}$ and 2 µrad per read on the offset as process noise. The offset state was added after the first results, in which the six-state filter left the challenge's program without its gap never in position, 2.1 counts off at 1.6 s, where the seven-state filter has it in position at 0.907 s. A motor-encoder read more than 10 mrad from the prediction is set aside once, as the glitch is, and the two reads after it are always taken, since setting every other read aside makes the error of the motor-encoder filter grow by 1.039 a read where two taken reads make it shrink by 0.879. Two large reads in a row reject the next decision as a model mismatch. The filter runs under both structures and writes nothing the cascade reads.

A decision is accepted only if a region is found, or the online solve is optimal with a plan of the right shape, and every planned move is finite and inside $\pm U$. Otherwise the cascade, whose speed PI has tracked the applied current and whose proportional position loop has tracked its own output, takes the torque without a bump, the historian marks the fallback, and the MPC takes it back over 20 ms after 20 acceptable decisions. A trip forced in the middle of case 10's move steps the current

command by 0.332 A, where the largest step the MPC itself makes in that window is 0.855 A, and a trip inside the blend back by 0.000 A.

A supervisor sits above the decision, because the model has no gap. While the teeth are apart the load is disconnected, and a law that prices the twist acts on a spring that is no longer there. With k_s and c_s set to zero the gain K as state feedback has a spectral radius of 2.08 per period, against 0.70 on the declared shaft. The supervisor rejects decisions, and the cascade holds the axis, while the estimate disagrees with an encoder between decisions by more than 5.6 mrad on the motor or 3.7 mrad on the load, while the estimated twist exceeds $2U/k_s$, while the estimated motor speed exceeds the cascade's 4500 rpm, or while the following error exceeds a quarter of the window e_{max}. The gates were sized before the worst cases were measured, from sixteen runs with no gap and no voltage loss, as twice the largest innovations (2.80 and 1.86 mrad), against the largest twist of 1.18 U/k_s, the largest speed of 1981 rpm and the largest following error of 2.6 % of the window. After a trip the MPC decides again only after 50 decisions back inside the gates, then the re-entry and its blend, and a second trip leaves the axis to the cascade until Reset. Without that lockout a machine the model does not fit cycles between the two structures. On case 6's move with the workpiece 80 % lighter the axis changes hands 15 times, at intervals of about 80 ms, and ends 5497 counts from its target, against 1.5 counts with the lockout. The supervisor was added after the first build ended the challenge as shipped in a following-error fault at 0.639 s with the motor at 3426 rpm. A reconstruction of that build shows the fault arising while the cascade held the axis after model-mismatch hand-overs, driven by a position-loop bias left from tracking the MPC and a diverging estimate, both since removed, rather than under the MPC's own law, which with the gap rings but did not run away.

The simulated plant, its instruments and the MPC's model are kept separate. The simulation integrates Equations (12.1) to (12.5) with whatever the run carries, the MPC predicts and estimates with the frozen declared values, and it reads only what the cascade reads, the two encoders, the measured current, the profile, K_t and the current limit. Because its estimator writes nothing the cascade uses, each earlier case gives the same logged values to the last bit on each column recorded before the MPC was added.

12.5 Ten Cases and a Move and Return

The cases of Table 12.4 start from the defaults of Tables 12.2 and 12.3, in automatic mode unless the row says otherwise, and depart from them only as the row states. Each number in this section comes from the engine itself, run headless as the browser runs it, with every setting a case changes made before Run, and the main ones are in Table 12.5. The same settings carried through Reset give the same run. Table 12.7 lists what the application requires before it credits each case, the completion rules of Section 1.6.4. A case counts as done once the changes it lists have been made and the simulated axis has reached the state it names.

12.5.1 Torque by Hand, and the Current Loop at Speed

Case 1 opens the speed and position loops and steps the current command, the drive's torque request, to 3 A, −3 A and zero. Torque integrates to speed and speed to position. After 0.1 s at 3 A, 0.36 N m against 6 kg cm^2, the motor turns at 429.8 rpm and the table has gone 3.58 mm. The reversed current has the guides' friction on its side, so it stops the axis before 0.25 s and drives it back to −163.7 rpm, and with the current at zero the friction brings the table to rest at 3.54 mm. Each step also rings the coupling. A damped sine fitted to the first 20 ms after the step swings at 398 Hz with a damping ratio of 0.054, about a mean of 0.31 N m late in the push with a peak of 0.57 N m. That is above the shaft's own 0.048 and close to the 0.056 of the linear model of Section 12.4, which includes the back-EMF that the current loop does not cancel at the mode. The speed loop will have to live with this mode.

Case 2 pushes the current loop against its bus. At rest the step to 10 A rises in 0.40 ms, but at the 1391 rpm of the scan that commands 20 A the back-EMF already takes 11.7 V of the 27.71 V circle, the step from 10 to 20 A needs 1.30 ms from 11 to 19 A, and 20 A holds only up to 1777 rpm before

Table 12.4 Scenario Settings and Teaching Points

#	Change from the defaults	Teaching point
1	speed and position loops manual, i_q^* 3, −3 and 0 A at 0.05, 0.15 and 0.25 s	torque, speed, position
2	as 1, i_q^* 10, 20, −20 and 0 A at 0.02, 0.10, 0.16 and 0.24 s	the voltage circle
3	position loop manual at 300 rpm, $K_p = 0.7$	the resonance caps the gain
4	position loop manual, $J_l = 1\,\text{kg cm}^2$, $K_p = 0.4$, $\tau_I = 15\,\text{ms}$, steps to 500 rpm and back twice, a workpiece of +400 % at 0.28 s	tuned at 1:1, run at 5:1
5	gap 1°, load encoder, $\tau_{I,x} = 50\,\text{ms}$, move to 10 mm	an integral that hunts
6	move to 100 mm	following error
7	as 3 at $K_p = 0.3$, relay at 0.2 s, 600 rpm at 0.5 s	what the relay measures
8	$K_p = 0.5$, feed to 200 mm at 50 mm/s, ripple 100 N at 425 Hz	a force at the mode
9	feedforward on, 250 mm at 600 mm/s and 2.5 m/s^2	field weakening
10	MPC in place of the position and speed loops, load encoder, feedforward, notch, $K_p = 0.8$, $\tau_I = 12\,\text{ms}$, $K_v = 70\,\text{s}^{-1}$, move to 100 mm, 300 N cut at 0.6 s, 1 s	a model of the axis
C	gap 0.5°, 100 mm out and back with a 300 N cut on the return, 1.6 s	everything at once

Table 12.5 Headless Measurements of the Nine Cases

Case, setting	Measured
1, after 0.1 s at 3 A	429.8 rpm, 3.58 mm, shaft rings at 398 Hz, $\zeta = 0.054$
2, at rest and at 1391 rpm	a 10 A step in 0.40 ms, 10 to 20 A in 1.30 ms, 20 A held to 1777 rpm
2, bus sagged 25 %	the step to 20 A reaches 13.6 A, at 1464 rpm
3, K_p 0.7, no notch	sings at 413.8 Hz, PM −8.4°, GM −1.4 dB
3, quiet limit, no notch, notch	0.55 and 1.75 (model 0.592 and 1.821)
3, gain margin 6 dB	0.297 without the notch, 0.904 with it
4, 1:1 and 5:1	bandwidth 58.1 and 22.6 Hz, PM 63.6 and 27.8°, GM 8.1 and 3.4 dB
4, 5:1 at K_p 1.2	unstable, or 55.5 Hz at 61.2° with the notch
5, K_v 10 to 80	hunts 994 to 271 counts, 381 with no gap
5, $\tau_{I,x}$ 200 ms or none	settles, within 0.4 counts with no integral
5, motor encoder	stops 58.7 counts short, 1.6 with $2\hat{\alpha} = 0.8°$
6, feedforward off and on	worst error 9.990 and 0.035 mm, in position 554 and 343 ms after the profile
7, relay, no notch and notch	432.4 Hz, K_u 0.641, and 500.0 Hz, 1.817
7, 600 rpm step	overshoot 82.2 (relay), 58.2 (relay, notch), 29.7 (shipped), 24.7 % (notch, K_p 1.2)
8, 425 Hz, K_p 0.5, 0.3, notch	motor ±2.80, ±0.79, ±0.35 µm
8, 163 Hz, any loop	motor ±0.76 µm, table ±0.69 to ±0.81 µm
9, field weakening off and on	top 3209 and 3609 rpm, error 15.19 and 0.02 mm, $\lvert i\rvert$ 32.3 and 14.7 A

the voltage runs out, with VOLTAGE LIMIT showing for 55.5 ms of the run. With the bus sagged by 25 % the step to 20 A reaches only 13.6 A, at 1464 rpm, and falls from there as the speed rises. A bandwidth of 926 Hz is a small-signal number, and a large step at speed is limited by the voltage left over.

12.5.2 The Gain That Sings, and the Notch

Case 3 runs the axis at 300 rpm under the speed loop alone with its gain raised from the commissioned 0.3 to 0.7 A s/rad. At 0.3 the open loop's resonance peak stands 0.91 dB below unity. At 0.7 it stands 6.45 dB above, and $|L(j\omega)|$ crosses unity three times, at 22.9 Hz with 64.3° of phase margin, upward at 360.8 Hz, and downward again at 441 Hz, where the phase is already 8.4° past −180°. The gain margin is −1.4 dB, as shown in Figure 12.2. The axis sings from 3.0 ms, at 413.8 Hz. The shaft torque swings between −10.6 and 10.3 N m, more than eight times the motor's rating, while the q-axis current, once the singing has settled, stays between −15.9 and 17.5 A. Switched on while the axis sings, the notch quiets it 84.9 ms later. Every run of the case opens with the step to 300 rpm at $t = 0$, which steps the current command by K_p times the 31.4 rad/s error, up to its limit, and rings the shaft, so even a quiet setting shows RESONANCE at first, for 49 to 55 ms with the notch at gains from 1.0

to 1.6, 113 ms at 1.75 and 218.7 ms without the notch at 0.55. Quiet here means quiet once that ring has died away, as it has by 0.6 s. Without the notch the axis is quiet up to a gain of 0.55 and sings from 0.60, where the frequency response puts the limit at 0.592, and with the notch it is quiet up to 1.75 and sings again from 1.80, at 489.8 Hz, where the loop above the notch runs out of phase (the model says 1.821). The quiet limits leave almost no margin, 0.65 dB at 0.55 and 0.26 dB at 1.75, and at an equal gain margin of 6 dB the notch still triples the usable gain, from 0.297 to 0.904, and the bandwidth from 16.7 to 41.3 Hz.

> **TRY IT IN THE APP** **3 · The Speed Gain Until the Axis Sings**
>
> Load the case and press Run. Within the first milliseconds the status line reads RESONANCE, the shaft sings at 414 Hz, and the shaft-torque trend swings far past the motor's rating. Then switch the Notch filter on in the speed-loop card while it sings, and watch the ringing die away 84.9 ms later.

12.5.3 Tuned at 1:1, Run at 5:1

Case 4 tunes the speed loop at $K_p = 0.4$ and $\tau_I = 15\,\text{ms}$ with a light fixture of $1\,\text{kg}\,\text{cm}^2$, steps to 500 rpm and back, clamps on a workpiece five times heavier at 0.28 s and steps again with the same gains. At 1:1 the step rises in 5.40 ms with 16.9 % overshoot, at a bandwidth of 58.1 Hz with 63.6° of phase margin and 8.1 dB of gain margin, the mode sitting at 522.1 Hz. At 5:1 the same gains rise in 12.20 ms with 31.4 % overshoot. The bandwidth has fallen to 22.6 Hz, and the resonance, now at 401.1 Hz and much taller, crosses unity again near 413 Hz and leaves 27.8° and 3.4 dB, so the shaft rings at a band RMS of 0.60 N m on each step against 0.24 N m before. On a frequency grid finer than the Bode view's 260 points that margin is 27.1°. A sine sweep of the simulated loop measures the same bandwidths, 58.1 and 22.6 Hz. Getting the bandwidth back needs the gain raised with the inertia, to about 1.2, which without the notch is unstable (−31.4°, −6.1 dB) and with it gives 55.5 Hz at 61.2°, though with only 3.5 dB of gain margin. A workpiece five times heavier clamped on between two speed steps while the drive holds the axis would not happen on a real machine, but it isolates what the inertia ratio does to a fixed tuning.

12.5.4 A Gap, Friction and an Integral That Hunts

Case 5 puts 1° of backlash, 182.0 load counts or 27.8 µm of table travel, in the gears and closes the position loop on the load encoder with a 50 ms integral added, so that the table itself is measured and every last micrometer is supposed to be corrected. It never settles. Over the last 2 s of a 4 s run the table hunts 482 counts (73.5 µm) peak to peak, reversing 69 times and crossing the gap 10 times, and no position gain removes it, 994 counts at $10\,\text{s}^{-1}$ and 271 at $80\,\text{s}^{-1}$. With the gap closed it still hunts, 381 counts, because the cycle is the integral winding up while the guides' friction holds the table, then driving it past, the hunting that the friction literature describes for integral action against friction [8]. The gap only turns each reversal into a rattle across the teeth. Removing the cause works. A 100 ms integral still hunts 245.9 counts over the last 6 s of a 12 s run, a 200 ms one stops it, and so does none at all, because the speed loop's own integral already holds the table, which then settles within 0.4 counts of its target with or without the gap. On frictionless guides the gap alone makes the same loop without an integral hunt 249.8 counts, the textbook backlash limit cycle [7], which the friction here suppresses. On the motor encoder there is nothing to hunt, but the table stops 58.7 counts short in the direction of every move, where the encoder cannot see it. Backlash compensation set to the gearbox's 1° overcorrects by 14.8 to 16.5 counts, because the table floats into the gap as it decelerates, and set to 0.8°, found by trying, it leaves at most 1.6.

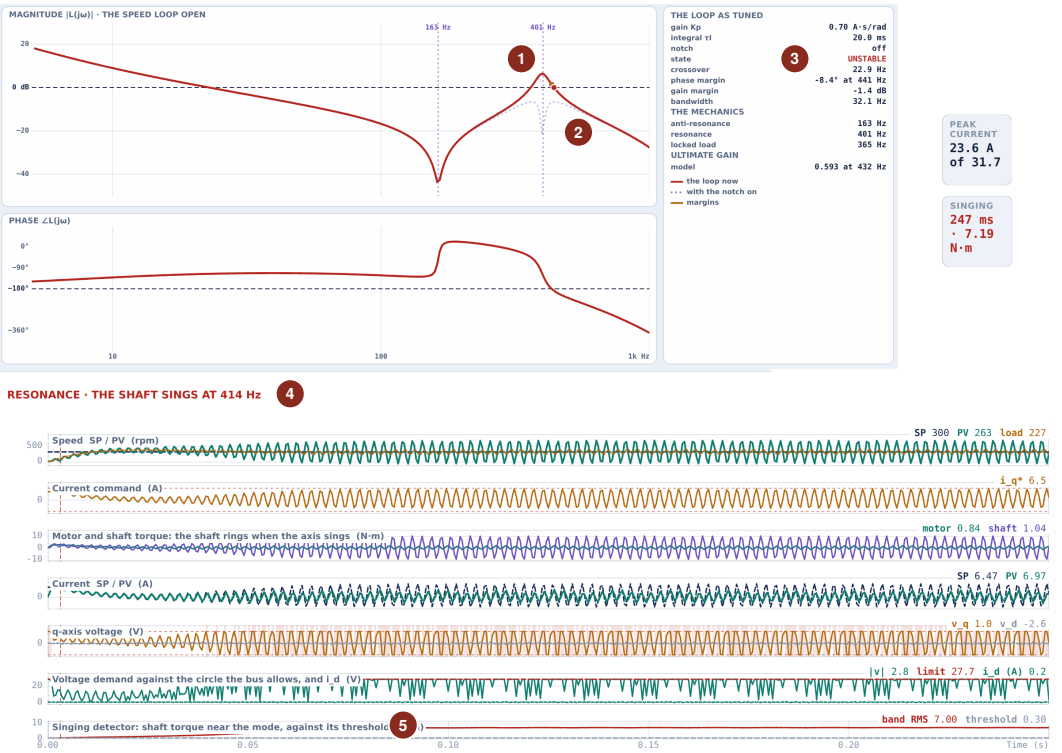


Figure 12.2 The speed loop of case 3, *The speed gain until the axis sings*, at $t = 0.25$ s with no control changed. The figure is a composite of the Bode view, two scorecard tiles set at its right and the trend strips below it. The open loop at $K_p = 0.70$ A s/rad (the solid curve, red on the screen) lifts its resonance peak at 401 Hz above 0 dB (1) and keeps more than unit gain near 432 Hz, where its phase passes $-180°$, and the dotted curve is the same loop with the 400 Hz notch on (2). The column beside the plots (3) reads $\tau_I = 20.0$ ms, the notch off, the state UNSTABLE, the crossover at 22.9 Hz, a phase margin of −8.4° at 441 Hz, a gain margin of −1.4 dB, a bandwidth of 32.1 Hz, the anti-resonance at 163 Hz, the resonance at 401 Hz, the locked load at 365 Hz and the model's ultimate gain of 0.593 at 432 Hz. The status line (4) reads RESONANCE · THE SHAFT SINGS AT 414 HZ. Over the 0.25 s window the trends show the speed, the current command and its tracking, the motor and shaft torques and the q-axis voltage oscillating at the mode, and at the last sample the speed at 263 rpm against 300 rpm, the shaft torque at 1.04 N m, the voltage at 2.8 V of 27.7 V and the singing detector (5) at 7.00 N m against its 0.30 N m threshold. The scorecard tiles count 247 ms of singing with a band RMS of up to 7.19 N m and a peak current of 23.6 A of 31.7 A.

> **✦ DID YOU KNOW? The Integral That Causes the Hunt**
>
> From the simulator: with an integral in the load-encoder position loop of case 5, the table hunts 994 to 271 counts at position gains from 10 to 80 s^{-1}, and still 381 counts with the gap closed. With no position integral at all it settles within 0.4 counts (Table 12.5).

12.5.5 Feedforward From the Profile

Case 6 moves the table 100 mm on the S-curve with the commissioned gains and a proportional position loop. Without feedforward the table runs 9.990 mm behind at cruise, is still 2.61 mm short when the 418.0 ms profile ends and comes within two counts of the target 554 ms later. With the feedforward the worst error is 0.035 mm, 285 times smaller, the table is 18.7 µm short at the end

of the profile and in position 343 ms later, with no overshoot, as feedforward from a jerk-limited profile intends [18]. Volume 20, Chapter 10 applies the same idea, torque feedforward computed along a planned trajectory, to a two-link arm. Both terms of Equation (12.11) matter. Without the friction term the speed integral carries the guides' friction through the move and pushes the table 152.5 counts past its target when it stops, and an inertia estimate taken at 1:1, 2 kg cm^2, leaves a 0.734 mm error and a 2683-count overshoot. Run on the speed loop of case 3 at the commissioned gain of 0.3, which is case 7 without its relay test and speed step, the chirp identification recovers the inertia, 6.010 to 6.068 kg cm^2 for mean speeds of the sweep from 0 to 600 rpm, within 1.1 % of the 6.000 kg cm^2 of the rigid body. Its friction estimate is another matter, 18.4 to 213.3 mN m against the guides' 80 over the same speeds, with the drag at −1.78 to 1.00 mN m s/rad against 0.30, which is why the drive passes only the inertia to the feedforward.

