INTERACTIVE CONTROL ENGINEERING
Principles and Applications

Volume 2 Practical PID Control

CHAPTER 13

Summary and Outlook

Pages 328–341

Edited by John D. Hedengren

How to cite this chapter. "Summary and Outlook," in *Practical PID Control*, John D. Hedengren, Ed., vol. 2 of *Interactive Control Engineering: Principles and Applications*. Advanced Process Solutions, LLC, 2026, ch. 13, pp. 328–341.

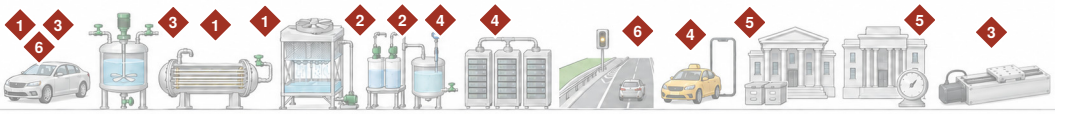
The page numbers in this file are those of the printed volume.

The volume online: contents, other chapters and citation records
https://apmonitor.com/apps/books/v02-practical-pid/

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

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).

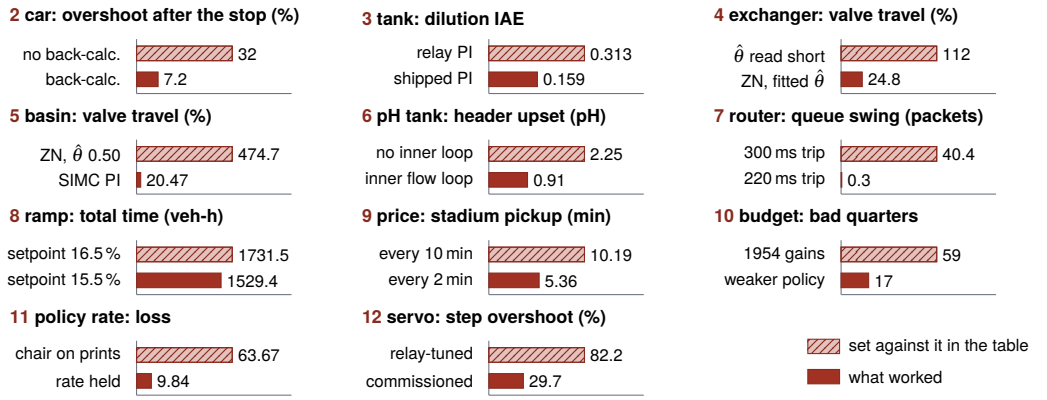


Figure 13.1 One pair of results from each chapter, with the numbers as the "Measured result" column of Tables 13.1 and 13.2 prints them. Each panel has its own scale from zero, and a smaller value is better in every panel. The pH tank's pair is its header upset, because its fixed gain never returned 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

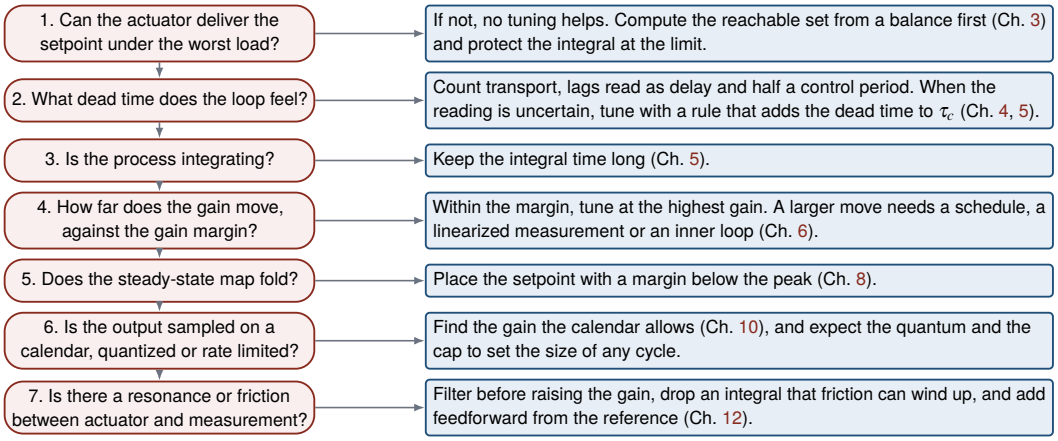


Figure 13.2 The seven questions of this section in order, with the list's answer and the chapters where it decided the result. Notice that the actuator and the dead time come first, and that the questions of calendars and resonances, which only some plants pose, come last.

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.

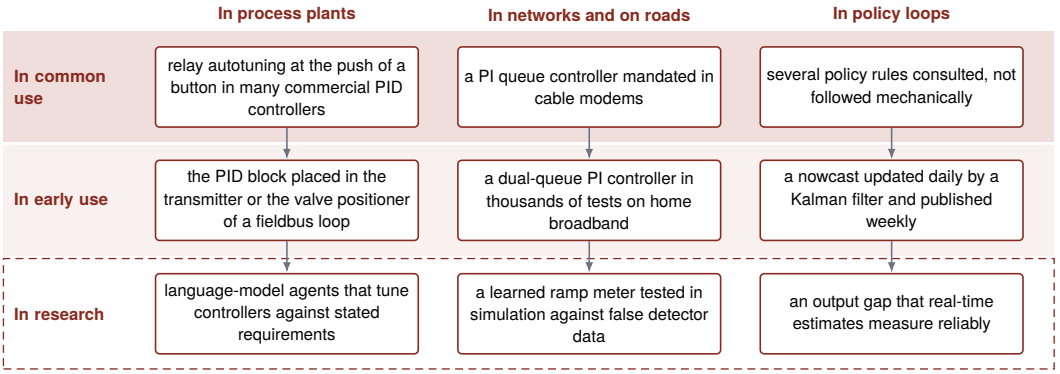


Figure 13.3 Developments in three threads. The top lane holds what the cited source calls standard, mandated or regular practice, the middle lane what it reports in service or in field tests, and the bottom lane what it reports from simulation, from a set of test tasks or as a need, in the editor's reading. 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.

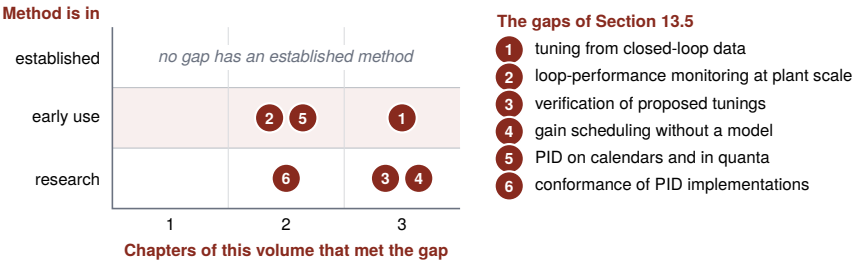


Figure 13.4 The six gaps of this section. Across is the number of chapters that each item names as evidence: three for gap 1 (Chapters 2, 4 and 5), two for gap 2 (5 and 6), three for gap 3 (2, 3 and 12), three for gap 4 (6, 7 and 9), two for gap 5 (10 and 11) and two for gap 6 (2 and 8). Up is the maturity 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.