INTERACTIVE CONTROL ENGINEERING
Principles and Applications

Volume 8 Batch, Run-to-Run, and Hybrid Control

CHAPTER 1

Introduction: Batches, Repeated Runs, and Switched Plants

Pages 1–22

Edited by John D. Hedengren

How to cite this chapter. "Introduction: Batches, Repeated Runs, and Switched Plants," in *Batch, Run-to-Run, and Hybrid Control*, John D. Hedengren, Ed., vol. 8 of *Interactive Control Engineering: Principles and Applications*. Advanced Process Solutions, LLC, 2026, ch. 1, pp. 1–22.

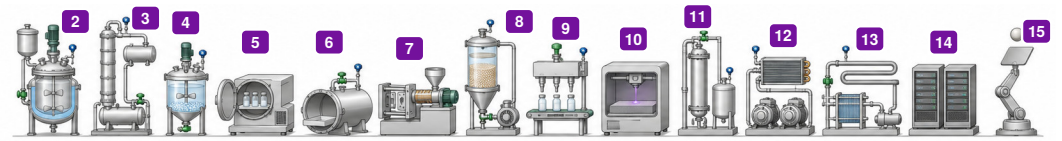
The page numbers in this file are those of the printed volume.

The volume online: contents, other chapters and citation records
https://apmonitor.com/apps/books/v08-batch-hybrid/

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

CHAPTER 1

Introduction: Batches, Repeated Runs, and Switched Plants



The chapters 2 semi-batch reactor 3 batch column 4 crystallizer 5 freeze-dryer 6 cure autoclave 7 injection molding 8 particle coater 9 vial filler 10 powder bed fusion 11 antibody capture 12 refrigeration pack 13 milk pasteurizer 14 web service 15 juggling paddle

Notice that the first five units each run one batch to an end, the next five repeat a shot, a coating, a vial, a layer or a column load, and in the last four the actuators or the dynamics switch. Illustration, not to scale.

IN PLAIN TERMS **Plants That Start, Repeat or Switch**

The main idea. Many plants never settle at one steady point. A batch is started, led through a recipe and stopped. A machine repeats the same short run hundreds of times, and a thermostat or a pool of servers switches in whole steps.

The challenge. The product of a batch is judged once, after the end, so a mistake found then cannot be undone in that batch. Each run can learn from the last. But a correction made after every result also copies the noise of one run into the next.

Why it matters. These plants make medicines, food, molded parts and fuel, and they run online services that people use every day.

THREE WAYS TO READ THIS CHAPTER

Explore	6 min	This page, Section 1.2 with its timeline and two History boxes, the Did you know? boxes in Sections 1.4.5, 1.4.8 and 1.4.10, 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 kettle example worked, then Exercises 1 to 10.
Practice	5 min	This page, Section 1.4.10 with Table 1.3, Section 1.5 with Figures 1.7 and 1.9, and Section 1.8 with Figure 1.11.

1.1 The Question This Volume Answers

The loops of the earlier volumes hold a plant near one operating point for as long as it runs. The plants of this volume have no such point. A batch starts cold, follows a program and ends, and its product is judged once, after the end. A molding machine, a vial filler or a printer repeats the same run hundreds of times, and each run can learn from the last. A thermostat, a contactor or a bouncing ball switches between modes, and the switching is the control. This volume answers one question: how is a plant controlled when its operation is a sequence, a repetition or a switch, rather than a steady state?

Fourteen chapters answer it on fourteen plants, each with an app that the reader runs in a browser. Chapters 2 to 6 run one batch along a trajectory to an endpoint under a limit that holds at every instant. Chapters 7 to 11 repeat a shot, a coating run, a vial, a layer or a column cycle, end each on an event or an estimate, and correct the next one from the last. Chapters 12 to 15 are plants whose actuators or dynamics switch: thermostats on one suction header, a diversion valve under a supervisor, a fleet of servers counted in integers and a paddle that strikes a ball.