> **TRY IT IN THE APP** **6 · S-Curve Feedforward**
>
> Press Run on the Profile view and read the following-error strip: the table runs 9.990 mm behind its command at cruise. Press Reset, set Profile feedforward On in the position card and run again. With no gain changed, the worst error falls to 0.035 mm, and the table is in position 343 ms after the profile ends instead of 554 ms.

12.5.6 What the Relay Measures

Case 7 replaces the speed PI at 300 rpm by a 2 A relay for 27.5 ms from 0.2 s. The loop oscillates at 432.4 Hz, just above the resonance, where its phase passes −180°, with an ultimate gain of 0.641 A s/rad, 8 % above the 0.593 that the linear model gives at the same 432.4 Hz and within the 10 % that the application's check allows. The Ziegler–Nichols rule turns that into $K_p = 0.288$ and $\tau_I = 1.93$ ms, an integral time tuned to a 2.31 ms period that belongs to the coupling rather than to the rigid-body motion the integral has to correct, near a crossover of 11.5 Hz. The step to 600 rpm on those gains overshoots 82.2 % with 8.6° of phase margin, against 29.7 % for the commissioned gains put back after the test. With the notch on before the test the relay finds 500.0 Hz and 1.817, the model 1.804 at 495.1 Hz, and the same rule gives 1.67 ms and a 58.2 % overshoot. A notch design by hand, $K_p = 1.2$ with $\tau_I = 15$ ms entered after the test, steps in 5.5 ms with 24.7 % overshoot and 61.2° of phase margin. The relay measures the ultimate point correctly, and what fails is the rule's assumption that the oscillation it finds is the loop's dominant dynamics.

Pause. Try Exercise 12.1 before reading on.

12.5.7 A Cut at the Mode and at the Anti-Resonance

Case 8 feeds the table at 50 mm/s with the speed gain at 0.5, stable with 1.5 dB of gain margin, while the cut ripples ±100 N at 425 Hz, as a cutter's teeth would at that tooth-passing frequency. Volume 7, Chapter 13 treats the cut itself, where each tooth meets the surface the previous one left. The loop feeds the mode. The motor shakes ±2.80 µm and the table ±0.44 µm, and RESONANCE shows from 44.7 ms for the rest of the run. At a gain of 0.3 the motor moves ±0.79 µm and nothing rings, and with the notch at 0.5 the motor moves ±0.35 µm and the table ±0.12 µm. At 163 Hz, the anti-resonance, torque at the motor cannot move the motor, so the motor shakes ±0.76 µm whatever the loop does, while the table shakes ±0.69 µm at a gain of 0.3, ±0.71 µm at 0.5 and ±0.81 µm at 1.5 with the notch, because the torque the loop answers with goes into the table. At 365 Hz, the locked-load frequency, the table barely moves (±0.05 µm) while the motor shakes for it (±0.26 µm), as a tuned absorber would. A notch removes the controller's gain at one frequency and leaves every other frequency to the mechanics.

Pause. Try Exercise 12.3 before reading on.

12.5.8 Field Weakening for a Rapid

Case 9 programs a 250 mm rapid at 600 mm/s, 3600 rpm at the motor, with the feedforward on and field weakening off. The voltage runs out first. The motor tops out at 3209 rpm, the following error

Table 12.6 The Move and Return in Nine Configurations

Configuration	In (s)	End	GM (dB)	$\|i\|$ (A)	Sings (ms)
as shipped	never	56.6	5.92	21.7	0
suggested, motor encoder	never	55.5	7.06	22.5	120
and compensation 0.4°	never	18.7	7.06	22.4	124
suggested, no feedforward	1.066	0.4	7.06	20.1	53
suggested, no notch	never	739.95	−2.60	31.5	1569
no notch, K_p 0.25, τ_I 20 ms, K_v 80	1.041	0.3	7.50	24.7	121
no notch, K_p 0.45, τ_I 8 ms, K_v 80	0.9795	−0.2	2.40	24.8	275
the first with no dwell	0.989	0.3	7.50	24.8	116
suggested	0.989	0.3	7.06	22.4	166

End is the table's final distance from zero in load counts. The suggested configuration is the load encoder, the feedforward, the notch, $K_p = 0.8$, $\tau_I = 12$ ms and $K_v = 70\,\mathrm{s}^{-1}$. Only it meets the whole target, and the run with no dwell is in time but is not the programmed move.

grows to 15.19 mm and VOLTAGE LIMIT shows for 389 ms. When the axis then brakes, the d-axis decoupling its braking current needs, up to 38.8 V at 2957 rpm, is more than the circle holds, so for 19.3 ms, from 3078 down to 1988 rpm, the voltage sits on the circle, and for 18.1 ms of it the d axis takes the whole circle, v_q is left at zero and the back-EMF rather than the drive sets the currents. The d-axis current swings to −11.8 A that nobody commanded, and the q-axis current overshoots its −31.67 A limit as the voltage comes back, so $|i|$ reaches 32.3 A, past the drive's peak. With field weakening the axis follows the profile to 3609 rpm with 0.02 mm of following error and never reaches the voltage limit, at the cost of a d-axis current down to −11.28 A that makes no torque, while $|i|$ peaks at 14.7 A. Slowing the rapid to 500 mm/s instead keeps the error to 0.34 mm with the limit touched for 76 ms.

12.5.9 The Move and Return

The graded case moves the table 100 mm, dwells 50 ms and brings it back while a 300 N cut resists the return and stays on at the end, with 0.5° of backlash and the 5:1 load. The target asks for the table within two load counts (0.305 µm) of zero by 1.02 s and staying there, never more than two counts past either target, no following-error fault, the current within the 31.7 A peak and at least 6 dB of gain margin in the speed loop, with the programmed profile, dwell and machine. The target is judged on the machine and the cut as given, so a changed model parameter or disturbance, or an extra disturbance button, voids the attempt. Table 12.6 compares configurations, Figure 12.3 charts three of its columns, and Figure 12.4 shows four returns. As shipped, on the motor encoder with no notch and no feedforward, the following error reaches 9.991 mm and the table ends 56.6 counts from zero, mostly the half gap of 45.5 counts and the twist of 9.5 under the cut, which a motor encoder cannot see, and the gain margin of 5.92 dB also falls short. The suggested gains on the motor encoder still end 55.5 counts off, and backlash compensation of 0.4° only brings that to 18.7 while overshooting the first target by 48.8. On the load encoder with the feedforward, the notch, the speed loop at $K_p = 0.8$ and $\tau_I = 12$ ms and a position gain of $70\,\mathrm{s}^{-1}$, the table is in position at 0.989 s with no overshoot, a peak current of 22.4 A and 7.1 dB of gain margin. Each piece is needed. Without the feedforward the table arrives at 1.066 s, without the notch the same gains sing, and the best found without the notch inside the 6 dB margin arrives at 1.041 s, while a quicker one at $K_p = 0.45$ arrives at 0.9795 s with only 2.4 dB of margin.

🕮 **Pause.** Try Exercise 12.2 before reading on.

The suggested configuration passes, but not quietly. The singing detector reports RESONANCE for 166 ms of its run, in five bursts, at the start and in the deceleration of each move and once just after the return, each following a crossing of the gap, with a band RMS of up to 0.58 N m. The teeth are apart for 77.8 ms in all, and with the gap closed the same gains never ring, which is the limit of a notch on a geared axis, since every strike rings the mode afresh. The target counts the time from the start of the run, so the dwell is part of the program. With the dwell set to zero, the best

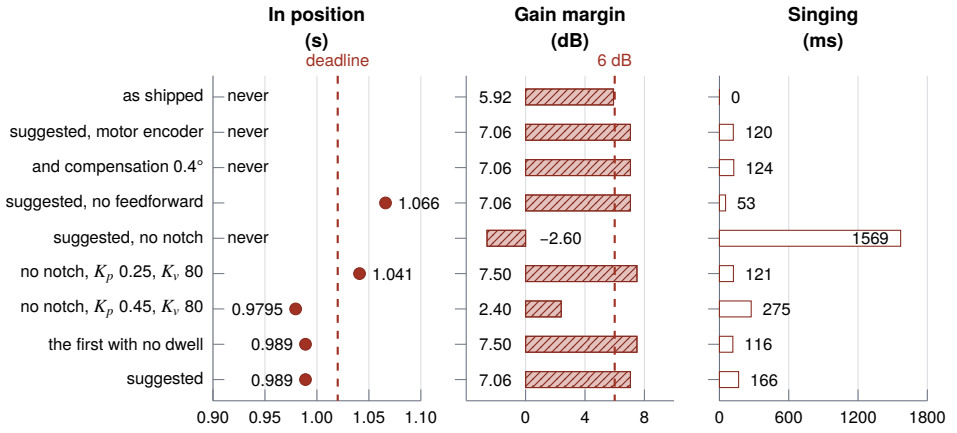


Figure 12.3 The move and return of Table 12.6 in its nine configurations: the time at which the table comes to stay within two load counts (dots, with the 1.02 s deadline dashed), the gain margin of the speed loop (hatched bars, with the 6 dB floor dashed) and the time the shaft sings (open bars). Notice that only the suggested configuration comes in on time with at least 6 dB on the programmed move, that the one quicker run pays for it with 2.40 dB of margin, and that every configuration but the shipped one, which never comes into position, sings for part of the run.

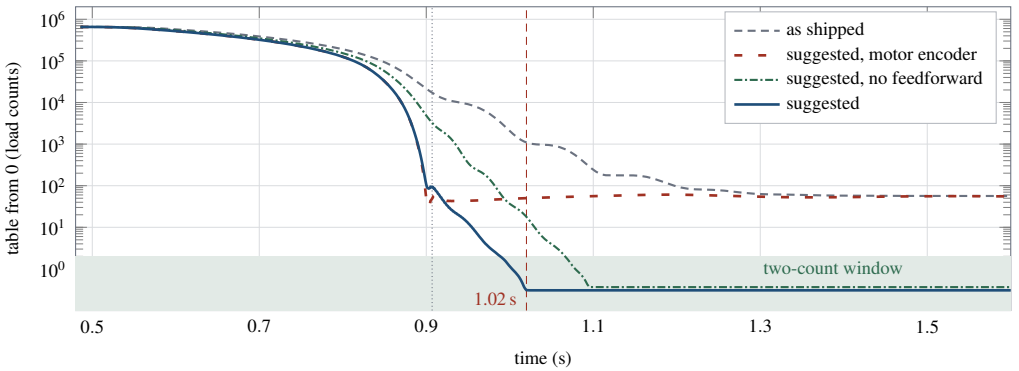


Figure 12.4 The return of the graded case, the table's true distance from zero in load counts. The motor-encoder runs, as shipped (densely dashed) and at the suggested settings (loosely dashed), settle 56.6 and 55.5 counts out, where the encoder cannot see the gap and the twist. The load-encoder runs enter the two-count window at 0.989 and 1.066 s with the feedforward (solid) and without it (dash-dotted). The vertical dotted line marks the end of the return's profile at 0.907 s and the vertical dashed line the deadline.

configuration without the notch is in position at 0.989 s, and the application scores such a run as not the programmed move (Table 12.6).

12.5.10 Explicit MPC Against the Cascade

Case 10 runs the 100 mm move of case 6 on the load encoder with no gap, then a 300 N cut that arrives at 0.6 s with the table at rest, and pins the MPC with the cascade beside it at the challenge's suggested tuning. Table 12.8 runs that program and a harder move, 10 m/s^2 and 2000 m/s^3, which asks for more than the drive's peak torque, on both encoders, one engine per row, with the same events, plant, quantization and duration in every row. The model has no random term, so every row replays exactly. The cascade appears as shipped, with its feedforward, at the suggested tuning

Table 12.7 Scenarios and Completion Requirements

#	Scenario	Kind	Required to complete
1	Current by hand	Experiment	Run through the scripted current steps of 3, −3 and 0 A, then from Reset drive the current command by hand until the motor turns faster than 100 rpm for 20 ms.
2	Current loop against the bus	Experiment	Hold the 20 A command on the full bus until $\|v\|$ has sat on the voltage circle for 20 ms, then from Reset, with the bus sagged by at least 20% (25% is suggested), run until the current stays below 18 A against its 20 A command for 10 ms.
3	Speed gain until the axis sings	Demonstration	Let the shaft sing at $K_p = 0.7$ without the notch for 0.1 s, switch the notch on while it sings, then with the notch on raise K_p to 1.0 or more and run 0.2 s with no resonance.
4	Tuned at 1:1, run at 5:1	Experiment	Run to 0.45 s with the shipped gains through the heavier workpiece at 0.28 s, then from Reset raise K_p to 1.0 or more and switch the notch on before 0.30 s, so that the linear loop keeps a bandwidth of at least 45 Hz and a positive phase margin for the first 0.1 s of the 5:1 step.
5	Backlash and an integral position loop	Demonstration	With the position integral on, see the table hunt more than 100 load counts peak to peak between 1 and 2 s, again from Reset at a position gain from 10 to 80 s^{-1} other than 30, then from Reset with the integral off run at least 1.5 s until the table is within one count of its target.
6	S-curve feedforward	Experiment	Run the 100 mm move without feedforward until the table is in position, then from Reset with the feedforward on until it is in position again with a worst following error below 0.1 mm.
7	Relay autotuner	Experiment	Run to 0.6 s through the relay test and the 600 rpm step with the notch off, then from Reset with the notch on before Run, again to 0.6 s.
8	Cutting force at the resonance	Experiment	Run 0.15 s against the 100 N ripple at 425 Hz with $K_p \geq 0.45$ and no notch, then from Reset with the notch on run 0.15 s against the same ripple with no resonance, then move the ripple to 163 ± 10 Hz and run 0.15 s more.
9	Field weakening for a rapid	Experiment	Run the 250 mm rapid without field weakening until its profile ends, then from Reset with field weakening on until the profile ends with a worst following error below 1 mm.
10	Explicit MPC against the cascade	Experiment	Run the move and the 300 N cut under the explicit MPC to 0.9 s, then from Reset switch the position and speed control to the cascade and run the same move and cut to 0.9 s.
11	Move, return and settle	Challenge	The table back within two load-encoder counts of 0 by 1.02 s and staying there to 1.6 s, never more than two counts past either target, no following-error fault, the current within 31.7 A and at least 6 dB of gain margin, with the programmed profile and dwell and the machine and cut as given.[a]

[a] With the shipped settings the table never comes within two counts, ending 56.6 counts from 0, and the gain margin of 5.92 dB is short of 6 dB. With the suggested settings (load encoder, feedforward, notch, $K_p = 0.8$, $\tau_I = 12$ ms, $K_v = 70$ s^{-1}) the table is in position at 0.989 s, 31 ms inside the deadline, with no overshoot, a peak current of 22.4 A and 7.06 dB of gain margin.

and retuned once by a rule fixed before measuring, the largest speed gain on a 0.01 grid that keeps 6 dB of gain margin with the notch and $\tau_I = 12$ ms, then K_v at a quarter of the resulting bandwidth ($K_p = 0.90$, $K_v = 71\,s^{-1}$), with the feedforward. The MPC runs at case 10's settings throughout, and two counterfactual rows remove its torque rows or its preview. Every MPC row makes one decision per millisecond (1001 on case 10, 601 on the hard move) with no fallback and no supervisor trip, at a mean logged time of 0.4 to 2.3 µs with the lookup in Node.js. The online solve reproduces every MPC row, except on the harder move with the motor encoder, where it differs by 0.1 ms and 0.2 counts.

On the load encoder the MPC brings the table within two counts 416.1 ms after the move begins, before the profile itself ends at 418.0 ms, against 476.0 ms for the suggested cascade, and it never strays more than 0.69 µm from the command during the move, against 12.78 µm, at a lower peak torque, 2.030 against 2.119 N m. When the cut arrives it yields 39.4 counts (6.0 µm) against 361.3

Table 12.8 Explicit MPC Against the Cascade on the Move, the Cut and a Move Past the Peak Torque

Structure	In (ms)	Past (cnt)	Worst (μm)	T_e (N m)	Cut (cnt)	Back (ms)	End (cnt)
Case 10, load encoder							
Cascade, shipped	—	0	9990	2.012	1625	—	32.8
with feedforward	—	0	38.48	2.090	1818	—	23.2
Cascade, suggested	476.0	0	12.78	2.119	361.3	90.8	0.3
Cascade, retuned	474.3	0	11.37	2.117	316.3	88.8	0.3
MPC	416.1	0	0.69	2.030	39.4	13.8	0.0
no torque rows	416.1	0	0.69	2.030	39.4	13.8	0.0
preview off	469.5	3.4	1527	2.026	42.8	13.5	0.4
Case 10, motor encoder							
Cascade, suggested	509.6	0	14.60	2.120	366.7	—	11.0
Cascade, retuned	502.1	0	13.16	2.119	321.4	—	11.1
MPC	—[a]	2.6	0.60	2.035	54.8	15.9	−2.1
Move past the peak torque, load encoder							
Cascade, suggested	512.8	544.5	471.8	3.806	—	—	0.2
Cascade, retuned	501.4	511.6	467.6	3.807	—	—	0.2
MPC	378.6	1.6	194.5	3.807	—	—	0.2
no torque rows	378.9	0.8	204.6	3.807	—	—	−0.7
Move past the peak torque, motor encoder							
Cascade, suggested	554.5	567.7	476.0	3.806	—	—	1.7
MPC	—[a]	4.6	192.5	3.805	—	—	1.9

In is the time after the move's start at which the table comes to stay within two load counts of the target, Past the largest excursion beyond it, Worst the largest distance between the table and the command at the position loop's instants during the move, T_e the peak motor torque of the move, Cut the excursion under the cut, Back the time after the cut at which the table comes to stay within ten counts (1.5 μm), and End the error at the end of the run (1 s, or 0.6 s for the harder move), all from the true state at every 0.1 ms step. [a]The table enters the two-count band 50 times before the cut on case 10 and 70 times before the end on the harder move, the last at 577.8 and 580.0 ms, without staying.