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

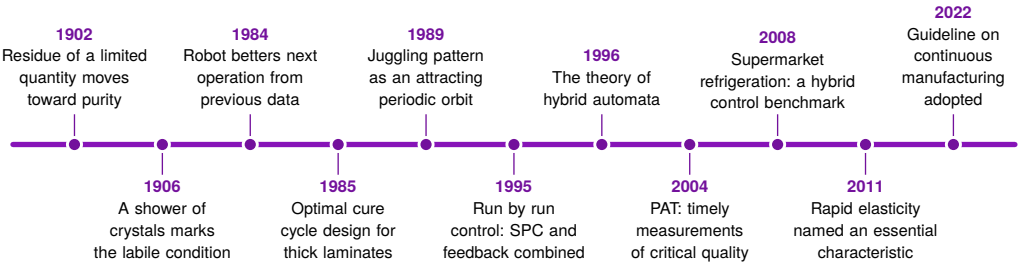


Figure 1.1 Milestones of batch, run-to-run and hybrid control in the systems of this volume. The dates are from [4] (1902), [5] (1906), [7] (1984), [8] (1985), [9] (1989), [10] (1995), [11] (1996), [12] (2004), [13] (2008), [14] (2011) and [15] (2022). Notice that the batch was described long before it was controlled, and that the last cards move the end of a phase from the clock to a measurement. Events are evenly spaced, not to a time scale.

- Write a recipe as phases with transition conditions, and choose whether a clock, a measured event or an estimate ends each phase.
- Track a trajectory with feedback and model feedforward, and say what each leaves behind.
- Find the path and endpoint constraints of a batch, the shape of the fastest feasible batch and the back-off each constraint needs.
- Design a run-to-run correction and predict its response to a step, a drift, noise, a gain error and a late measurement.
- Design an iterative learning update and test it for monotone convergence.
- Model a switched plant as a hybrid automaton, set a hysteresis band or a dwell time, handle an integer actuator, and find the cyclic steady state a repeated operation settles into.

This chapter develops each of these on one small example and names the chapter that uses it. Overviews of batch control and optimization [1, 2] and of learning in batch processes [3] give fuller treatments. PID forms and tuning are taken as known from Volume 2.

1.2 A Short History

The milestones of Figure 1.1 follow the methods of this volume through very different plants. The first two describe a batch long before anyone controlled one. In 1902 experiments on boiling mixtures showed that, as the distillation of a limited quantity proceeds, the composition of the residue moves always in one direction toward purity [4], the moving composition that Chapter 3 follows. In 1906 a cooling solution watched by its refractive index grew a few crystals and then, at a lower temperature, threw a shower of them throughout the liquid, and cooling slow enough never reached that labile condition [5], the trajectory question of Chapter 4.

> ⧖ **HISTORY** **1930s to 1980s: Batch Control Waits Its Turn**
>
> Batch processing was dominant until the 1930s, yet batch process control drew significant interest only 50 years later. When control theory emerged in the 1940s, industry was shifting from batches to continuous operation, and the dynamics and control of batch processes became a subject of study only in the 1980s. Engineers then tried to carry over what continuous plants had taught them, and the lack of a steady state made that hard. Programmable logic controllers entered the process industries in the early 1970s, following the automotive industry, and replaced relays, counters and timers that had offered much less flexibility [6].

Optimization of the batch followed: a report of 1985 treated the cure cycle of thick composite laminates as a design variable [8], the recipe that Chapter 6 reshapes.

Switching became a subject of its own. A robot that juggled by batting, presented in May 1989, was designed so that its juggling pattern was an attracting periodic orbit [9], the orbit of Chapter 15. A 1996 summary of results on hybrid automata [11] gave Section 1.4.7 its language, and by 2008 a supermarket refrigeration system had become a benchmark problem of hybrid control [13], the plant of Chapter 12. A 2011 definition of cloud computing named rapid elasticity among its essential characteristics [14].