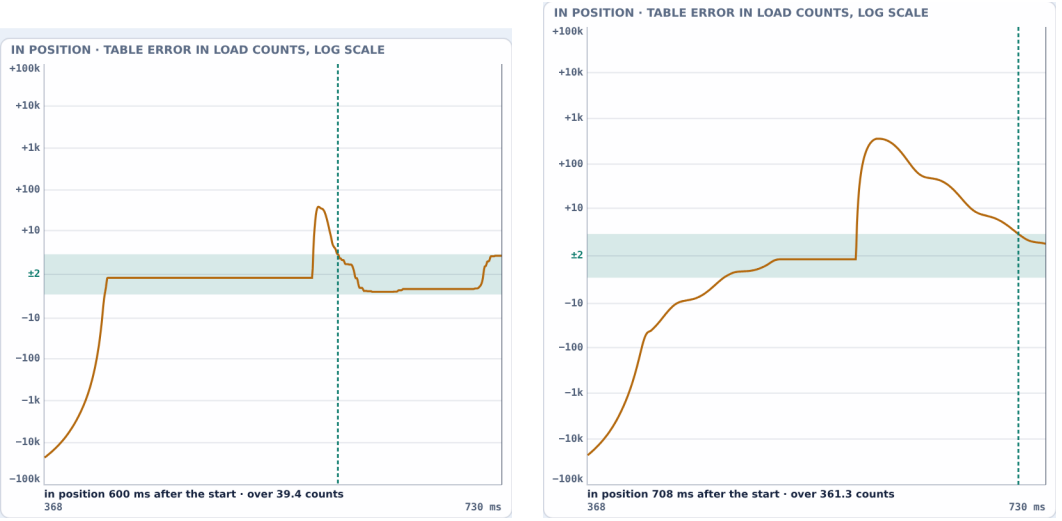
and is back within ten counts in 13.8 against 90.8 ms. The retuned cascade recovers less than 2 ms of the 60 ms (474.3 ms) and 45 counts of the excursion. Within two counts rather than ten the recovery reads 20.4 against 128.4 ms, but that figure turns on the last count at the guides' friction edge, and with the physics step halved it becomes 264.5 against 128.3 ms, while the in-position times, 416.8 and 475.9 ms, and the excursions, 43.0 and 361.4 counts, keep their order. The cut costs the MPC a higher peak torque than the cascade, 0.811 N m against 0.628, the price of holding the table closer.

The two counterfactual rows locate the gain. Without its torque rows the MPC reproduces every number on case 10, because the bound never binds on that move, and on the harder move it comes in at 378.9 ms against 378.6 with 0.8 against 1.6 counts past the target, so the drive's own clamp would serve nearly as well. The faster move and the stiffer hold come from the model, which puts in the twist an acceleration needs, from the load-torque estimate and from the preview, not from the constraint. Without the preview the MPC lags the profile by 1.5 mm, as a loop without feedforward must, and still comes in at 469.5 ms, ahead of the suggested cascade with its feedforward, with 3.4 counts past the target. The cascade's own trouble on the harder move is windup. With the current at its limit the speed integral winds, and the table passes the target by 544.5 counts, where the MPC passes it by 1.6 and is in position at 378.6 ms against 512.8.

The motor encoder reverses part of the result. There the MPC never settles inside two counts before the cut, entering the band 50 times between the move and the cut, and it ends the cut 2.1 counts off, because the table's position and the twist under the cut are then only the filter's estimate. The suggested cascade is in position at 509.6 ms but ends the cut 11.0 counts off and never returns within ten counts. On the harder move the MPC passes the target by 4.6 counts and is still crossing the band at the end of the run, where the cascade passes it by 567.7 counts and is in position at 554.5 ms.

The two runs of case 10 as loaded and with the cascade selected are shown at 0.75 s in Figure 12.5, on the profile view's strip of the table's true error in load counts.

Table 12.9 sets the worst value of each safety-relevant output under the MPC against the cascade over 45 runs, every existing case (the challenge also at its suggested settings) and case 6's move with every machine setting, parameter and disturbance at its extremes, the events and the move shapes.



(a) Explicit MPC, as loaded

(b) Cascade at the suggested tuning

Figure 12.5 Case 10, *Explicit MPC against the cascade: a move and a cut*, at $t = 0.75$ s on the profile view, the table's true error in load counts on a signed logarithmic scale from 368 to 730 ms after the move began, with the two-count band shaded. (a) The MPC enters the band before the cut, the cut lifts the table to 39.4 counts, and the readout gives in position 600 ms after the start (the dashed line), 20 ms after the cut. (b) The cascade enters the band later, the cut lifts the table to 361.3 counts, and the readout gives in position 708 ms after the start, 128 ms after the cut.

Table 12.9 Worst Case of Each Safety-Relevant Output Over 45 Runs

Output	Cascade	MPC
Following-error faults	step move	step move
Top speed (rpm)	5268 ($c_s = 0$)	3209 (case 9)
Peak measured $\lvert i \rvert$ (A)	32.96	32.98
Runs with $\lvert i \rvert$ above the peak	10	18
Largest band RMS (N m)	91.28 ($c_s = 0$)	7.21 (case 3)
Peak shaft torque (N m)	128.8 ($c_s = 0$)	10.60 (case 3)
I^2t trips	0	0

The step move has no profile and faults under both structures. The peak current of both is reached with the load at 20 kg cm^2. Case 3 runs the speed loop alone, so its singing is the cascade's under either selection.

The MPC never faults or overspeeds where the cascade does not, and its top speed of 3209 rpm is the cascade's own in case 9, while the cascade with the coupling's damping at zero is unstable, reaching 5268 rpm and a band RMS of 91.3 N m, a case the MPC holds quiet. The commanded current never passes the drive's limit under either structure, but the measured current passes the peak in 18 runs under the MPC against 10 under the cascade, by at most 1.31 A (4 %, with the load at 20 kg cm^2), because the current loop overshoots a command held at the limit. The MPC is worse than the cascade in 26 of the 45 runs on at least one of singing, peak current, shaft torque or end error, and better in 15. It rings with the gear gap, in case 5 for 187.2 ms against 81.4, and with a machine its model does not describe, for 1.06 s against none with the shaft stiffened to 2000 N m/rad, 706.8 ms with the guides' friction at 0.4 N m and 1.18 s with a 500 N ripple at the 163 Hz anti-resonance. At the voltage limit of case 9 the supervisor trips once, on the following error, and the MPC sings for 42.1 ms where the cascade does not. It is better where the mode, not the model, decides. A cut rippling at 425 Hz in case 8 makes the cascade sing for 1.56 s and the MPC not at all, and both glitches leave the MPC quiet where the cascade rings for 36 and 48 ms.

12.6 Technology in Practice

The axis of Section 12.3 reads its phase currents and two encoders and acts through an inverter. Figure 12.6 draws its three loops with real devices in the place of the schematic's symbols.

What Measures It

Motor speed was once read from a tachogenerator, a small direct-current generator whose voltage is proportional to speed, and an optical encoder does the same work by chopping a light beam into pulses [23]. The motor-side loops rest on such an encoder, described below. A linear encoder can also read the table of a machine tool directly, and an enclosed one keeps out dust, coolant and chips [24]. The 20-bit motor encoder and the 16-bit screw encoder of Section 12.3 quantize exactly and do nothing else, so the mounting errors of a real encoder are what the model leaves out.

> **TECHNOLOGY Sensor: The Optical Rotary Encoder**
>
> A code disc turns with the motor shaft. Its transparent and opaque, or reflective, segments interrupt a light beam, and a stationary index grating and photodetectors turn the passing pattern into signals. Single-chip sensors now carry the light source, the grating, the detectors and an interpolator that divides each line further [25]. An incremental encoder counts pulses and must return to zero after power-up, while an absolute encoder reads a coded pattern and keeps its position with the power off [24]. Mounting sets the accuracy. A disc 5 µm off center on a 65 mm grating reads up to about 32 arcseconds wrong, an error that repeats once every revolution, and one test head measured a span of about 206 arcseconds, mostly from that cause [25].

What Moves It

The final element is the power stage of the drive. Its inverter turns the voltages that the current loop computes into pulses of the bus voltage. In 1993 a motion controller typically sent its motor amplifiers a velocity reference of plus or minus 10 volts, and an open controller of that year also supported a pulse-width modulated one [26]. The averaged inverter of Equation (12.2) stands for the bridge without its switching ripple, dead time or braking circuit, and the bus sag of case 2 is the one fault of it that the model carries.

> **TECHNOLOGY Final Element: The Drive's Inverter**
>
> A drive rectifies its supply into a direct-current bus, smooths the bus with a capacitor and switches it through a bridge of power transistors. The duty cycle of the transistors varies rapidly, so that the pulses build the alternating voltage the motor needs. Braking reverses the flow of energy. Current from the motor charges the bus capacitor and raises the bus voltage, and with nowhere to go that energy would destroy the capacitor. A braking transistor and resistor load the bus in proportion to its over-voltage, or an active front end sends the energy back to the line [23]. On a trend, a braking axis shows the bus voltage climbing above its normal level until the clamp takes hold.

Where the Controller Runs

A machine controller synchronizes drives electronically instead of through a mechanical linkage, and its loops are nested, with cycle times from milliseconds to minutes [27]. In the open machine controller that NIST laid out in 1993, a trajectory generator sends setpoints at an interpolation interval to a servo level, which computes actuator commands at a higher rate with PID or other laws [26]. The profile and feedforward of this chapter belong to the first, and the position, speed and current loops at 1, 2 and 10 kHz to the second. Servo loops are so fast that analog controllers were long favored for them [23]. The velocity form of Equation (12.7) runs all three loops at their own periods in one digital controller.

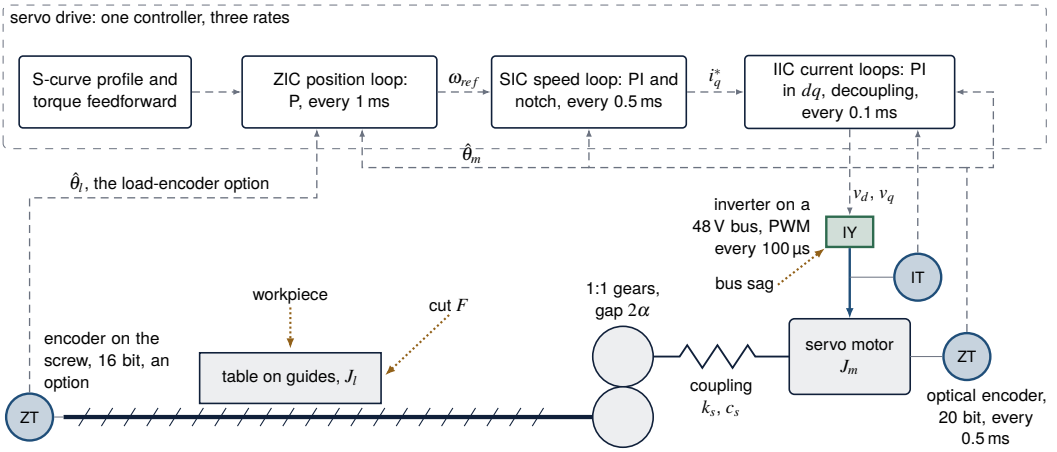


Figure 12.6 The loops of the chapter with real devices in place of the schematic's symbols. Phase-current sensors IT feed the current loops IIC, the optical encoder ZT on the motor feeds the speed loop SIC, the position loop ZIC and the decoupling, and an encoder on the screw can replace it for ZIC. The inverter IY is the final element. Notice that every measurement but the optional screw encoder sits on the motor side of the coupling and the gear gap, so the cut and the twist reach the motor-side loops only through the motion of the motor.

> ⚠ **IN PRACTICE** **Commissioning From the Inside Out**
>
> Motion control is a classic cascade of position, velocity and motor current loops. When a cascade is first commissioned or later retuned, the inner loop is tuned first, because the outer loop depends on it and not the other way round. With the inner loops closed, the position controller no longer faces a runaway process but an integrating one [23]. Errors that repeat are often corrected statically, and lead screw compensation tables were already common in machine controllers in 1993 [26]. The backlash compensation of case 5 is such a static correction, set by trying.

What Is Modernizing It

Encoders are the first development. Absolute encoders, which need no homing and keep their position without power, are replacing incremental ones in high-end motion control [24], and a modular encoder that averages several read heads around its disc calibrates out its own mounting error, by more than 90 % with a second head [25]. Diagnosis is the second. An inertial measurement unit carried on the moving carriage detects micrometer-level and microradian-level degradation of a linear axis, a step toward machine tools that diagnose themselves [19], and digital twins pair physics-based models of a feed drive with models learned from data [28]. Networks are the third. Drives that share one motion network depend on synchronized clocks, which NIST counts as critical for motion control as well as for security [27]. The explicit MPC of Section 12.4.8 is a fourth, computed offline so that a lookup fits the position loop's period.

12.7 Limitations

The axis has one torsional mode, an averaged inverter, an ideal surface-magnet motor, Coulomb friction alone and encoders that do nothing but quantize, as Section 12.3.6 states, so it shows the first resonance of an axis and not the second, which on a real machine is often what limits a notched loop. The cascade's speed loop runs at 2 kHz on a backward difference of the encoder, while observers that estimate the speed or the load can move the limit [3] and appear here only inside the MPC, and derivative feedback of the motor speed, which improves the damping of a two-inertia loop at small inertia ratios [5], is not offered. The linear model behind the Bode view leaves out aliasing and the

nonlinearities, and while it agrees with a simulated sweep within 1.6 % from 200 to 500 Hz, it is 9.3 % off at 700 Hz. The gain margin it reports is that of the proportional loop, within 0.11 dB of the PI loop's, and its phase margin is read on a grid of 260 frequencies, 0.7° high in case 4 at 5:1. The move profiles, a trapezoid and the jerk-limited S-curve, are the only shaping of the command. Input shaping, which Volume 7, Chapter 12 applies to a swinging load, is another way to keep a move from exciting a lightly damped mode, and it is not offered.

> **OPEN CHALLENGE** **A Notch That Follows the Load**
>
> A notch filter is a simple and effective way to suppress a lightly damped resonance, but a load that changes mass moves the resonant frequency. On one piezo-driven positioning stage a shift of only 1 % can make a high-gain loop with a fixed notch unstable. An adaptive notch moves its center as the resonance moves, but identifying the plant inside a closed loop is hard, because the loop yields less information than an open test and the estimate is sensitive to noise [29]. On this axis the heavier workpiece of case 4 moves the mode from 522.1 to 401.1 Hz. Progress would be a notch that follows the mode safely while the machine cuts.

The explicit MPC's model leaves out the gap, the voltage circle, the Coulomb friction beyond its expected direction, the cogging and the quantization, it stands a 0.25 ms lag in for the current loop, and it keeps the declared machine whatever the run carries. It has no speed or voltage row, so at the voltage limit of case 9 the supervisor, not the plan, hands the axis back. The supervisor prevents a runaway but does not make the law robust, and on the motor encoder, with the gap and on a changed machine the cascade remains the better controller. A model of the gap, or a law that stops pricing the twist while the teeth are apart, lies outside this design. Its timings were taken while other computations loaded the host.

Every threshold is a teaching value. The two counts, the 6 dB, the 20 mm window and the 0.30 N m at which the shaft is said to sing were chosen for teaching, and a real axis takes its in-position window and its margins from its machine's accuracy class and its drive maker's commissioning guide. No number in this chapter describes an industrial drive.

> **OPEN CHALLENGE** **Knowing When an Axis Wears**
>
> Machine tool axes degrade in service, and judging how far is usually a manual, time-consuming job whose measurements interrupt production [19]. A ball screw loses its preload with fatigue and wear, which costs precision and opens a backlash [28]. Methods that diagnose from data need long histories that are rarely available in the field, and their features shift with speed and operating conditions [28, 30]. Progress would be an axis that reports a growing gap or a softening coupling from its own encoders and currents during normal moves, before its parts go out of tolerance.

12.8 Conclusion

This chapter commissioned a servo axis from the current loop outward, and the simulator made each step of it measurable against the table's true position. A 1 A current step rises in 0.30 ms at rest, but the current loop holds 20 A only to 1777 rpm, because the back-EMF fills the voltage circle. The two-mass resonance at 399.9 Hz caps the speed gain at 0.55 without the notch and 1.75 with it, and at an equal 6 dB of margin the notch still triples the gain. Gains tuned at 1:1 lose more than half their bandwidth at 5:1, from 58.1 to 22.6 Hz. An integral in a load-encoder position loop hunts 482 counts with a gap and 381 without one, and removing the integral removes the hunt. Feedforward cuts the worst following error from 9.990 to 0.035 mm, the relay tunes to the coupling's 432.4 Hz, and the graded move and return is met at 0.989 s only with the load encoder, the feedforward and the notch, while the gap still rings the shaft for 166 ms. On the declared machine an explicit MPC with four moves and 81 regions is in position at 416 ms against 476 ms and yields 39 counts to the cut against 361, a gain that comes from its model, estimate and preview rather than its torque limit, while the motor encoder, the gear gap and a changed machine leave it behind the cascade.

The simulator's model keeps one mode, an averaged inverter and ideal sensors. A fuller model would add a second mode, static friction, an observer-based speed estimate, an adaptive notch that follows the workpiece, and a load-side encoder with its own errors. The axis shows that a drive is tuned around its coupling rather than its motor.

KEY POINTS

- A machine axis is tuned around the springy coupling between motor and table, not around the motor.
- The two-mass resonance at 399.9 Hz caps the speed gain at 0.55 without the notch and 1.75 with it, and at an equal 6 dB of margin the notch still triples the gain.
- An integral in a load-encoder position loop hunts 482 counts with a gap and 381 without one, and removing the integral removes the hunt.
- Feedforward cuts the worst following error from 9.990 to 0.035 mm.
- The graded move and return is met at 0.989 s only with the load encoder, the feedforward and the notch.

Where else this applies. A fast inner current loop under a slower outer loop, with an explicit MPC beside it, returns in the converter of Volume 3, Chapter 9, and a lightly damped mode under a moving drive in the swinging load of Volume 7, Chapter 12.

References

[1] Y. Altintas, A. Verl, C. Brecher, L. Uriarte, and G. Pritschow, "Machine tool feed drives," *CIRP Annals*, vol. 60, no. 2, pp. 779–796, 2011.

[2] W. Leonhard, *Control of Electrical Drives*, 2nd ed. Berlin, Heidelberg: Springer, 1996.

[3] G. Ellis and R. D. Lorenz, "Resonant load control methods for industrial servo drives," in *Conference Record of the 2000 IEEE Industry Applications Conference. Thirty-Fifth IAS Annual Meeting and World Conference on Industrial Applications of Electrical Energy*, vol. 3, 2000, pp. 1438–1445.

[4] S. N. Vukosavic and M. R. Stojic, "Suppression of torsional oscillations in a high-performance speed servo drive," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 1, pp. 108–117, 1998.

[5] G. Zhang and J. Furusho, "Speed control of two-inertia system by PI/PID control," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 3, pp. 603–609, 2000.

[6] R. Dhaouadi, K. Kubo, and M. Tobise, "Two-degree-of-freedom robust speed controller for high-performance rolling mill drives," *IEEE Transactions on Industry Applications*, vol. 29, no. 5, pp. 919–926, 1993.

[7] M. Nordin and P.-O. Gutman, "Controlling mechanical systems with backlash—a survey," *Automatica*, vol. 38, no. 10, pp. 1633–1649, 2002.

[8] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit, "A survey of models, analysis tools and compensation methods for the control of machines with friction," *Automatica*, vol. 30, no. 7, pp. 1083–1138, 1994.

[9] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[10] H. W. Leonard, "Volts vs. ohms; speed regulation of electric motors," *Transactions of the American Institute of Electrical Engineers*, vol. XIII, pp. 373–386, 1896.

[11] C. Johnson, "A homopolar tachometer for servomechanism application," *Proceedings of the IRE*, vol. 40, no. 2, pp. 158–160, 1952.

[12] W. Leonhard, "30 years space vectors, 20 years field orientation, 10 years digital signal processing with controlled AC-drives, a review (part 1)," *EPE Journal*, vol. 1, no. 1, pp. 13–19, 1991.

[13] P. Pillay and R. Krishnan, "Modeling, simulation, and analysis of permanent-magnet motor drives. I. The permanent-magnet synchronous motor drive," *IEEE Transactions on Industry Applications*, vol. 25, no. 2, pp. 265–273, 1989.

[14] T. M. Jahns, "Motion control with permanent-magnet AC machines," *Proceedings of the IEEE*, vol. 82, no. 8, pp. 1241–1252, 1994.

[15] S. Morimoto, Y. Takeda, T. Hirasa, and K. Taniguchi, "Expansion of operating limits for permanent magnet motor by current vector control considering inverter capacity," *IEEE Transactions on Industry Applications*, vol. 26, no. 5, pp. 866–871, 1990.

[16] K. J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, no. 5, pp. 645–651, 1984.

[17] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the ASME*, vol. 64, no. 8, pp. 759–765, 1942.

[18] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Engineering Practice*, vol. 13, no. 2, pp. 145–157, 2005.

[19] G. W. Vogl, M. A. Donmez, A. Archenti, and B. A. Weiss, "Inertial measurement unit for on-machine diagnostics of machine tool linear axes," in *Annual Conference of the PHM Society*, vol. 8, no. 1, 2016.

[20] S. Thomsen, N. Hoffmann, and F. W. Fuchs, "PI control, PI-based state space control, and model-based predictive control for drive systems with elastically coupled loads—a comparative study," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 8, pp. 3647–3657, 2011.

[21] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[22] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model-predictive control," *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[23] T. R. Kuphaldt, *Lessons in Industrial Instrumentation*, version 2.33 ed. Open textbook, Creative Commons Attribution 4.0 license, ibiblio.org/kuphaldt/socratic/sinst, 2024.

[24] M. Zhao, Y. Yuan, L. Luo, and X. Li, "A review: Absolute linear encoder measurement technology," *Sensors*, vol. 25, no. 19, p. 5997, 2025.

[25] D. Gurauskis, D. Marinkovic, D. Mažeika, and A. Kilikevičius, "Self-calibratable absolute modular rotary encoder: Development and experimental research," *Micromachines*, vol. 15, no. 9, p. 1130, 2024.

[26] F. M. Proctor and J. Michaloski, "Enhanced machine controller architecture overview," National Institute of Standards and Technology, Gaithersburg, MD, NIST Interagency/Internal Report 5331, 1993.

[27] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, "Guide to operational technology (OT) security," National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-82r3, 2023.

[28] H. Gupta and P. Kundu, "Digital twin development for feed drive systems condition monitoring and maintenance planning," in *PHM Society European Conference*, vol. 8, no. 1, 2024.

[29] C. Huang and H. Li, "Adaptive notch filter for piezo-actuated nanopositioning system via position and online estimate dual-mode," *Micromachines*, vol. 12, no. 12, p. 1525, 2021.

[30] X.-Y. Wang, J.-L. Xie, C.-G. Zhou, H.-X. Zhou, and C. Yin, "A feature-enhanced multi-kernel adaptation framework for cross-domain fault diagnosis of ball screw feed systems," *Sensors*, vol. 26, no. 19, p. 6291, 2026.

12.9 Exercises

12.1 ○ Concept check · 2 min

In case 7 the relay autotuner finds an ultimate gain of 0.641 A s/rad at 432.4 Hz, and the Ziegler–Nichols gains it sets overshoot the 600 rpm step by 82.2 %, against 29.7 % for the commissioned gains. Which explanation is right?

(a) The 2 A relay is too weak to excite the loop, so the measured ultimate gain is far too low

(b) The sampled peaks of the cycle overstate the ultimate gain, so the rule sets a gain that is too high

(c) The backward-difference speed estimate adds a delay that the relay test cannot see in the loop

(d) The relay finds the phase crossing at the coupling's mode, and the rule tunes the integral to that period

12.2 ○ Concept check · 1 min

In the graded move and return as shipped, the position loop on the motor encoder ends with the table 56.6 load counts from its target, while the motor encoder reads the motor on its target. Which change removes the cause of this error?

(a) Close the position loop on the load encoder, so that it measures the table itself
(b) Raise the position gain K_v from 30 to 80 s^{-1}, so that the loop corrects faster
(c) Add a position integral, so that the loop removes the error that remains
(d) Set backlash compensation to the whole 0.5° gap of the gear pair

12.3 Concept check · 2 min

A cutting force ripples on the table at 163 Hz, the anti-resonance of the axis. What do the speed gain and the 400 Hz notch do to the table's vibration?

(a) The notch removes it, because it cuts the loop's gain at every frequency of the mechanics
(b) Little, because the motor barely moves there and the torque the loop applies goes into the table
(c) A higher speed gain reduces it in proportion, because a stiffer loop holds the axis at every frequency
(d) The vibration grows without bound, because a force at the anti-resonance destabilizes the loop

12.4 Theory and calculation · 7 min

The axis has a rotor inertia $J_m = 1.0\,\mathrm{kg\,cm^2}$, a load inertia referred to the motor $J_l = 5.0\,\mathrm{kg\,cm^2}$, a coupling stiffness $k_s = 526\,\mathrm{N\,m/rad}$ and a coupling damping $c_s = 0.020\,\mathrm{N\,m\,s/rad}$.

(a) Find the anti-resonance, the resonance and the locked-load frequency. *Checkpoint:* 163.2, 399.9 and 365.0 Hz.

(b) Find the damping ratio of the mode and the factor by which the gain of the loop above the resonance exceeds that of a rigid axis. *Checkpoint:* 0.048 and 6.

(c) A softer coupling halves k_s. Predict the three frequencies and the damping ratio. *Checkpoint:* 115.4, 282.7 and 258.1 Hz, and 0.068.

12.5 Theory and calculation · 6 min

The motor has $p = 4$ pole pairs, a flux linkage $\lambda_m = 0.020\,\mathrm{Wb}$, $R = 0.5\,\Omega$ and $L = 1.0\,\mathrm{mH}$, a rated torque of 1.27 N m and a peak torque of 3.8 N m, on a 48 V bus.

(a) Find the torque constant and the currents for the rated and the peak torque. *Checkpoint:* 0.12 N m/A, 10.58 A and 31.67 A.

(b) Find the radius of the voltage circle and the motor speed at which the back-EMF alone fills it. *Checkpoint:* 27.71 V and 3308 rpm.

(c) Find the voltage needed at 3000 rpm with the rated current and $i_d = 0$, and say whether the drive can deliver the rated torque there. *Checkpoint:* 33.20 V, so it cannot without field weakening.

12.6 Theory and calculation · 6 min

In the graded case the position loop runs on the motor encoder. The screw has a lead of 10 mm, the load encoder has 16 bits per revolution, the gear pair has a whole gap of 0.5°, the coupling has $k_s = 526\,\mathrm{N\,m/rad}$, and a 300 N cut resists the return.

(a) Find the table travel of one load count and the half gap in load counts. *Checkpoint:* 0.1526 µm and 45.5 counts.

(b) Find the torque the cut puts on the screw, the twist it causes and the twist in load counts. *Checkpoint:* 0.477 N m, 0.908 mrad and 9.5 counts.

(c) Add the two terms and compare the sum with the measured final error of 56.6 counts. *Checkpoint:* 55.0 counts, 8.4 µm.

12.7 App challenge · 8 min

Load scenario 7, The relay autotuner, and change no setting. The relay runs from 0.2 s, and the speed reference steps from 300 to 600 rpm at 0.5 s. The model has no random terms, so no seed applies.

(a) Run to 0.6 s. Read the autotuner's note: the frequency of the cycle, the ultimate gain and the PI gains it sets. Compare the frequency with the resonance on the Bode view. *Checkpoint:* 432 Hz, $K_u = 0.641$, $K_p = 0.288$ and $\tau_I = 1.93$ ms, above the 401 Hz resonance.

(b) Export the CSV file. Find the peak of the column `w_motor_rpm` after 0.5 s and the overshoot as a share of the 300 rpm step. *Checkpoint:* About 845 rpm and 82 %, within 1 percentage point.

(c) Press Reset, set the Notch filter On, run to 0.6 s again and repeat (a) and (b). *Checkpoint:* 500 Hz, $K_u = 1.817$, $\tau_I = 1.67$ ms, and an overshoot of 58 %, within 1 percentage point.

12.8 App challenge · 6 min

Load scenario 6, S-curve feedforward. The table moves 100 mm on an S-curve at up to 300 mm/s with $K_v = 30\,\mathrm{s}^{-1}$, and the profile lasts 418.0 ms.

(a) Change no setting and run to 1.2 s. Read the Worst following error and Last move tiles, and compare the first with v/K_v. *Checkpoint:* 9.990 mm and in position 972 ms after the move began, against 10 mm.

(b) Press Reset, set Profile feedforward On and run to 1.2 s again. Read the same two tiles. *Checkpoint:* 0.035 mm and 761 ms.

12.9 App challenge · 9 min

Load the Challenge, Move, return and settle. The target is the table within two load counts of 0 by 1.02 s and staying there, with no fault, the current within 31.7 A and at least 6 dB of gain margin. Run each case the full 1.6 s.

(a) Run the case as shipped. Read the gain margin on the Bode view and, from the exported file, the column `x_table_m` at 1.6 s in load counts of 0.1526 µm. *Checkpoint:* 5.9 dB and 56.6 counts, target not met.

(b) Press Reset and set Position feedback to the load encoder, Profile feedforward On, the Notch filter On, $K_p = 0.80$, $\tau_I = 12$ ms and $K_v = 70\,\mathrm{s}^{-1}$. Run again and read the challenge result, the Peak current tile and the gain margin. *Checkpoint:* In position at 0.989 s, 22.4 A, 7.1 dB, target met.

(c) Press Reset, set Position feedback back to the motor encoder and run again. Read the final error. *Checkpoint:* 55.5 counts, target not met.

12.10 Application area · 4 min

A four-flute end mill cuts on the table. Its tooth-passing frequency is the spindle speed in revolutions per second times the number of flutes.

(a) Find the spindle speeds that put the tooth-passing frequency at the 425 Hz of case 8 and at the 163 Hz anti-resonance. *Checkpoint:* 6375 and 2445 rpm.

(b) For each, say whether the drive's tuning can reduce the vibration, and what the engineer can do if it cannot. *Checkpoint:* Yes at 425 Hz, no at 163 Hz.

12.11 Application area · 4 min

The drive protects the winding with the thermal image of Equation (12.5), with $\tau_{th} = 10$ s and a rated current of 10.58 A, starting cold.

(a) An axis is stalled against a hard stop with 31.6 A in the winding. Find the time until the image reaches 1 and the limit folds back. *Checkpoint:* About 1.19 s.

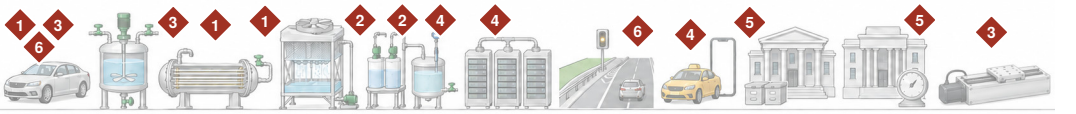
(b) Field weakening holds the rated torque at 3000 rpm with $i_q = 10.58$ A and $i_d = -8.65$ A. Find the value toward which the image settles and the time to the fold-back. *Checkpoint:* 1.67 and about 9.2 s.

12.12 Review the agent · 5 min

An AI assistant is asked to recover the speed loop of case 4 after the workpiece raises the inertia ratio to 5:1, and replies as follows. “Raise K_p from 0.4 to 1.2 with $\tau_I = 15\,\text{ms}$ and switch the 400 Hz notch on. The speed-loop card then reads a bandwidth of 55.5 Hz and a phase margin of 61.2°, as well damped as the 63.6° of the 1:1 tuning, so the loop is robust and also meets the 6 dB rule of the challenge.” Find the flaw in the proposal and say how it could be checked.

CHAPTER 13

Summary and Outlook



Open gaps 1 closed-loop tuning 2 loop monitoring 3 verified tunings 4 model-free scheduling 5 calendar loops 6 PID conformance

Notice that the car carries three gaps and that every gap reaches at least two units, so no gap belongs to one plant alone. Illustration, not to scale.

IN PLAIN TERMS **What Eleven Plants Taught About One Rule**

The main idea. Every chapter of this book used the same simple feedback rule, on a car, tanks, an internet router, a freeway ramp, prices and interest rates. Yet one controller settled quickly while another swung back and forth for an hour.

The challenge. The rule itself rarely decided the result. A handle stuck at its stop, a delay read wrong, a plant that changed from day to day or a slow calendar of decisions did.

Why it matters. Checking those four things before tuning avoids most of the trouble a controller can give.

Eleven chapters put one algorithm on eleven plants. The PID law of Section 1.6.2 ran unchanged in every app, and the results still ranged from a loop that settled in minutes to one that cycled for an hour or broke a freeway merge four times in a morning. This chapter sets those results side by side, draws a short guide for choosing a design, looks at where PID practice is heading, reports how the volume itself was made with AI agents and what they got wrong, and lists the problems that remain open.

13.1 What the Chapters Showed

Tables 13.1 and 13.2 compare the chapters. Each row names the difficulty the plant posed, a structure or setting that worked, one that failed, and the measured result with the table or section of the chapter that reports it. The algorithm is the same in every row. The rows differ in the tuning choices and the implementation details around it.

Figure 13.1 draws one pair of those results from each chapter. The columns of the two tables make five points.

The identification decided more than the rule. In Chapter 4 and Chapter 5 the same Ziegler–Nichols arithmetic produced a good loop and a bad one. The difference was a dead time read from the trend instead of from the fit. On the basin the misread loop matched the SIMC loop on IAE and spent 23 times the valve travel, because it was cycling. A rule that adds the dead time to τ_c, as SIMC does, is tolerant of that reading error by construction (Section 1.4.5).

Limits decide the result more often than gains. Chapter 3 raised the gain fourfold and left the offset at 0.1667, because the feed was on its bound. Chapter 2 and Chapter 3 owe their clean exits from saturation to back-calculation, and Chapter 11 owes a loss of 17.43 instead of 22.84 to tracking

Table 13.1 What Decided Each Result, Chapters 2 to 6

Chapter	Difficulty	What worked	What failed	Measured result
2, cruise control	Gain and time constant fall by factors of 3.26 and 3.09 with speed. A pedal stop.	Velocity form with back-calculation at $T_t = \tau_I$.	Back-calculation removed (a model run). A relay test that read $K_{cu} = 33.9$ where the plant has 469.	Grade IAE 18.29 m at three speeds (Table 2.6). Overshoot after 290 s on the stop 7.2 % against 32 % (Section 2.12).
3, blending tank	Feed concentration capped. Setpoint out of reach above 33.3 m^3/min of diluent.	Back-calculation: the feed leaves its bound 0.30 min after the upset clears, at either gain.	A fourfold gain against the bound. A relay-written integral time of 1.83 min on a 1.0 min tank.	Offset 0.1667 at $K_c = 2$ and 8 (Table 3.7). Relay raised the dilution IAE from 0.159 to 0.313 (Table 3.10).
4, steam exchanger	Hidden constants. Fitted dead time 0.94 min on a 0.30 min transport delay. Gain moves 2.6 times with load.	SIMC PI ($K_c = 4.5$) and Ziegler–Nichols PID ($K_c = 10.8$) from the same fit.	Ziegler–Nichols with the dead time read 0.44 min short ($K_c = 20.3$). It oscillates at a lower feed.	IAE 8.61, 3.37 and 3.43 °C min, valve travel 17.4, 24.8 and 112 % (Table 4.6).
5, cooling tower basin	Integrating level. Apparent dead time 0.83 min on a 0.50 min delay.	SIMC PI, $K_c = 18.1$, $\tau_I = 6.6$ min.	Ziegler–Nichols fed 0.50 min ($K_c = 56.4$). A reset time cut to 1.0 min at a fixed gain.	IAE 4.42 against 4.72, travel 20.47 against 474.7 % (Table 5.6). Settling 5.7 times longer (Table 5.3).
6, pH tank	Gain changes by a factor of 1.96×10^4 over one valve stroke.	Gain scheduling, or an inverted titration curve. An inner flow loop for header pressure.	A fixed gain tuned at 12 % of valve (a cycle from pH 4.50 to 10.97) or at the setpoint.	Back in permit 15.0 min after a load change when scheduled, 21.3 when linearized and never at the fixed gain (Table 6.4). Header upset 0.91 against 2.25 pH (Section 6.6.5).

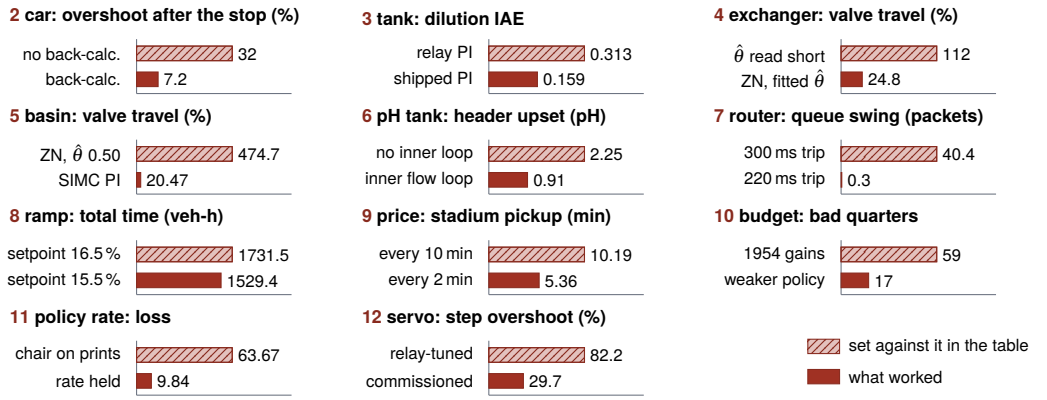


Figure 13.1 One pair of results from each chapter, with the numbers as the "Measured result" column of Tables 13.1 and 13.2 prints them. Each panel has its own scale from zero, and a smaller value is better in every panel. The pH tank's pair is its header upset, because its fixed gain never returned to the permit after the load change. Notice that in most panels the two bars share the plant and differ in one choice, such as a dead time, a limit, a period or a structure, while the router's pair shares its tuning and differs only in the path.