The last milestones move the end of a phase from the clock to a measurement. A 2004 guidance on process analytical technology asked that manufacturing be designed, analyzed and controlled through timely measurements of critical quality attributes [12]. A guideline on continuous manufacturing, adopted in November 2022, noted that such technology lets disturbances be detected in real time and illustrated a therapeutic-protein process with continuous capture chromatography [15], the train of Chapter 11.

> ⧖ HISTORY **1984 to 1995: Learning From the Last Run**
>
> A paper of 1984 proposed a betterment process that improves the next operation of a robot with the data of the previous one. Each input was the last input plus an increment built from the difference between the motion achieved and the motion desired. Its convergence was assured under reasonable conditions and shown in computer simulation [7]. In chip making, a 1993 analysis gave a simple condition relating the EWMA weight and the estimated process gain that keeps such a control stable [16]. A run by run controller of 1995 combined statistical process control with feedback. A rapid mode met sudden shifts, such as those after maintenance, and a gradual mode met drifts, such as deposits building up inside a reactor [10].

1.3 Notation and Conventions

A batch has two time axes. Time within the batch is t, in minutes here. The batch number is j, which the chapters call a run, a shot, a vial, a layer, a cycle or a bounce. The framework's PID counts its executions with k inside a batch (Section 1.6.2). A recipe setting chosen before batch j is u_j, the result measured after it is y_j, its target is y^* and an unmeasured disturbance is d_j. The weight of an exponentially weighted moving average (EWMA) is λ, the gain of an iterative learning update is γ, its lead is ℓ samples and its filter is Q. A hysteresis band runs from T_{lo} to T_{hi}, an integer actuator has the count n, and a cycle-to-cycle map has the eigenvalue μ.

The chapters keep their own symbols. Chapter 9 writes the EWMA weight of each filling head as λ_h, and Chapter 11 uses λ for its capacity filter. Chapter 10 writes the learning gain γ as here but uses λ for the lead, a length in millimeters. Chapter 7 writes the loop gain of its shot-to-shot correction as k, which is $\lambda\xi$ in Section 1.4.5. Chapter 14 uses δ for a deadband in servers. The framework's h is its integration step and a its controller action, so neither appears here with another meaning.

1.4 Theory on One Batch Kettle

1.4.1 The Running Example

The running example is a kettle that heats a charge of liquid food with three identical electric elements, as shown in Figure 1.2. Its temperature obeys Equation (1.1), with the time constant $\tau = 20\,\text{min}$, the room at $T_a = 20\,^\circ\text{C}$ and each element worth $K = 40\,^\circ\text{C}$ of steady rise.

$$\tau \frac{\mathrm{d}T}{\mathrm{d}t} = T_a - T + K u, \qquad 0 \le u \le 3 \tag{1.1}$$

The input u counts elements. Under a thyristor it varies continuously, and under contactors it is an integer, so one, two or three elements hold the kettle at 60, 100 or 140 °C. The product is judged on a thermal dose accumulated over its whole history, Equation (1.2).