Table 13.2 What Decided Each Result, Chapters 7 to 12

Chapter	Difficulty	What worked	What failed	Measured result
7, router queue	The dead time is the round trip, which grows with the queue and the path. Direct action.	The PI at its design round trip of 220 ms.	The same tuning at 300 ms. A fast tuning moved from a 100 ms to a 264 ms path.	Phase margin 38.7, 16.9 and −1.3°, queue swing 0.3, 2.0 and 40.4 packets (Table 7.6).
8, ramp meter	The merge map folds past its critical occupancy. One detector report a minute.	Integral control (ALINEA) at 15.5 % occupancy.	A setpoint of 16.5 %, past the fold. A fixed plan on a heavier day.	Total time 1840.7 dark, 1529.4 metered, 1731.5 with four breakdowns and 2667.7 for the plan, in vehicle-hours (Table 8.7).
9, ride-hailing price	Throughput folds. The price moves on a calendar, in quanta.	A PI deciding every 2 min.	The same PI every 10 min. A lookup table that reverses itself.	Stadium peak pickup 10.23 min at the base fare (Table 9.5), 5.36 and 10.19 min for the two calendars (Section 9.6).
10, national budget	Recognition and implementation lags. One budget a year, a cap per decision.	$K_c = 0.5$, $\tau_I = 4$ yr, inside the annual gain limit.	The 1954 gains under the lags. A derivative term on noisy prints.	Bad quarters 59 for the 1954 gains, 17 for the weaker policy and 43 with the derivative (Table 10.5). Largest decaying gain 0.76 yearly against 1.53 monthly (Table 10.6).
11, policy rate	Lags of about two years, late prints, quarter-point steps, a floor.	The PID's integral against an unannounced fall in the neutral rate. Floor tracking.	A chair who chases the print. The PID on the 1970s challenge.	Final inflation 1.52 (rule) and 1.86 (PID) on a 2 % target. Loss 63.67 for the chair against 9.84 held (Table 11.5).
12, servo axis	A coupling resonance near 400 Hz caps the speed gain. Backlash, friction, a 5:1 load.	A notch, feedforward from the profile, the load encoder, no position integral.	A position integral against friction. Relay and Ziegler–Nichols tuned to the coupling.	Quiet gain 0.55 without the notch and 1.75 with it, following error 9.990 against 0.035 mm, step overshoot 82.2 against 29.7 % (Table 12.5).

at the floor of the policy rate. The first question on a loop that misbehaves is whether its actuator can deliver what is asked.

A moving gain needs a schedule or a new structure. The pH tank moved its gain far beyond any gain margin, and only a schedule or a linearized measurement recovered it. The router lost its phase margin to a longer path with no setting touched. The servo axis lost more than half its bandwidth at a 5:1 load. Where gain and time constant move together, as in Chapter 2 and Chapter 3, the ratio that sets the tuning stays fixed and there is little to schedule.

The calendar is part of the plant. An annual budget allows half the gain of a monthly one, and a ride-hailing PI deciding every 10 min lets the stadium into the wild goose chase. Faster is not always worth it. In Chapter 11 weekly decisions lowered the Taylor rule's boom loss only from 3.64 to 3.46, because two years of lag dominate any sample time.

An automatic tuning tool measures something, not always the loop. The relay of Chapter 2 read an ultimate gain of 33.9 where the sampled vehicle has 469, and its tuning still helped. The relay of

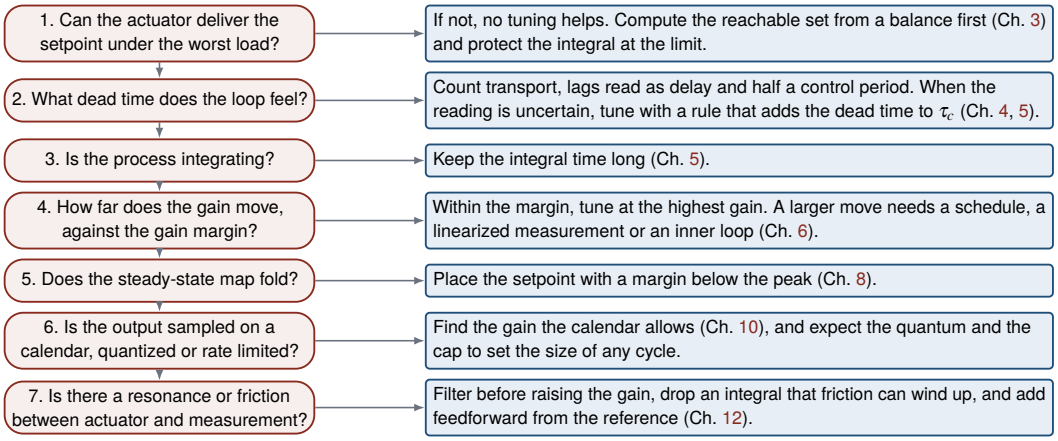


Figure 13.2 The seven questions of this section in order, with the list's answer and the chapters where it decided the result. Notice that the actuator and the dead time come first, and that the questions of calendars and resonances, which only some plants pose, come last.

Chapter 3 wrote an integral time almost twice the tank's residence time and doubled the IAE. The relay of Chapter 12 measured the coupling resonance correctly, and the rule applied to it produced 82.2 % overshoot. In each case the number was right and the assumption behind its use was not.

13.2 Choosing a Method

The chapters suggest an order of questions for a new loop. Each question names the chapter where its answer decided the result, and Figure 13.2 draws the same order.

1. *Can the actuator deliver the setpoint under the worst load?* If not, no tuning helps. Compute the reachable set from a balance first, as Chapter 3 does, and protect the integral at the limit.

2. *What dead time does the loop feel?* Count transport, lags read as delay, and half a control period (Section 1.4.11). Tune with a rule that adds the dead time to τ_c when the reading is uncertain, as Chapter 4 and Chapter 5 show.

3. *Is the process integrating?* Keep the integral time long. On the basin a reset time cut from 6.6 to 1.0 min multiplied the settling time by 5.7.

4. *How far does the gain move, against the gain margin?* A rise within the margin (a factor of π for SIMC with $\tau_c = \theta_p$) is met by tuning at the highest gain. A larger one needs a schedule, a linearized measurement or an inner loop (Chapter 6).

5. *Does the steady-state map fold?* Place the setpoint with a margin below the peak. In Chapter 8 15.5 % never broke down and 16.5 % broke down four times.

6. *Is the output sampled on a calendar, quantized or rate limited?* Find the gain the calendar allows (Table 10.6), and expect the quantum and the cap to set the size of any cycle.

7. *Is there a resonance or friction between the actuator and the measurement?* Filter before raising the gain, drop an integral that friction can wind up, and add feedforward from the reference (Chapter 12).

A predictive controller earned its place where it carried a model of the load or a preview. It cut the IAE of the exchanger's comparison case from 20.64 to 14.05 °C min (Table 4.7) and, at the suggested output weight, passed the inflation challenge that the shipped rule missed (Table 11.6). It lost where

the retuned PID already did the job. On the basin's unmeasured load step the Ziegler–Nichols PID kept the lower IAE, 1.04 against 1.58 (Table 5.7), and on the ramp meter's two-ramp comparison ALINEA cost 1459.917 vehicle-hours against 1469.389 for the predictive meter with its forecast on (Table 8.9). Volume 9 treats that choice in full.

13.3 Where Practice Is Heading

PID stays. In a 2018 survey of the IFAC Industry Committee, 66 of its 77 members answered, and 91 % said PID control had demonstrated high impact in practice, the highest share of any technology listed. For future impact, 78 % named PID and 85 % model predictive control [1]. A study of chemical process control in Japan stressed PID and conventional advanced control as keys to success [2]. In the editor's view the regulatory layer will remain PID for as long as plants are built from valves, pumps and drives, and the work will shift to keeping very many such loops healthy.

Loop health is monitored from routine data. A benchmark computed from routine operating data, without a step test, was proposed in 1989 [3], and an overview of the technology and its industrial applications followed in 2006 [4]. A 2016 survey of users found that many methods exist to detect malfunctioning loops, that their success in plants is not clear, and that organization, data availability and access can undermine them [5]. Plant-wide work traces disturbances to limit cycles, valve stiction, interactions and tuning problems, and points to joining data analysis with a qualitative model of the process [6]. The cooling tower of Chapter 5 is the case such monitors exist for: a cycling loop whose level moved less than two percent while its valve traveled 23 times as far as that of the SIMC loop.

> ⧖ HISTORY **1975 to Today: Smart Instruments and Self-Tuning Loops**
>
> The last history box of Chapter 1 ended in 1975 with the first distributed control system, still adjusting independent controllers. A history of control in a chemical engineering magazine carries the story on. Cheaper microprocessors then allowed redundancy and a greater flow of information to and from smart instruments and controllers. A protocol was developed that carried digital data on the same 4 to 20 mA wires as the analog signal, and the 1990s brought what were called the Fieldbus Wars, with makers pushing different digital systems and 20 industrial standards. Self-tuning controllers using fuzzy logic had already appeared in the 1980s [7]. In the editor's reading the arc ends where this volume begins: the same PID law, now a block in a networked system, with its tuning increasingly proposed by software.

Tuning moves to closed-loop data. Identification in closed loop has a body of theory that states when the data determine a model [8]. Direct methods design a controller from measured data without first fitting a process model [9]. Reinforcement learning has been demonstrated on physical hardware with a PID controller as the trained policy, so that the product is a tuning practitioners can read and deploy [10]. Language-model agents have been built that tune controllers iteratively against stated stability, performance and robustness requirements, and have been evaluated on a set of 500 design tasks [11]. In the editor's view the deployed law will stay a PID while the tuning behind it is increasingly proposed by software, which makes the checks of the next section the engineer's main contribution.

Anti-windup travels into the field devices. In a fieldbus loop with no separate controller, the input block must reside in the transmitter and the output block in the valve positioner, and the PID block may be placed in either device. The output block sends a back-calculation signal to the PID block to report whether the final element reached the state the PID commanded, which matters for reset windup when the element fails to respond [12]. That is the back-calculation of Chapter 2 and Chapter 3, running inside a valve. In the editor's view the details that decided this volume's results, the limit, the tracking and the transfer, travel with the block wherever it runs, so they have to be tested where it runs.

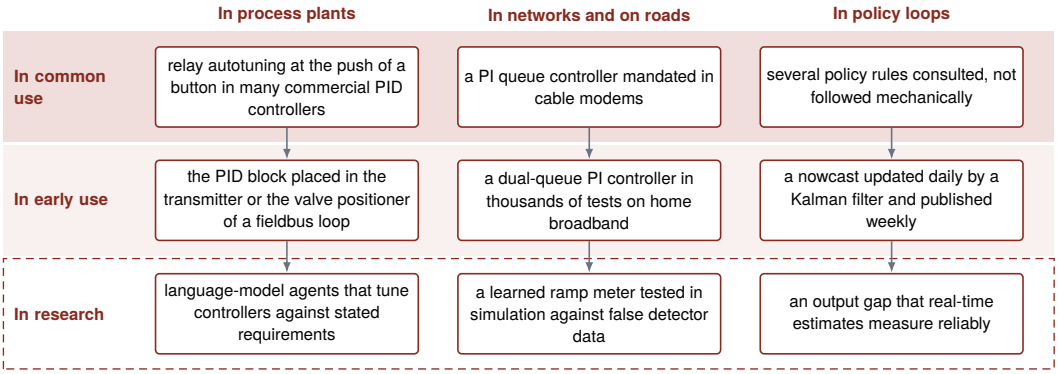


Figure 13.3 Developments in three threads. The top lane holds what the cited source calls standard, mandated or regular practice, the middle lane what it reports in service or in field tests, and the bottom lane what it reports from simulation, from a set of test tasks or as a need, in the editor's reading. Sources by thread: in process plants [11, 12, 18], in networks and on roads [13–15] and in policy loops [16, 17, 19]. Notice that each thread moves from a law applied as written toward one tuned, measured or checked in service.

PI control inside network equipment. A specification for cable modems mandated in October 2013 a variant of a proportional-integral queue controller as their active queue management. Its two gains are designed by feedback loop analysis and scaled with the present drop probability, and as a rule of thumb, halving its update interval calls for halving one gain and adjusting the other [13]. A dual-queue successor whose base is a PI controller was run in thousands of tests on home broadband, and kept the average queuing delay of every low-latency packet below 1 ms [14]. On roads, learned meters remain in research. A study of deep reinforcement learning for ramp metering names assumptions that are hard to match in the real world and rich inputs open to attack and noise, and tests one design in simulation against false detector data [15]. The rule for the update interval is the lesson of the ramp meter's period defect in Section 13.4.1.

Faster measurements for loops on a calendar. The policy loops of Chapter 10 and Chapter 11 decide on late and revised prints. One central bank's staff nowcast, a dynamic factor model in state-space form, is updated by a Kalman filter each day that new data are released, and its updates are published weekly [16]. Rules still inform rather than decide: the Federal Reserve's policymakers consult, but do not mechanically follow, several policy rules [17]. In the editor's view a timelier measurement shortens the recognition lag of such loops, while the decision calendar that limits their gain stays. Figure 13.3 sets developments of this section in three threads.

13.4 Engineering With AI Agents

An AI agent is a language model that works toward a goal in a loop. It plans, calls tools such as a compiler, a simulator or a browser, reads the results and continues [20]. In the editor's view, agents can now carry out much of the work that used to fill the hours of a control project. They derive a model from balances or fit one to plant data. They write the simulator and its tests, code the controller, tune it against stated criteria, script commissioning checks, write the documentation and watch running loops for degradation. The apps and chapters of this series were produced that way under the editor's direction, and the account that follows this section reports what the agents got wrong in this volume and how it was caught.

In the editor's view, this changes where an engineer's value lies. When a model, a controller and a report can each be produced in minutes, producing them is no longer the scarce skill. The work moves to five things.

Quality control. An agent's output arrives as a claim: this model fits, this tuning settles in four minutes, this code implements the equation. The engineer checks claims against the plant or the simulator. A reported number is measured again on a fresh run at the stated seed. The code is read for what it clamps, what it initializes and what it reads, because a display that shows the controller's model in place of the plant looks correct until the two differ. A tuning is tried on a disturbance the agent did not test. The most useful checks are independent of the author, made by a second agent or a person who has not seen the first answer, and they are checks that can fail.

Supervision. The engineer sets the scope of each task and decides what an agent may touch. Reading a historian and proposing a tuning is one level of access. Writing a setpoint, downloading a configuration or changing an interlock is another, and it passes through the plant's management of change with a named approver. Supervision also fixes the order of work: a baseline before a change, one change at a time and a comparison that shows what moved.

Creativity. An agent answers the question it is given. Choosing the question remains the engineer's task: which loop is worth improving, whether a disturbance calls for feedforward, a cascade or a change to the process, which step test the operation can tolerate, and which comparison would show that a predictive controller earns its place over a retuned PID loop. The experiment that separates two explanations is designed by someone who can imagine both.

Process insight. A model contains what its author put in. The equipment does more. A valve sticks, the dead time of an analyzer grows when its sample flow drops, a thermowell lags, a heat exchanger fouls and an operator intervenes at the same hour each night. The engineer who knows the unit recognizes a result that is too good, asks what the model left out and knows which measurement to distrust first.

Accountability. Responsibility does not transfer to a tool. Functional safety standards assign the activities of the safety lifecycle to persons and organizations of stated competence, and they require an assessment that includes someone independent of the design team [21]. An agent may draft the hazard analysis or the test procedure. A person owns the decision to accept it, to start up and to keep running.

These five are the professional form of the habit introduced in Section 1.7: check before trusting, and know enough to check.

13.4.1 How This Volume Was Made

Each of the eleven apps was built and tested by agents. An agent then wrote the paper on it, keeping verification notes that trace every number to a run, and a referee report was written on the paper, for some papers as a self-review by the agent that wrote it. Defects found along the way went into a defect log for each app with a severity, a cause, a suggested fix and a status. For this book, one agent per chapter reran every number that an exercise uses. The editor set the scope of each round and the rules on what each agent could change, and statements that runs contradicted were left for the editor to decide. The record holds the following errors and the checks that caught them.

The defect logs. The eleven logs hold 148 entries with a severity. Of these, 45 are wrong behavior of an app (a wrong output, a wrong number, a wrong lesson or a controller reading what it should not), 71 are documentation and 25 are cosmetic or display. Five cases show the range.

The velocity form lost its memory. In the first cruise control app, a setpoint change re-initialized the controller. In the velocity form the running output is the integral state, so the change discarded the pedal that held the car on a hill. Raising the setpoint from 25.0 to 25.5 m/s on a 4 % grade dropped the pedal by 6.016 %, and the car slowed when asked to speed up. The paper agent found it by driving the deployed app while measuring for the manuscript, and the fixed app now moves the pedal by K_c times the setpoint step and keeps the 9.146 % the hill needs.

The names of controller action were inverted. The first drafts of the series called a controller "direct acting" when the process gain was positive. That is the opposite of the convention a student is examined on. The referee found it and called it the most consequential defect it had found, because the signs in the code were right and only the words were wrong. The book round found two apps of this volume, Chapter 9 and Chapter 11, whose texts called a direct-acting loop reverse acting. Both apps were repaired, their chapters state the mapping once, and Section 1.4.3 tells the reader to trust the error equation and not the name.

A law that was right only at its default period. Writing the ramp meter's equation from its code showed that the per-minute gains mapped onto the shared PID correctly only at the default reporting period of one minute. At two minutes the meter released 940 vehicles an hour where its law gives 730. At five minutes one case spent 49.2 min in the capacity drop where the law gives none. The gains now scale with the period, and the fix left every shipped scenario bit for bit unchanged.

A number taken from the wrong run. A draft of the pH paper reported a peak of pH 10.89 for a feedforward experiment that had never been run. The figure belonged to a closed-loop run. The referee found it by tracing every number to a harness output. The first attempt to run the missing experiment was itself wrong, because it used a scenario whose timed events overwrote the manual dose.

Controllers that read the plant's truth. In two apps of this volume a controller took values it could not measure on a real plant. The ride-hailing optimizer rolled its plans through the run's hidden parameters and the true market. Of ten parameters scaled by 1.25, nine moved its plan, and six of those changes left every measurement unchanged. No shipped run showed it: 0 of 143 hashed runs moved when the leak was closed. The cruise control started its controller from the run's own balances, which are the plant's initial condition. Both log entries document the same probe: change a hidden value before Run and ask whether the controller moves while the measurements do not. This is the check the shared section calls reading code for what it reads.