$$F = \int_0^{t_f} 10^{(T-80)/10}\,\mathrm{d}t \tag{1.2}$$

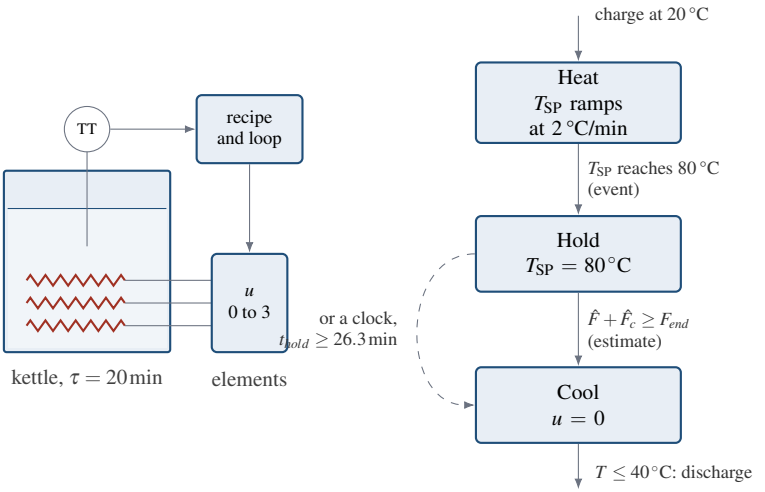


Figure 1.2 The Running Example. Left, a Kettle Heated by Three Elements Under a Thyristor (u Continuous) or Contactors (u an Integer). Right, Its Recipe as Three Modes With the Guard That Ends Each: an Event, an Estimate of the Dose, or a Clock (Dashed).

A minute at 80 °C counts as one minute, and each 10 °C hotter counts ten times as much, the form used for the lethality of a thermal treatment. The specification is $F \geq 30$ min, an endpoint constraint, and the product scorches above 90 °C, a path constraint.

The kettle shows at once what separates a batch from a continuous plant. It has no steady state to hold, since it is charged cold, heated, held, cooled and emptied. Its result is a functional of the whole path, so the same final temperature can come with any dose. And its sensitivity moves along the batch: near 80 °C a 1 °C error changes the dose rate by 25.9 %, while at 50 °C the dose rate is a thousandth of its value at 80 °C and the same error costs almost nothing. Every chapter of the volume has these three properties. The batch column of Chapter 3 grows harder to separate with no disturbance applied, and its process gain decays toward zero over the run. The crystallizer of Chapter 4 is graded only after the batch, on crystals that carry its whole cooling history.

1.4.2 Recipes and Phases

A recipe divides the batch into phases. Each phase has a setpoint program, an active controller, and an end condition, the guard on the way out of the phase in Figure 1.2. A guard can be a clock, a measured event or an estimate, and the choice matters as much as any tuning. The kettle's heat phase ends on an event, the setpoint reaching 80 °C. Its hold ends on an estimate: the dose $\hat{F}$ integrated from the thermocouple, plus the dose the model predicts for free cooling from the present temperature, $\hat{F}_c(T)$, must reach the endpoint setting F_{end}. From 80 °C the cooling still adds 1.57 min of dose, and from 88 °C it adds 8.66 min.

Table 1.1 compares that guard with a clock set to the nominal hold of 26.3 min, on four batches with $F_{end} = 30$ min. With a well-tracked heat-up both deliver 30.09 min. A batch heated by feedback alone lags the ramp, as Section 1.4.3 shows, and the clock then delivers 21.0 min while the estimate extends the hold to 35.4 min and delivers 30.1 min. Weaker elements cost the clock 1.8 min of dose and the estimate nothing. A thermocouple that reads 1 °C high fools both alike, to 23.9 min, because the estimate integrates the same wrong reading. An estimate removes what the model can see and keeps what the instrument cannot.

Every phase change also hands the actuator from one controller or program to the next. The framework's PID tracks the applied output while it is not in charge, so the controller that takes over starts from the value its predecessor left (Section 1.6.2), and Volume 2 treats bumpless transfer in

Table 1.1 Ending the Hold on a Clock or on the Dose Estimate, With $F_{end} = 30\,\text{min}$

Batch	Dose, clock (min)	Dose, estimate (min)	Hold, estimate (min)
PI with feedforward, nominal	30.09	30.09	26.3
PI alone, lagging the ramp	21.03	30.11	35.4
Elements 10 % weak	28.30	30.10	28.1
Thermocouple 1 °C high	23.93	23.93	26.3

full [17]. The chapters use all three kinds of guard. Chapter 8 ends each coating layer on a timer, on an integrated thickness estimate or on that estimate corrected by a load cell, and abandons a phase with a latched fault when it cannot finish. Chapter 7 switches from fill to pack on screw position, cavity pressure or time, with a time limit behind all three, and hands the valve from the speed loop to the pressure loop by tracking. Chapter 5 ends primary drying on a recipe time, on its probes, on a gauge ratio held for 15 min or on its estimator. Chapter 11 ends each loading on a time, an integrated load, a measured breakthrough or a learned capacity. Chapter 13 logs each parcel's history apart from the instruments, the dose idea applied to every parcel of a continuous stream.

1.4.3 Trajectory Tracking

The heat phase asks the kettle to follow a ramp of $\rho = 2\,^\circ\text{C/min}$ from 20 to 80 °C. A PI loop with $\tau_I = \tau = 20\,\text{min}$ and $K_c = 0.1\,\text{element}/^\circ\text{C}$ cancels the plant's lag and leaves a first-order closed loop with the time constant $\tau_c = \tau/(K_c K) = 5\,\text{min}$. On a ramp such a loop lags by Equation (1.3).

$$e_{ss} = \rho\,\tau_c = 2\,^\circ\text{C/min} \times 5\,\text{min} = 10\,^\circ\text{C} \tag{1.3}$$

The lag is no tuning fault. The output must keep rising, and an integral term rises at a constant rate only on a constant error. The simulated kettle reads 70.01 °C when the setpoint reaches 80, collects 0.22 min of dose on the ramp against 2.17 min for a ramp tracked exactly, and reaches 79 °C at 41.5 min, which is $30 + \tau_c \ln 10$. The supersaturation controller of Chapter 4 meets the same fact from the other side: its integral term generates the cooling ramp, and a proportional term alone could not.

The cure is to compute the input the trajectory needs and give it to the loop as its bias, Equation (1.4), so that feedback corrects only what the model misses.

$$u_{ff}(t) = \frac{T_{\text{SP}}(t) - T_a + \tau\,\dot{T}_{\text{SP}}(t)}{\hat{K}} \tag{1.4}$$

On the ramp u_{ff} runs from 1 to 2.5 elements, and in the hold it is 1.5. With the model exact the ramp error has an RMS of 0.012 °C and the batch takes 78.3 min instead of 87.4 min. With elements 10 % weaker or stronger than the model, the RMS is 0.96 or 0.79 °C, and the stronger set overshoots the hold by 0.90 °C. That residue repeats in every batch, which is the opening for Section 1.4.6. Chapter 3 enters its reflux trajectory as the controller bias in the same way, with a proportional trim around it, and finds that a composition loop holding its setpoint closely collects less product than the open-loop trajectory. In Chapter 4 three cooling curves between the same endpoints give mean sizes that differ by a factor of 4.3, and the best-tracked batch gives the second-worst product. Tracking well matters less than tracking the right trajectory.

1.4.4 Path and Endpoint Constraints

A path constraint holds at every instant, here $T(t) \le 90\,^\circ\text{C}$. An endpoint constraint holds once, at the end, here $F(t_f) \ge 30\,\text{min}$. The fastest batch that meets both is made of arcs on which an input or a constraint is at a bound [18]. For the kettle it heats at full power until the temperature meets the path constraint, holds it there with the holding input, and cools with the elements off. The dose then accumulates fastest at every instant.