The book round. The eleven chapter records list 220 rows of numeric verification, each a stated value set beside its reproduction. No chapter showed drift. Two prose statements were contradicted by runs and left for the editor. The router chapter said its flash-crowd peak was set by dead time "rather than by the tuning", yet the peak was 226.8 packets at the default gain, 197.1 with the gain doubled and 263.6 with it halved. The exchanger chapter said its two rules differed in sensitivity to the dead time, yet in the app both gains scale by the same factor of 1.88. Both chapters now state what the runs show. The agent writing Chapter 1 recorded three of its own statements, drafted from memory, that were wrong until computed.

The pattern is plain. Every error above was caught by a check that could fail: a second measurement, an equation written from the code, a number traced to its run, a hidden value moved to see what else moved. None was caught by rereading the text that contained it. Four working rules follow from this record. Re-measure every reported number. Read what a controller reads. Keep the defect log with a status for every entry. Let a person decide what changes.

13.5 Gaps in PID Practice

Six gaps belong to PID practice, and the gaps common to agent-era practice follow them in Section 13.5.1. In the editor's view none is closed. Each item gives the chapters' evidence, then the state of the literature with its sources. Why each gap is hard and what progress would look like are the editor's view. Figure 13.4 places the gaps by the chapters that met them and by the maturity of the method available, and the same numbers mark the plants in the picture that opens this chapter.

1. *Tuning from closed-loop data on a plant that will not be stepped.* The fitted constants of Chapter 4 belong to one operating load, the basin's dead time includes a valve lag, and the gain Chapter 2 finds depends on the size of the step. The Open challenge box after this list takes the gap further.

2. *Loop-performance monitoring at plant scale.* The basin's cycling loop scored an IAE close to the good one, and the pH tank's average sat inside the permit while it cycled from pH 4.50 to 10.97. A textbook of feedback notes that industrial perspectives on PID control all mention a significant fraction of poorly tuned controllers [18], and an open textbook of instrumentation warns that no degree of tuning will fix a poor control valve, a noisy transmitter or an ill-designed process [12]. The gap is hard because a loop's purpose, which decides whether its error or its travel matters, is not in its data. A monitor that knows each loop's purpose, weighs travel and oscillation as well as error, and runs across thousands of loops without false alarms remains an open engineering problem.

3. *Verification of agent-proposed tunings.* The relay results of Chapter 2, Chapter 3 and Chapter 12 show that a tool can return a correct number and a wrong tuning. Automatic tuning by relay feedback is used in many commercial PID controllers and is started by pushing a button, and the same textbook calls the relay auto-tuner a partial answer to the process engineers' dream of a controller that adjusts itself [18]. A survey of predictive control found adaptive and on-demand tuning PID controllers very successful in the marketplace [22]. The gap is hard because a tool's number can be checked only against a second description of the plant, which is seldom at hand. Before a proposed tuning is downloaded it needs a check that does not share the tool's assumptions: a margin computed on an independent model, a test on a disturbance the tool did not see, and a statement of the operating range it covers.

4. *Gain scheduling without a model.* The schedule of Chapter 6 follows an assumed titration curve, the surge price meets a plant gain that changes by more than an order of magnitude near the fold, and the router of Chapter 7 lost its phase margin to a longer path with no setting touched. Gain scheduling is widely used and is still the dominant scheme in flight control, one reason being developments in the air-data sensors that measure the scheduling variable [18]. A queue controller mandated for cable modems scales its two gains with its present drop probability, and its specification lists autotuning for the average round-trip time as an area for further study [13]. The gap is hard because the loop must stay stable while the schedule it is learning is still wrong. A schedule learned safely from the loop's own response, with a guarantee during the learning, is not available.

5. *PID on calendars and in quanta.* The fiscal and monetary loops decide on a calendar, in fixed steps, under caps, and their cycles are set by the quantum and the cap rather than by the gains. In practice such rules inform rather than decide. The Federal Reserve's policymakers consult, but do not mechanically follow, policy rules, and following one strictly would first require choosing the measures of inflation and of activity and the neutral rate [17]. Ex post revisions of the output gap are of the same order of magnitude as the gap itself, and real-time estimates tend to be severely biased around turning points [19]. The gap is hard because such a plant cannot be stepped and its lags are known only from its history. Design rules for such loops are thinner than for loops sampled fast, and progress would be rules that set the gain from the period, the quantum and the cap together.

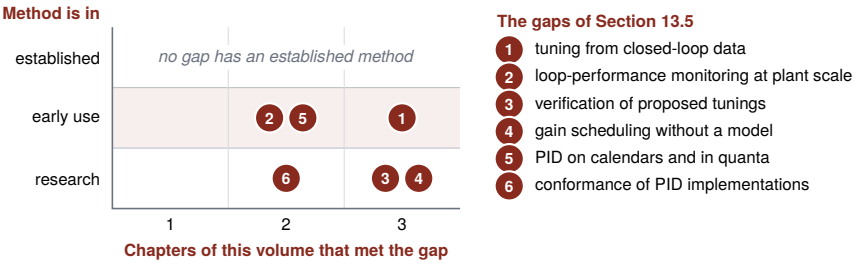


Figure 13.4 The six gaps of this section. Across is the number of chapters that each item names as evidence: three for gap 1 (Chapters 2, 4 and 5), two for gap 2 (5 and 6), three for gap 3 (2, 3 and 12), three for gap 4 (6, 7 and 9), two for gap 5 (10 and 11) and two for gap 6 (2 and 8). Up is the maturity of the method available, the editor's reading of the sources cited in each item and in the Open challenge box. Notice that the gaps about tuning from data, monitoring and calendar loops have methods in early use, while checking a proposed tuning, learning a schedule and testing an implementation are still research.

6. *Conformance of PID implementations.* The velocity form of Chapter 2 keeps the integral through a setpoint change, and the ramp meter of Chapter 8 matches its published law away from the limits at reporting periods of 1 and 2 min, which the application's checks confirm. Section 13.4.1 tells how both were once details of implementation, invisible on the shipped runs. An open textbook counts no fewer than three forms of the PID equation in modern controllers, notes that some controllers offer more than one, and shows that the position and velocity algorithms differ in how they react to a sudden change of gain [12]. A queue controller's specification ties its gains to its update interval by a rule of thumb [13]. The gap is hard because such differences show only in transfer, at limits and under changes of gain or period, which routine runs seldom exercise. A shared test suite that any vendor's or agent's PID block must pass, covering transfer, limits, gain changes and period changes, would catch such defects before a plant does.

⚠ OPEN CHALLENGE Tuning a Plant That Will Not Be Stepped

Three chapters name gap 1, and in each the constants of a tuning came from a test that a running plant resists. A survey of industrial model predictive control lists the limits of the identification behind it. Plant test signals are typically step-like and need the close attention of experienced engineers. No tool tells whether the collected data are adequate, so the plant is generally over-tested, and closed-loop identification of several variables is little practiced [22]. In the editor's view the gap is hard because normal operation seldom moves a loop enough to separate the plant from its controller. Progress would be a routine that tunes from normal records and states which constant those records could not determine.

13.5.1 Gaps Common to Agent-Era Practice

The final chapter of each volume lists gaps specific to its topic. Seven more are common to every topic in the series. In the editor's view none of them is closed.

1. *Verifying agent-written control code and models before they reach a plant.* Tests show that a controller behaves on the cases tested. For controllers learned from data, guarantees of stability and constraint satisfaction rest on assumptions about the model and its uncertainty, and open problems remain [23]. In the editor's view, code written by an agent has no comparable body of method yet. Practice needs a routine path from an agent's draft to code a plant can trust: independent derivation, tests of what the code reads and clamps, and trials with hardware in the loop.

2. *Trustworthy models from sparse or poor plant data.* Plant records are short, taken in closed loop, noisy and marked by unmeasured disturbances. A fitting routine returns a model from any record. What is missing is a dependable statement, delivered with the model, of where it can be trusted, and a method that says so when the data cannot identify a parameter.

3. *Safety cases and standards for agents in commissioning and operation.* Functional safety standards define a lifecycle of specified requirements, verification and assigned human roles [21]. In the editor's view, how an agent that writes configuration, runs commissioning checks or advises an operator is qualified, limited and audited under them is not yet settled.

4. *Security of agent access to control systems.* An agent with credentials to a control system is a new path into it. Industrial security standards state the requirements a control system must meet at each security level [24]. An agent that reads documents, messages or web pages can also be steered by text placed there. In the editor's view, least privilege, read-only access by default and human approval of every write are a starting point and not a solution.

5. *Benchmarks and reproducibility.* An agent asked twice may deliver two different controllers, and a result reported without its plant, seed and scoring cannot be checked. Control needs shared benchmark plants with fixed disturbances and published measures, on which agent-built and conventional designs are compared like for like. The apps of this series, with a default seed and an exported record of every run, are a start.

6. *Transfer from simulation to hardware.* A controller developed on a simulator inherits what the simulator omits. The robotics literature documents the gap between simulated and real performance and the methods used to narrow it, such as randomizing the simulated plant and adapting on the real one [25]. Process plants add slow dynamics, rare upsets and little freedom to experiment, so each transfer still needs a staged plan from the simulator to a bench rig to the unit.

7. *Keeping engineers' judgment sharp when agents do the routine work.* Automation removes the routine work through which skill is built and leaves the person the hard cases, an irony described for process operators in 1983 [26]. The same now applies to engineers. An engineer who has never tuned a loop by hand or closed a balance on paper is poorly placed to review an agent that

has. Deliberate practice, of the kind the exercises in this book provide, has to be designed into education and into work.

KEY POINTS

- The same rule ran in every chapter, and the results differed in the tuning choices and the details around it.
- The dead time read from a test decided more than the tuning rule applied to it.
- Limits decided the result more often than gains, so the first question is whether the actuator can deliver what is asked.
- A gain that moves needs a schedule or a new structure, and the decision calendar is part of the plant.
- An automatic tuning tool can return a correct number and a wrong tuning when the assumption behind its use fails.

References

[1] T. Samad, M. Bauer, S. Bortoff, S. Di Cairano, L. Fagiano, P. F. Odgaard, R. R. Rhinehart, R. Sánchez-Peña, A. Serbezov, F. Ankersen, P. Goupil, B. Grosman, M. Heertjes, I. Mareels, and R. Sosseh, "Industry engagement with control research: Perspective and messages," *Annual Reviews in Control*, vol. 49, pp. 1–14, 2020.

[2] M. Kano and M. Ogawa, "The state of the art in chemical process control in Japan: Good practice and questionnaire survey," *Journal of Process Control*, vol. 20, no. 9, pp. 969–982, 2010.

[3] T. J. Harris, "Assessment of control loop performance," *The Canadian Journal of Chemical Engineering*, vol. 67, no. 5, pp. 856–861, 1989.

[4] M. Jelali, "An overview of control performance assessment technology and industrial applications," *Control Engineering Practice*, vol. 14, no. 5, pp. 441–466, 2006.

[5] M. Bauer, A. Horch, L. Xie, M. Jelali, and N. Thornhill, "The current state of control loop performance monitoring – a survey of application in industry," *Journal of Process Control*, vol. 38, pp. 1–10, 2016.

[6] N. F. Thornhill and A. Horch, "Advances and new directions in plant-wide disturbance detection and diagnosis," *Control Engineering Practice*, vol. 15, no. 10, pp. 1196–1206, 2007.

[7] M. Pitt, "Taking a look back at control: Part 2," *The Chemical Engineer*, no. 989, Nov. 2023, thechemicalengineer.com/features/taking-a-look-back-at-control-part-2.

[8] U. Forssell and L. Ljung, "Closed-loop identification revisited," *Automatica*, vol. 35, no. 7, pp. 1215–1241, 1999.

[9] M. C. Campi, A. Lecchini, and S. M. Savaresi, "Virtual reference feedback tuning: A direct method for the design of feedback controllers," *Automatica*, vol. 38, no. 8, pp. 1337–1346, 2002.

[10] N. P. Lawrence, M. G. Forbes, P. D. Loewen, D. G. McClement, J. U. Backström, and R. B. Gopaluni, "Deep reinforcement learning with shallow controllers: An experimental application to PID tuning," *Control Engineering Practice*, vol. 121, p. 105046, 2022.

[11] X. Guo, D. Keivan, U. Syed, L. Qin, H. Zhang, G. Dullerud, P. Seiler, and B. Hu, "ControlAgent: Automating control system design via novel integration of LLM agents and domain expertise," arXiv preprint arXiv:2410.19811, 2024.

[12] T. R. Kuphaldt, *Lessons in Industrial Instrumentation*, version 2.33 ed. Open textbook, Creative Commons Attribution 4.0 license, ibiblio.org/kuphaldt/socratic/sinst, 2024.

[13] R. Pan, P. Natarajan, F. Baker, and G. White, "Proportional integral controller enhanced (PIE): A lightweight control scheme to address the bufferbloat problem," RFC Editor, RFC 8033, 2017.

[14] K. De Schepper, B. Briscoe, and G. White, "Dual-queue coupled active queue management (AQM) for low latency, low loss, and scalable throughput (L4S)," RFC Editor, RFC 9332, 2023.

[15] D. Liu, L. Liu, and L. D. Han, "Analyzing robustness of the deep reinforcement learning algorithm in ramp metering applications considering false data injection attack and defense," arXiv preprint arXiv:2301.12036, 2023.

[16] B. Bok, D. Caratelli, D. Giannone, A. Sbordone, and A. Tambalotti, "Macroeconomic nowcasting and forecasting with big data," Federal Reserve Bank of New York, Staff Report 830, 2017. [Online]. Available: https://www.newyorkfed.org/medialibrary/media/research/staff_reports/sr830.pdf

[17] "Policy rules and how policymakers use them," Board of Governors of the Federal Reserve System, Monetary Policy Principles and Practice, 2026, accessed 2026-11-20. [Online]. Available: https://www.federalreserve.gov/monetarypolicy/policy-rules-and-how-policymakers-use-them.htm

[18] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed. Princeton, NJ: Princeton University Press, 2021.

[19] A. Orphanides and S. van Norden, "The reliability of output gap estimates in real time," Board of Governors of the Federal Reserve System, Finance and Economics Discussion Series 1999-38, 1999. [Online]. Available: https://www.federalreserve.gov/econres/feds/the-reliability-of-output-gap-estimates-in-real-time.htm

[20] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[21] International Electrotechnical Commission, "Functional safety – Safety instrumented systems for the process industry sector – Part 1: Framework, definitions, system, hardware and application programming requirements," IEC 61511-1:2016, Edition 2.0, 2016.

[22] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[23] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[24] International Electrotechnical Commission, "Industrial communication networks – Network and system security – Part 3-3: System security requirements and security levels," IEC 62443-3-3:2013, Edition 1.0, 2013.

[25] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: A survey," in *Proc. 2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 737–744.

[26] L. Bainbridge, "Ironies of automation," *Automatica*, vol. 19, no. 6, pp. 775–779, 1983.

13.6 Exercises

13.1 Discussion · 8 min

In Chapter 4 and Chapter 5 the same Ziegler–Nichols rule produced one good loop and one poor loop from the same plant. An agent hands you a tuning for a new integrating level loop, computed with that rule from a dead time it read off a trend. List what you would check before the tuning is downloaded, and say which check would have caught the basin's poor loop.

13.2 Discussion · 8 min

The relay tests of Chapter 2, Chapter 3 and Chapter 12 each returned a measured number. Compare what each relay measured with what the loop needed, and explain why "the relay result reproduced" is not enough to accept the tuning it writes.

13.3 Discussion · 7 min

A plant asks for a monitor that flags poorly performing loops among several thousand. Using the cooling tower basin and the pH tank, explain why a monitor built on the IAE or on the average of the PV would miss loops that need attention, and propose three measures it should compute instead.

13.4 Discussion · 8 min

A finance ministry proposes to decide its budget every month instead of once a year, and a central bank proposes to meet weekly instead of eight times a year. Using Chapter 10, Chapter 11 and Chapter 9, argue when a faster calendar is worth its cost and when it is not.

13.5 Discussion · 10 min

Section 13.4.1 lists five errors made by the agents that built this volume's apps and papers. For each, name the check that caught it. Then write a procedure, in five to eight steps, for an agent that will

retune a PID loop on a running plant, stating what the agent may read, what it may write and what a person must approve.

13.6 Discussion · 7 min

The pH tank of Chapter 6 was rescued by a gain schedule that follows an assumed titration curve. Discuss what would go wrong if the curve were learned instead from the loop's own operation, and what would have to be true for such a learned schedule to be safe. Use the buffering case and the ride-hailing fold in your answer.

Answers to Selected Exercises

Short answers are given for the odd-numbered exercises. Numeric answers from an app are for its default seed and settings unless the exercise says otherwise.

Chapter 1. Introduction

1.1 (c). The process gain is negative, so $aK_cK_p > 0$ needs $a = -1$, and an output that rises with the PV is direct action.

1.3 $K_p = 1.5\,^\circ\text{C}/\%$, $\tau_p = 7.5\,\text{min}$, $\theta_p = 1.5\,\text{min}$.

1.5 (b). An integrating process with a PI controller has two integrators in the loop, and a short integral time removes the phase lead that keeps it stable.

1.7 (a) 30 % of output. (b) 48 min.

1.9 (a) About 54°, down from 61.4°. (b) $K_c = 1.0\,\%/^\circ\text{C}$, $\tau_I = 10\,\text{min}$.

Chapter 2. Cruise Control With Velocity-Form PID

2.1 (d). Both constants carry the factor $1/(2k\bar{v}_r)$ of Equation (2.6), and their ratio is F_p/M.

2.3 (b). The fitted gain is a secant across a large excursion on a curve whose slope falls with speed.

2.5 (a) $d = 342.8\,\text{N}$ and $d/F_p = 11.43\,\%$. (b) 22.85 m, independent of the cruise speed. (c) About 1.58 m/s.

2.7 (a) 36.56 m/s. (b) 37.96 m/s and 0.648 m/(s %). (c) 14.1 s. Both are below the tangent values of 0.816 and 19.05 s because the step is large.

2.9 (a) 4.25 m, 24.89 m/s. (b) 4.87 m, 24.58 m/s. (c) 23.59 m, 23.81 m/s. The estimate removes most of the error, and the map removes most of the remaining dip.

2.11 (a) 459.4 N and 11.5 kW at 25 m/s, 661.5 N and 19.8 kW at 30 m/s. (b) 45.9 and 66.2 MJ, which is 44 % more energy for 11.1 min saved.

Chapter 3. Blending Tank Control at an Actuator Limit

3.1 (c). Equation (3.5) gives $K_c = V/(q\tau_c)$, which holds no dilution term, and τ_I follows the residence time.

3.3 (a). Back-calculation holds $u_b + I$ at 2.0000 mol/L, and the 33 m^3/min upset gives a peak of 1.6253 mol/L without touching the bound.

3.5 (a) 33.3 m^3/min. (b) A ceiling of 1.429 mol/L and an error of 0.071 mol/L. (c) A feed flow of 150 m^3/min or more, which shortens the residence time to 0.50 min.