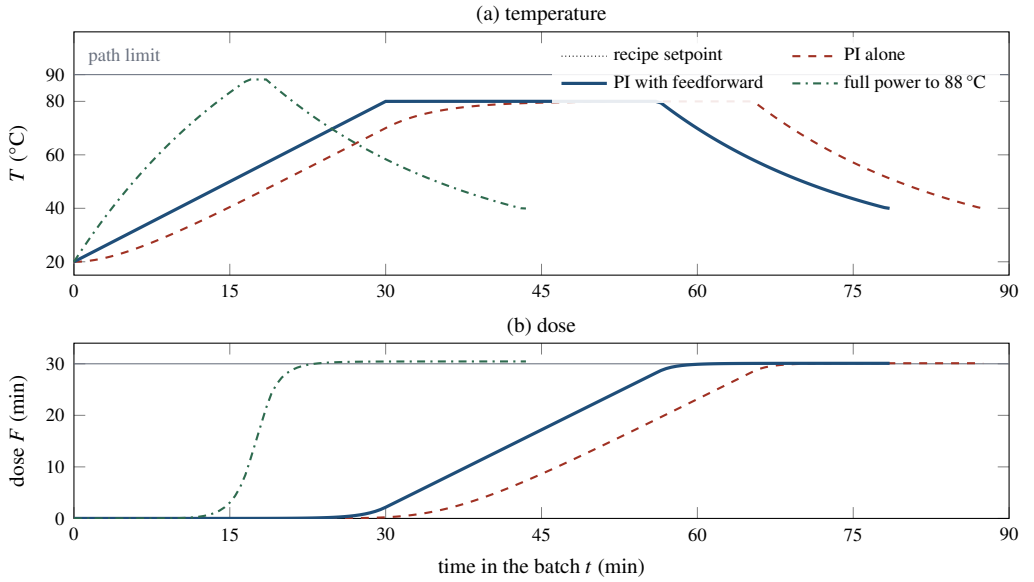


Figure 1.3 One Batch Under Three Policies, Each Ending Its Hold on Equation (1.5). (a) Temperature: PI Alone on the Recipe (Dashed), PI With the Feedforward of Equation (1.4) (Thick Solid), and Full Power to a 2 °C Back-Off From the Path Limit (Dash-Dotted), With the Recipe Setpoint Dotted. (b) The Dose, With the Endpoint Constraint at 30 min.

The constraint cannot be ridden at its value, so the policy switches at 88 °C, a back-off of 2 °C. Full power reaches it in $\tau \ln(120/52) = 16.7$ min, the feedforward of Equation (1.4) then holds it with 1.7 elements and the PI trims, and the hold ends on the guard of Section 1.4.2, written as Equation (1.5).

$$\hat{F}(t) + \hat{F}_c\big(T(t)\big) \geq F_{end} \tag{1.5}$$

Figure 1.3 compares this policy with the recipe. The dose is 11.6 min when the hold starts and 21.5 min after a hold of only 1.7 min, and the cooling supplies the last 9.0 min, so the batch takes 43.1 min against 78.3 min for the recipe, 45 % shorter. Ending the hold on the dose so far, without the predicted remainder, would overshoot the endpoint by the whole cooling dose.

The back-off pays for what the model does not know. With elements 10 % weaker, nominal or stronger, the fast policy peaks at 88.07, 88.19 and 88.57 °C, so 2 °C covers that uncertainty with 1.4 °C to spare. A noisier measurement or a slower loop would need more. The same structure, a fast arc to a limit and a hold on it with a margin, recurs through the batch chapters, each with its own limit. In Chapter 2 the limit is on the temperature the batch would reach if cooling failed, held below the decomposition threshold by the dosing controller rather than by the temperature loop. Chapter 4 states its wall limit as a lower output limit of the jacket loop that moves with the measurement. In Chapter 5 the hottest vials limit the shelf while the slowest decide the endpoint, and in Chapter 6 the slowest thermocouple sets the ramp while the hottest one and a core no thermocouple reaches set the limit, through a selector (Volume 4). Several of these chapters also offer a shrinking-horizon predictive controller that plans to the endpoint (Section 1.6.3, and Volume 9).

1.4.5 Run-to-Run Control

Some errors no estimate inside the batch can see. The thermocouple bias of Table 1.1 is one, and only a measurement of the product, here a laboratory test of the dose after each batch, reveals it. Run-to-run control uses that result to adjust the next batch's recipe. Near its operating point the result