3.7 (a) 1.897 mol/L, a gain of 0.99 that is still rising to 1.00. (b) About 1.0 min. (c) The outlet moves from 1.000 to 1.267 mol/L, a gain of 0.67, with a time constant of about 0.66 min. Gain and time constant fall together.

3.9 (a) 0.207, 1.497 and 1.639 mol/L. (b) 0.281, 1.434 and 1.629 mol/L. The ceiling is 1.667 mol/L. The MPC wins while it rejects the dilution, before the bound is reached, and gains nothing on the bound.

3.11 (a) $K_c = 1.0$ and $\tau_I = 1.0\,\text{min}$, half the default gain. (b) 0.25 min. A tuning found on the app is optimistic by the delay the plant carries, so the delay must be measured before the gain is used.

Chapter 4. Steam-Heated Exchanger: Step Tests and Tuning

4.1 (b). Every structure ends at FCV-102, and only the calculation that positions it changes.

4.3 (d). A working loop hides the fouling from the controlled variable and shows it in the controller output.

4.5 (a) 4.5 %/°C, 3.0 min. (b) 10.8 %/°C, 1.88 min, 0.47 min. (c) SIMC 8.5 %/°C and 3.0 min. Reaction curve 20.3 %/°C, 1.0 min and 0.25 min. Both gains rise by 1.88, and only the reaction-curve rule also shortens its times.

4.7 (a) 0.354 °C/%. (b) $t_{28} = 2.00$ min and $t_{63} = 3.96$ min, so $\hat{\tau} = 2.94$ min and $\hat{\theta} = 1.02$ min. (c) The least-squares fit reads 0.355, 3.02 and 0.94 min. Both dead times are about three times the transport delay.

4.9 (a) 79.82 °C with 1411 kW required and 1248 kW available. (b) The valve leaves its stop at 22.66 min and TT-102 peaks at 86.85 °C. (c) Without anti-windup the valve is still at 100 % at 28 min with TT-102 at 87.70 °C, because the integral term stands at 254.6 % when the cleaning comes, against 100.0 %.

4.11 (a) 65 % and 43.2 %. (b) FIC-102 holds the valve open and its own anti-windup holds its integral, but TIC-101 is not at its own limit and keeps integrating. The outer output limit is set at the flow the header can deliver, or the inner saturation is fed back to the outer loop.

Chapter 5. Cooling Tower Makeup: Integrating Loop Tuning

5.1 (a). Evaporation, blowdown, drift and makeup are all set by something other than the level, so a constant imbalance gives a constant slope.

5.3 (c). Integral action is needed for the load offset, its time must be long on a plant that already integrates, and derivative action bought 6.6 % less IAE for 7.1 times the valve travel.

5.5 (a) For large t the bracket tends to $t - \theta - \tau_v$, which is zero at $t = \theta + \tau_v = 0.83$ min. (b) $K_I \Delta u \, \tau_v / \mathrm{e} = 0.040$ %. (c) About 0.64 min, and 0.65 min without the approximation.

5.7 (a) 0.333 %/min, so $\hat{K}_I = 0.0333$ %/min per percent. (b) 0.83 min. (c) 0.89 min. (d) $K_c = 18.1$, $\tau_I = 6.6$ min, $\tau_D = 0$, and a Model check of excellent.

5.9 (a) 14.06 % min, 65.03 %, 144 %. (b) 12.95 % min, 65.31 %, about 2700 %. (c) 18.02 % min, 66.31 %, 150 %. The MPC with the shipped move suppression is the design to install. The lowest IAE belongs to a valve that strokes from stop to stop.

5.11 (a) The valve travel. The basin integrates the swing of the valve, so the level moves little. (b) An aggressive tuning from a dead time read too short, or valve stiction. Halve the gain and lengthen the reset time: a tuning cycle dies out, and a stiction cycle stays.

Chapter 6. pH Neutralization and Gain Scheduling

6.1 (a). The states respond alike in both tests, and Equation (6.4) converts the same change in W_a into a pH change that depends on where the tank sits.

6.3 (c). An inner loop rejects what appears in its own measurement, and the flow transmitter sees the pressure drop and not the concentration.

6.5 (a) $J = 35.7$. (b) -1.215×10^5 pH per mol/L, or -0.12 pH per µmol/L. (c) 33.8 pH per percent. (d) $J = 554$ and 2.14 pH per percent, lower by a factor of 15.8.

6.7 (a) pH 2.790 and 2.872, a gain of 0.082 pH per percent. (b) 5.3 min. (c) pH 7.003 and 8.701, a gain of 56.6 pH per percent, about 690 times larger. W_a moves by 0.28 mmol/L per percent of valve in both tests.

6.9 (a) pH 5.71, 50.36 %, 65.0 min off permit. (b) pH 7.00, 53.13 %, 12.6 min. (c) 2.2 min, from a limit cycle on the clean stream while the controller's curve assumes a buffered one.

6.11 (a) 235.7 min and 0.98. (b) A lag of 1.96 min, an offset of -0.88 pH and a true pH of 7.88. (c) The true pH is still inside the permit, and the loop holds its own reading on the setpoint.

Chapter 7. Delay and Stability in TCP Congestion Control

7.1 (d). A longer round trip subtracts phase and adds gain at the same time, and crossover moves from 2.09 to 3.02 rad/s.

7.3 (b). The steady gain $R_0^3 C^3 / 4N^2$ is highest on the lightly loaded link.

7.5 (a) $a = 0.356/\mathrm{s}$, $b = 3.33/\mathrm{s}$, $|P(0)| = 32\,959$ packets, 4.63 times the default. (b) 51.9°. (c) About –1.2°.

7.7 (a) 2.99 % at the router while the senders still see 1.556 %, which first moves in the row at 6.25 s. (b) 226.8 packets at 7.20 s. (c) 11.8 % and 3.88 packets, on the way to 12.45 % and 3.75 packets.

7.9 (a) 172.5 packets, then 74.1 to 144.9 packets. (b) 214.7 packets, then 111.5 to 119.9 packets. (c) 9191 and 1471 packets. Only run (b) holds the band, because the plant gain is 6.25 times higher with 20 sessions than with 50.

7.11 (a) $11.8^\circ + 6.0^\circ = 17.8^\circ$, which leaves about 50°. (b) No. The crossover at 260 ms is higher, so the two take more than the 16.9° available.

Chapter 8. Ramp Metering Short of Freeway Breakdown

8.1 (c). The drop lowers the discharge to $(1-\delta)\lambda Q = 5400\,\text{veh/h}$, and the demand exceeds that until 8:40.

8.3 (a). The override releases the arrivals once the queue is at its limit, and the merge receives the traffic the meter was holding.

8.5 (a) $e(k+1) = (1 - K_R T/g)\,e(k)$, a factor of 0.810 and a time constant of 4.8 min. (b) $K_R = 369$ and $K_R = 738$, and 0.458 at $K_R = 200$. (c) $K_c = 70\,\text{veh/h}$ per percent with $\tau_I = 1\,\text{min}$, and $K_c = 120$ with $\tau_I = 1.71\,\text{min}$.

8.7 (a) 104.9 min and 1841 veh·h. (b) 104.5 min and 1841 veh·h. (c) 1405 veh·h. The lost tenth of capacity holds the merge in the drop. The width of the recovery margin moves the result by less than half a minute.

8.9 (a) 1766 veh·h, 89.1 min and 51 of 60. (b) With 105 vehicles the merge still breaks down, 1708 veh·h and 74.6 min. With 110 it does not, and the morning costs 1474 veh·h with a worst queue of 94. (c) The storage has to last only until the overload has passed. The margin is a hair, so the designer should not rely on it.

8.11 (a) Metering puts the delay on the drivers of the two ramps. The speed limits spread it over every mainline driver. (b) Metering with the override and coordination, which meets the target on all ten other seeds against one of ten for the speed limits. (c) Any two of fuel, emissions, reliability, and fairness between ramp and mainline drivers.

Chapter 9. Surge Pricing Feedback on a Throughput Fold

9.1 (b). A higher price shortens the pickup, so $K_p < 0$, and only $a = -1$ makes the loop gain aK_cK_p positive.

9.3 (d). Without preview a slow price cannot catch the crowd. A fast one reaches its price in two decisions after the release.

9.5 (a) 1/6. (b) 848.5 an hour and 894.2 drivers, within 0.1 % of the measured 894.81. (c) 1/4, a factor of 1.189. With $\eta_d = 1$ the exponent is 0.

9.7 (a) 848.5 an hour, \$35.70 an hour, 1126.6 drivers. (b) 913.2 drivers at 0.55 h. (c) 894.8 drivers, a ratio of 1.123 against $2^{1/6} = 1.122$.

9.9 (a) 100.0 %, 538 abandoned, a mean of 1.04, not met. (b) 222.6 before 17:00 and 178.3 during the rain. (c) 384 abandoned at a mean of 1.13, met, with 70.4 during the rain.

9.11 (a) The cap leaves 1 as the only legal multiplier. (b) It saves 2682 abandonments, about 80 %, but the cap binds within a few minutes of the first surge and the pickup still peaks above 8 min.

Chapter 10. Fiscal Stabilization, Budget Lags and Debt

10.1 (b). The stabilizers are proportional feedback with a multiplier of 1.51, so the gap settles at $-2 \times 1.51 = -3.02\,\%$.

10.3 (d). The derivative is taken on the published gap without a filter, so each quarter's fresh print error becomes a budget swing at the next October.

10.5 (a) 119.9 % and 201.0 %, and a surplus of 1.8 %. (b) 180 % and 105 %, with a primary surplus of 2.1 %. (c) $\rho_B > r - g$.

10.7 (a) 10 quarters and –6.23 %, with the recovery peaking at 0.97 % at year 5.42. (b) 21 quarters and –5.48 %, with a boom of 4.71 % at year 4.04. The stimulus works mostly in the recovery.

10.9 (a) 9, 1 and 99.5 %, not met. (b) 7, 1 and 98.3 %, not met. (c) 5, 1 and 98.9 %, met. The tax stabilizer acts at once, and the brake acts only through the annual budget.

10.11 (a) $10\,e^{-0.1} = 9.05$ points. (b) About 0.95 % of output a year. A maturity structure, or inflation, which the model omits.

Chapter 11. Sampled, Quantized Inflation Targeting

11.1 (a). The process gain is negative, so only $a = -1$ makes the loop gain aK_cK_p positive, and Chapter 1 calls that controller direct acting.

11.3 (c). The rate the economy needs is off the grid, so the rule alternates between its two neighbors, and the step sets the cycle.

11.5 (a) $c = 0.43$ points and 0.286 points a year. (b) 7.6 % and 2.4 %. (c) 2.8 %, −1.525 % and 6.075 %, which the grid rounds to 6.125 %.

11.7 (a) 4.24 and 2.95 %. (b) 4.85, 2.90 % and −1.33 %. (c) 3.93 and 2.88 %. The gap moves about a year before inflation, so it is the earlier warning of the same demand shock.

11.9 (a) 2.25 yr, 2.74 %, 17.43. (b) 3.00 yr, 3.23 %, 22.84, and a request of −2.79 % at year 3.5. (c) The damage of the crisis is done through the lags before the floor matters. The wound-up integral holds the rate at the floor for 0.75 years longer and the recovery overshoots.

11.11 (a) The economy lines out at $r^* + \pi^*$, which is 1.875 % with a 1 % target, so the same crisis uses up the room to cut a point sooner. (b) Asset purchases and forward guidance. An operating point chosen with margin from a valve or heater limit so that the loop keeps authority in an upset.

Chapter 12. Servo Axis Tuning Around a Coupling Resonance

12.1 (d). The relay measures the ultimate point correctly, but that point belongs to the coupling resonance and not to the rigid-body motion the integral has to correct.

12.3 (b). At the anti-resonance torque at the motor cannot move the motor, so the loop has almost nothing to act on.

12.5 (a) $K_t = 0.12\,\mathrm{N\,m/A}$, 10.58 A, 31.67 A. (b) 27.71 V and 3308 rpm. (c) 33.20 V, which is more than the circle holds.

12.7 (a) 432 Hz, $K_u = 0.641$, $K_p = 0.288$, $\tau_I = 1.93$ ms. (b) About 845 rpm, an overshoot of 82 %. (c) 500 Hz, $K_u = 1.817$, $K_p = 0.818$, $\tau_I = 1.67$ ms and about 58 %.

12.9 (a) 5.9 dB and 56.6 counts. (b) In position at 0.989 s with 22.4 A and 7.1 dB. (c) 55.5 counts.

12.11 (a) About 1.19 s. (b) The image settles toward 1.67, so it reaches 1 after about 9.2 s.

Chapter 13. Summary and Outlook

13.1 A good answer separates identification error from rule error. It asks how the dead time was obtained (fit or trend reading) and whether it includes valve lag and half a control period, recomputes the settings with a rule that adds the dead time to τ_c (SIMC) and compares, checks the ratio of the proposed gain to the ultimate gain, checks that the integral time is long for an integrating process, and asks for valve travel as well as IAE in the test. On the basin, travel (474.7 against 20.47 %) or a gain at 99 % of the ultimate gain would have caught it, while IAE alone would not.

13.3 A good answer cites the basin, where the cycling loop's IAE of 4.72 was close to the good loop's 4.42 while its valve travel was 23 times larger, and the pH tank, whose mean pH of 8.57 sat inside the permit while it cycled from 4.50 to 10.97. It proposes measures such as actuator travel per hour against a reference, detection of sustained oscillation in the MV or PV, time outside a band that reflects the loop's purpose, and time at a limit, and says that thresholds must come from each loop's purpose.

13.5 A good answer pairs the velocity-form reset with driving the deployed app while measuring, the inverted action names with an independent referee reading against the textbook convention, the ramp meter's period error with writing the equation from the code, the pH number with tracing every number to its run, and the controllers that read the truth with a probe that changes hidden values while measurements stay fixed. The procedure gives read-only historian access by default, a baseline before any change, one

change at a time, a check of the proposed tuning that does not share the tool's assumptions, a person's approval through management of change for any write, a rollback value, and a logged record of each change.

Glossary

Each term is given as this volume uses it, in one plain sentence, followed by the chapter where it is first used in earnest. The two indexes give every page.

active queue management Logic in a router that marks or drops packets before its buffer is full, to hold the queue and its delay low. *Chapter 7*.

actuator saturation The state in which an actuator is at a limit, so a larger command moves the process no further. *Chapter 1*.

adaptive control Control that adjusts its own settings while it runs, from what it learns about the process. *Chapter 6*.

ALINEA A feedback law for ramp meters that changes the release rate in proportion to the gap between measured and desired occupancy. *Chapter 8*.

anti-windup Logic that stops a controller's integral term from growing while the actuator is held at a limit. *Chapter 1*.

automatic stabilizers Taxes and benefits that rise and fall with the economy by themselves, damping its swings without any new decision. *Chapter 10*.

back-calculation An anti-windup method that feeds the gap between the applied and the computed output back into the integral term. *Chapter 1*.

backlash Free play in a gear or screw, so the driven part does not move until the drive has taken up the gap. *Chapter 12*.

blowdown Water drained from a cooling tower on purpose to carry away the minerals that evaporation leaves behind. *Chapter 5*.

budget calendar The fixed dates on which a government adopts its budget, so a fiscal decision can change only once a year. *Chapter 10*.

bufferbloat Long delays caused by oversized network buffers that stay full, so every packet waits behind a standing queue. *Chapter 7*.

bumpless transfer A switch between manual and automatic control that does not jolt the actuator. *Chapter 1*.

capacity drop The fall in the flow a freeway can carry once traffic at a merge has broken down into a queue. *Chapter 8*.

cascade control Two nested loops in which an outer controller sets the setpoint of a faster inner loop, such as a flow under a temperature. *Chapter 4*.

conditional integration An anti-windup method that stops the integral term while the output is at a limit and the error would push it further. *Chapter 1*.

congestion collapse A state of a network in which senders retransmit so much that buffers stay full and little useful data gets through. *Chapter 7*.

congestion window The number of packets a sender may have in flight before it must wait for acknowledgments. *Chapter 7*.

consumer price index A published measure of the prices of a fixed basket of goods and services bought by households. *Chapter 11*.

control valve A valve moved by an actuator to set the flow in a pipe in answer to a controller's signal. *Chapter 1*.

cooling tower A structure that cools water by letting part of it evaporate into a stream of air. *Chapter 5*.

current loop The fastest loop of a motor drive, which sets the winding current and so the motor torque. *Chapter 12*.

cycles of concentration How many times more concentrated the minerals in a cooling tower's water are than in its makeup water. *Chapter 5*.

dead time The time between a change at the controller's output and the first response of the measurement. *Chapter 1*.

debt brake A fiscal rule that caps spending at revenue adjusted for the business cycle and requires overruns to be made up later. *Chapter 10*.

decision calendar The set times at which a controller is allowed to change its output, such as every two minutes. *Chapter 9*.

derivative filter A small lag on the derivative term that keeps measurement noise from being amplified into large output moves. *Chapter 1*.

derivative kick A spike in the controller output when the derivative of a setpoint step is taken through the error. *Chapter 1*.

direct action Controller action in which the output rises as the measurement rises, used when the process gain is negative. *Chapter 1*.

distributed control system (DCS) A plant-wide control system of networked controllers and operator stations that runs the loops of a process site. *Chapter 1*.

droop The steady offset that proportional action alone leaves after a lasting disturbance. *Chapter 10*.

effective lower bound The level below which a

central bank cannot usefully cut its policy rate, close to zero. *Chapter 11*.

encoder A sensor that counts pulses as a shaft or table moves, giving its position. *Chapter 12*.

explicit congestion notification A mark set in a packet header by a router, telling the sender to slow down without dropping the packet. *Chapter 7*.

feedforward control Control action computed from a measured disturbance or a known setpoint change before any error appears. *Chapter 2*.

first-order-plus-dead-time model A process model with one gain, one time constant and one dead time, fitted to a step test. *Chapter 1*.

following error The distance by which a moving axis lags behind its commanded position. *Chapter 12*.

fouling A deposit that builds up on heat-transfer surfaces and slowly lowers the heat an exchanger can pass. *Chapter 4*.

fundamental diagram The relation between how dense traffic is and how much flow a road carries, with a peak at capacity. *Chapter 8*.

gain margin The factor by which the loop gain can rise before the loop starts to oscillate. *Chapter 1*.

gain scheduling Changing a controller's gain with a measured variable that tells where the process is operating. *Chapter 1*.

ideal form The PID form in which one gain multiplies the error, its integral divided by the integral time and its rate times the derivative time. *Chapter 1*.

implementation lag The delay between a policy decision and the time its spending or taxes reach the economy. *Chapter 10*.

inflation targeting A monetary policy framework in which a central bank publishes a numerical inflation goal and sets its rate to reach it. *Chapter 11*.

integral control Control that moves the output by an amount proportional to the accumulated error, so no steady error remains. *Chapter 8*.

integrating process A process whose output keeps ramping after a step, such as a tank level, because nothing restores it. *Chapter 1*.

integrator windup Growth of a controller's integral term while the actuator sits at a limit, which later causes overshoot. *Chapter 1*.

internal model control (IMC) A tuning method that chooses the desired closed-loop response and derives the controller settings from a process model. *Chapter 1*.

inverter The power stage of a drive that switches a direct-current bus to make the voltages the motor needs. *Chapter 12*.

Kalman filter An estimator that blends a model's prediction with noisy measurements, weighting each by its uncertainty. *Chapter 1*.

limit cycle A steady oscillation that a nonlinear loop settles into, of fixed size and period. *Chapter 6*.

live zero A signal range that starts above zero, such as 4 to 20 milliamperes, so a dead signal can be detected. *Chapter 1*.

loop detector A wire loop cut into the road surface that senses the metal of a passing or waiting vehicle. *Chapter 8*.

makeup water Fresh water added to a cooling tower basin to replace what evaporates, drifts away and is drained. *Chapter 5*.

manipulated variable (MV) The quantity a controller moves, such as a valve opening, a price or a policy rate. *Chapter 1*.

marking probability The share of arriving packets that a router marks or drops to signal congestion. *Chapter 7*.

measurement noise Random error added to a reading, which the controller cannot tell apart from real change. *Chapter 2*.

mechanical resonance A frequency at which a flexible machine part vibrates strongly, limiting how high a loop gain can go. *Chapter 12*.

model predictive control (MPC) Control that uses a model to predict the process over a horizon and picks the moves that best meet targets and limits. *Chapter 1*.

notch filter A filter that removes a narrow band of frequencies, used to keep a loop from exciting a resonance. *Chapter 12*.

occupancy The share of time a road detector is covered by a vehicle, a measure of how dense traffic is. *Chapter 8*.

offset The steady error that proportional control alone leaves after a setpoint or load change. *Chapter 1*.

output gap The difference between what an economy produces and what it could produce at normal use of its resources. *Chapter 11*.

parallel form The PID form in which the proportional, integral and derivative terms each have their own gain. *Chapter 1*.

pH electrode A glass probe whose voltage changes with the acidity of the liquid around it. *Chapter 6*.

phase margin The extra phase lag a loop can take at its crossover frequency before it becomes unstable. *Chapter 1*.

pickup time The time a rider waits for an assigned driver to arrive. *Chapter 9*.

PID control Feedback that moves the output in proportion to the error, to its accumulated sum and to its rate of change. *Chapter 1*.

policy rate The short-term interest rate that a central bank sets to steer the economy. *Chapter 11*.

position form A digital PID that computes the whole output at each execution, with a stored bias and a running sum of the error. *Chapter 1*.

positioner A small controller on a valve that compares the commanded and actual stem positions and moves the stem to agree. *Chapter 1*.

process variable (PV) The measured quantity a loop controls, such as a temperature, a level or a queue length. *Chapter 1*.

ramp metering Traffic signals on freeway on-ramps that release vehicles at a set rate to keep the main road flowing. *Chapter 8*.

rate limit A cap on how fast an actuator or an output may change from one moment to the next. *Chapter 6*.

ratio control Control that holds one flow in a set proportion to another flow. *Chapter 3*.

recognition lag The delay before policymakers learn from published data that the economy has changed. *Chapter 10*.

relay autotuning Finding a loop's ultimate gain and period by switching the output between two levels and timing the cycle that results. *Chapter 1*.

residence time The average time a portion of liquid spends in a tank, its volume divided by the flow through it. *Chapter 3*.

resistance temperature detector A temperature sensor whose metal wire, usually platinum, changes resistance with temperature. *Chapter 4*.

reverse action Controller action in which the output falls as the measurement rises, used when the process gain is positive. *Chapter 1*.

round-trip time The time for a packet to reach its destination and for its acknowledgment to come back. *Chapter 7*.

sample time The interval between two executions of a digital controller, during which its output is held. *Chapter 1*.

scenario A saved starting state of an app, with its settings and events, that a reader loads to run one case. *Chapter 1*.

series form The PID form in which a proportional-integral part is multiplied by a derivative part, so the settings interact. *Chapter 1*.

servo drive The electronic unit that powers a motor and runs its current, speed and position loops. *Chapter 12*.

setpoint (SP) The value a controller tries to hold the measured variable at. *Chapter 1*.

SIMC rule A simple model-based tuning rule that sets the gain from the model and shortens the integral time of slow processes. *Chapter 1*.

Smith predictor A controller structure that uses a model of the dead time so the controller can act on a prediction of the response. *Chapter 4*.

split range One controller output divided between two valves, so one moves over part of the range and the other over the rest. *Chapter 6*.

step test An experiment that changes the controller output by a fixed amount in manual and records how the measurement responds. *Chapter 1*.

surge pricing Raising a ride price by a multiplier when riders outnumber available drivers. *Chapter 9*.

Taylor rule A policy rule that sets the interest rate from the gap between inflation and its target and from the output gap. *Chapter 11*.

titration curve The pH of a solution plotted against the amount of acid or base added to it. *Chapter 6*.

two-point method Fitting a time constant and a dead time from the two times at which a step response reaches 28.3 and 63.2 percent. *Chapter 1*.

ultimate gain The proportional gain at which a loop oscillates steadily, neither growing nor dying away. *Chapter 1*.

velocity form A digital PID that computes only the change of its output at each execution, so the last output is its memory. *Chapter 1*.

wild goose chase A state of a ride-hailing zone with so few idle drivers that pickups are far away and drivers spend their time driving empty. *Chapter 9*.

Ziegler–Nichols rules Tuning rules of 1942 that set the controller from a step response or from the ultimate gain and period. *Chapter 1*.

Index of Control Topics

actuator saturation, 35, 55, 187, 286
  capacity limit, 135
  open loop at the limit, 68
  steady-state offset, 68, 101
actuator travel, 74, 103, 134
adaptive control, 150
  open problems, 165
  queue management, 191
adaptive cruise control
  open problems, 49
AI agents
  engineering practice, 333
  open problems, 338
  reproduction, 335
  truth audit, 335
  verification record, 334
  while studying, 18
anti-resonance, 303

back-calculation, 9, 35, 65, 96, 128, 156, 181, 205, 233, 282
bumpless transfer, 11, 16, 35, 65, 96, 129, 181, 206, 257, 281

cascade control, 97
  commissioning, 321
  reagent flow loop, 157
  servo drive, 298
characteristic equation
  with dead time, 179
chattering, 238
composition control, 55
condition monitoring, 322
conditional integration, 9
constraint control, 196
control layers, 14
controller action, 5, 96, 128, 180, 233, 282, 335
cruise control
  operating modes, 48
current loop, 305

damping ratio, 32
data export, 18
dead time, 12, 172
  apparent, 95, 127, 278
  misread, 328
  misread from a step test, 97, 128
  moving with throughput, 93
  state dependent, 171, 176
  transport delay, 88, 124
dead zone, 302
decoupling, 305
delay differential equation, 173
  numerical solution, 181
derivative kick, 8
direct shooting, 234
distributed control system
  history, 3
droop, 254

Euler integration
  equivalent time constant, 61

feedforward
  from a motion profile, 306
feedforward control, 45, 81, 164
  bias from the operating point, 181
  heat balance, 97
fieldbus
  PID function block, 332
final control element, 13
first-order process
  without dead time, 59
first-order-plus-dead-time model, 3, 87
float regulator
  history, 122
following error, 306

gain margin, 12, 307
gain scheduling, 12, 25, 96, 147, 188, 330
  history, 149
  on a titration-curve slope, 157

hysteresis, 201, 234

identification
  closed-loop, 338
  of inertia, 307
input-output linearization, 148
integral action
  history, 27

reset, history, 57
integral control, 196, 283
on a lagged plant, 285
integrating process, 7, 119, 252
integrating-plus-dead-time model, 120
reset time, 130
tuning in practice, 140
integrator
bounded, 257
integrator windup, 9, 35, 65, 96, 135, 181, 204, 257, 282, 286
at a rate limit, 157
in service, 80

Kalman filter, 17
input disturbance, 36, 66, 98, 131, 257
steady-state, 309

level control, 119
averaging, 122
limit cycle, 260, 280
backlash and friction, 312
from excess loop gain, 158
linearization, 31
about an operating point, 178
loop performance monitoring, 332

measurement
late and revised, 14, 277
measurement linearization, 147
inverse titration curve, 157
measurement noise, 43, 74, 134, 261
measurement validation
stuck-detector watchdog, 206, 218
mechanical resonance, 297
two-mass mode, 303
model fitting
least squares, 95
two-point method, 5
model predictive control
actuator bound, 65
against PID, 331
constraint tightening, 100
explicit, 307
history, 99
in the apps, 17
move blocking, 283
move suppression, 132
nonlinear, 234
on a reduced model, 206
on an identified model, 98, 131
preview of a disturbance, 35, 99, 207, 234
soft constraint, 258
motion control, 297
history, 299
motion profile, jerk-limited, 306
moving horizon estimation, 17, 37, 67, 114

natural frequency, 32
notch filter, 306
adaptive, 322

offset, 4, 32, 56, 285
proportional control, 72, 130, 254, 281
operating point
feasibility, 177
gain and time constant, 24, 62
process gain, 154, 179, 231
override control
queue limit, 205

phase margin, 12, 179, 307
loss with dead time, 171
PI control
on a pricing calendar, 233
queue management, 174
PID control, 1
anti-windup, 9, 16, 328
derivative action, 43, 134, 261
derivative filter, 8
design procedure, 331
gain change in automatic, 74
history, 3, 332
ideal form, 4
milestones, 2
open problems, 49, 336
parallel form, 4
position form, 5
series form, 4
through policy lags, 247, 274
tuning, history, 38
velocity form, 5, 16, 24, 64, 96, 128, 156, 180, 204, 224, 257, 282, 304, 335
PID tuning
data-driven, 332
internal model control, 6, 62, 155
internal model control, history, 63
SIMC rule, 6, 97, 129
Ziegler–Nichols rules, 6, 97, 129
process gain
mapping in manual, 164

quantization, 302
quantized actuator, 238, 256, 272
    limit cycle, 280

rate limit, 152, 238, 256, 260, 281
ratio control, 62, 164
relay autotuning, 8, 37, 67, 306, 330
    switching band, 78

sample time, 11, 260, 285, 330
sampled-data control, 11, 233, 247, 272
scenarios, 17
    completion rules, 17, 132, 182, 207, 235, 283
scorecard, 18, 153, 188
secant gain, 34, 155
self-regulation, 120
setpoint reachability, 55, 68
simulation engine, 14
simulator framework, 14
Smith predictor, 98, 190
species balance, 58
stability boundary, 180, 260, 279
state estimation
    disturbance bias, 282
steady-state inverse, 68
steady-state map
    fold, 198, 218, 224, 254
step test, 5, 63, 87, 278, 328
    in industry, 112
    on a nonlinear plant, 33, 95, 154
    on an integrating process, 127
    open problems, 113

ultimate gain, 8, 77, 129, 180

Wiener model, 150

Ziegler–Nichols tuning
    ultimate-gain rule, 306

Index of Applications

active queue management, 174, 333
    recommendations, 190
adaptive cruise control, 26
additive increase and multiplicative decrease, 173
aerodynamic drag, 25
agitated vessel, 58
ALINEA, 197
    PI-ALINEA, 199
analyzer noise, 60
automatic stabilizers, 251, 265
automobile
    speed control, 24

backlash, 302
    compensation, 306
blending, 56
blending tank, 55
blowdown, 121
Bohn's rule, 257
bottleneck router, 174
budget calendar, 256
buffer capacity, 152
bufferbloat, 174
buffering, 155

capacity drop, 197
caustic dosing, 150
cell transmission model, 197
charge balance, 152
coagulant dosing, 58
composition analyzer, 57
condensate flooding, 92
conductivity, 125
congestion collapse, 173
    history, 173
congestion window, 176
consumer price index, 277, 289
continuous pharmaceutical manufacturing, 58
    open problems, 81
control valve, 13
    equal-percentage trim, 152
    failure mode, 139
    installed characteristic, 92
    makeup service, 124
    split range, 163
    steam service, 111
    stiction, 101, 141, 154
cooling tower, 119
    induced-draft counterflow, 122
cost-push shock, 285
Coulomb friction, 302
coupling, flexible, 300
cruise control, 24
cycles of concentration, 121
    open problems, 141

debt brake, 250, 266
    history, 250
decision calendar, 233, 281
diluent stream, 58
discharge permit, 148
disinflation, 286
drift eliminator, 124
driver supply, 229, 241
drop-tail router, 173
    standing round trip, 187

eco-driving, 49
effective lower bound, 275
effluent treatment, 147
electric oven, 3
electronic throttle, 47
encoder, 302, 320
evaporation loss, 123
evaporative cooling, 121
expectations
    adaptive, 276
    anchored, 276
explicit congestion notification, 189

field weakening, 300, 305
film coefficient, 92
fiscal policy, 247
fiscal rules, 250
flash crowd, 183
float valve, 121
fluid-flow model, 173
fouling, 89
    open problems, 113
freeway traffic control, 196
fundamental diagram, 200

gas pedal, 27

gasoline blending, 57
gross domestic product
    revisions, 277

headwind, 28
heat exchanger, 87

I^2t protection, 302
implementation lag, 252
inertia ratio, 312
inflation targeting, 272
    history, 275
inflation-forecast targeting, 274
instrument, 13
internet congestion, 171
inverter, 320
    voltage circle, 302

leadscrew, 300
level transmitter, 122
    hydrostatic and echo, 138
log-mean temperature difference, 91
longitudinal vehicle dynamics, 27
lookup table, 233
loop detector, 202, 216
Lucas critique, 266

Maastricht deficit limit, 257
machine tool
    feed axis, 297
makeup water, 119
marking probability, 174
monetary policy, 272
monetary transmission, 275
multiplier-accelerator model, 249

neutralization tank, 150
nowcasting, 333

occupancy, 202
oil shock, 286
Okun's law, 277
output gap, 275
    misperceived, 275
    real-time estimate, 267

perfect mixing, 61
permanent-magnet synchronous motor, 300
pH electrode, 14, 150
    fouling, 152
    glass, 162
pH neutralization, 147
Phillips curve, 274
Phillips machine (MONIAC), 249
pickup time, 228, 240
policy rate, 275, 290
price cap, 239
price elasticity of demand, 227, 242
process analyzer
    sample system, 78
product giveaway, 57
propagation delay, 176
public debt, 247
    interest rate and growth, 252

queueing delay
    measurement, 189

ramp metering, 196, 199
    coordinated, 199
    equity, 218
    history, 198
    release rate, 197, 217
ramp queue, 202
    spillback, 202
random early detection, 173
ratio control
    blending, 79
reaction invariants, 150
reagent valve, 152
real interest rate, 275
real-time data, 275, 291
recognition lag, 252, 265
residence time, 56
    distribution, 58
resistance temperature detector, 109
ride-hailing, 224
risk premium, 254
road grade, 28
rolling resistance, 30
round-trip time, 172
router queue, 174

sacrifice ratio, 286
servo drive, 297
shell-and-tube exchanger, 88
speedometer, 27
stagflation, 286
startup
    empty tank, 69
steam header, 90
steam heating, 87
steam trap, 90

stirred tank
    composition control, 55
surge multiplier
    static gains, 231
surge pricing, 224, 241
    alternatives, 242
    history, 227

taxi
    regulated fare, 226
Taylor curve, 285
Taylor principle, 279
Taylor rule, 274, 290
TCP congestion control, 171
TCP Reno, 176
throughput fold, 230
titration curve, 149
total time spent, 202
two-mass system, 302
two-sided market, 227

variable speed limits, 199
vehicle platoon, 27

water consumption, 141
wheel speed sensor, 46
wild goose chase, 226

About the Editor

John D. Hedengren is a professor in the Department of Chemical and Biological Engineering at Brigham Young University, where he leads the Process Research and Intelligent Systems Modeling (PRISM) group. He received the B.S. and M.S. degrees in chemical engineering from Brigham Young University and the Ph.D. degree from the University of Texas at Austin in 2005. He developed advanced process control applications for polymer production at ExxonMobil Chemical Company and commissioned nonlinear model predictive controllers with PAS, Inc. before joining the faculty at Brigham Young University in 2011.

His teaching covers process dynamics and control, engineering computing, data science, machine learning, and dynamic optimization, with course material published online for use inside and outside the classroom. He developed the Temperature Control Lab, a portable heat transfer device now used by many universities for process control and data science instruction, and he maintains the APMonitor and GEKKO optimization software used in those courses. He serves as Chair of Control Education for the IEEE Control Systems Society and as a trustee of the CACHE Corporation. He received the AIChE David Himmelblau Award in 2014, the AIChE Computing Practice Award in 2018, and the John R. Ragazzini Award in 2024, and he was inducted into the Process Automation Hall of Fame in 2025.

Volumes in This Series

Interactive Control Engineering: Principles and Applications has 20 volumes. Nine are organized by control technology and eleven by industry or application area. Every chapter in every volume has a simulator at https://apmonitor.com/apps/.

1 **Process Dynamics, Modeling, and Identification.** How plants respond, and how to find the model.
2 **Practical PID Control.** Tuning, anti-windup and implementation that works on real loops.
3 **Feedforward, Cascade, and Ratio Control.** Choosing a structure from where the disturbance enters.
4 **Override, Split-Range, and Constraint Control.** When limits decide what the controller may do.
5 **Delays, Slow Measurements, and State Estimation.** Controlling what arrives late or cannot be measured.
6 **Multivariable and Plantwide Control.** Pairing loops, managing inventories, closing the recycle.
7 **Nonlinear Dynamics, Oscillations, and Stabilization.** Plants that will not stay put without control.
8 **Batch, Run-to-Run, and Hybrid Control.** Recipes, endpoints and learning from the last run.
9 **Model Predictive Control and Economic Optimization.** Planning moves ahead, and choosing where to operate.
10 **Control in Upstream and Midstream Oil and Gas.** From the wellbore to the terminal.
11 **Control in Petroleum Refining.** Throughput, quality and limits from crude unit to blending.
12 **Control in Petrochemicals and Polymers.** Severity, selectivity and grade from cracker to pellet.
13 **Control in Chemical Plants.** Recycle, equilibrium and energy in continuous chemical processes.
14 **Control in Semiconductor Manufacturing.** Run-to-run control where little can be measured in the chamber.
15 **Control in Mining and Mineral Processing.** Variable ore, late assays and equipment at its limit.
16 **Control in Iron, Steel, and Metals Production.** Slow furnaces, fast strip, and measurements that arrive once.
17 **Control of Energy Conversion and Storage.** Fast regulation and slow dispatch on a changing grid.
18 **Control of Nuclear and Fusion Systems.** Reactor physics, protection limits and coordinated control.
19 **Control of Biological Systems and Human Performance.** Variable subjects, sparse samples and limits that cannot be crossed.
20 **Motion Control, Robotics, and Autonomous Systems.** Tracking, navigation, contact and allocation.