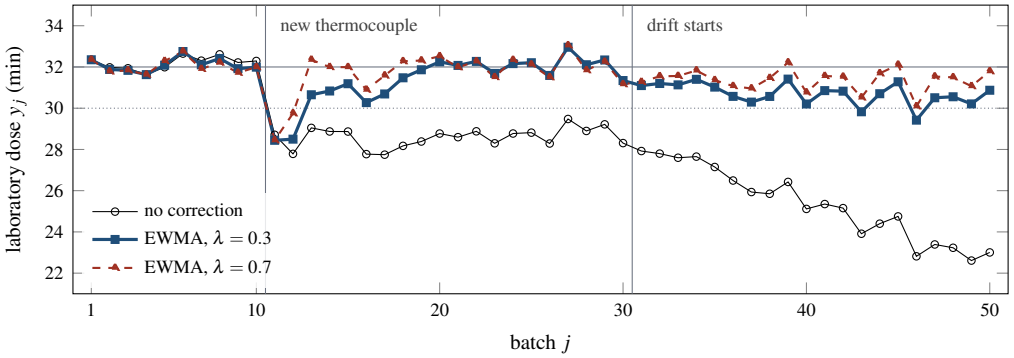


Figure 1.4 Run-to-Run Control of the Kettle's Dose Over 50 Batches With the Same Laboratory Noise: No Correction (Thin Solid, Circles), EWMA With $\lambda = 0.3$ (Thick Solid, Squares) and $\lambda = 0.7$ (Dashed, Triangles). Thin Horizontal Lines Mark the Target of 32 min and the Specification of 30 min (Dotted). A Thermocouple Reading 0.5 °C High Arrives at Batch 11 and Drifts From Batch 31.

of batch j is modeled by Equation (1.6), with the true gain β, a slowly moving disturbance d_j and a noise ε_j new in every batch.

$$y_j = \beta\, u_j + d_j + \varepsilon_j \tag{1.6}$$

The EWMA controller [10, 19] estimates the intercept from each result less the model's prediction from the setting that produced it, and sets the next batch to meet the target with the estimate, Equation (1.7), where b is the model's gain.

$$\hat{d}_j = \lambda\,(y_j - b\,u_j) + (1-\lambda)\,\hat{d}_{j-1}, \qquad u_{j+1} = \frac{y^* - \hat{d}_j}{b} \tag{1.7}$$

With $\xi = \beta/b$, substituting Equation (1.7) into Equation (1.6) gives the four properties collected in Equation (1.8). A step in d shrinks by the factor $1-\lambda\xi$ each batch, so the loop is stable for $0 < \lambda\xi < 2$ [16]. A drift of δ per batch leaves a steady offset δ/λ, and white noise of variance σ^2 leaves an output variance raised by the factor $2/(2-\lambda)$.

$$\mu = 1 - \lambda\xi, \qquad e_{drift} = \frac{\delta}{\lambda}, \qquad \sigma_y^2 = \frac{2\,\sigma^2}{2-\lambda} \tag{1.8}$$

A simulation of the linear model checks each: a drift of 0.1 per batch leaves 0.333 at $\lambda = 0.3$ and 0.143 at 0.7, and 200,000 batches of unit noise give standard deviations of 1.0844 and 1.2406 against 1.0847 and 1.2403. The weight trades the two: $\lambda = 0.3$ brings a step within 10 % in seven batches and widens the noise by 8.5 %, $\lambda = 0.7$ takes two batches and widens it by 24 %. Correcting after every result can only widen pure noise, the lesson of the funnel experiment, and pays only against disturbances that persist [20].

On the kettle the setting is F_{end} and the result is the laboratory dose, with a target of 32 min, two minutes above the specification, and a laboratory noise of 0.5 min. The model gain $b = 1$ is right, since the simulated gain is 1.0013. Figure 1.4 runs 50 batches. From batch 11 a new thermocouple reads 0.5 °C high, which cuts the dose by 3.48 min and the gain to 0.89, and from batch 31 it drifts up by 0.05 °C a batch, about 0.33 min of dose per batch. Without correction 40 of the 50 batches fail the specification. With $\lambda = 0.3$ four fail and with $\lambda = 0.7$ two. Over batches 41 to 50 the two weights fall short of the target by 1.5 and 0.7 min, against δ/λ of 1.2 and 0.5 min with the drift of batch 45, a drift that is itself speeding up.

A result that arrives late changes the picture. Written as an integral action on the error, $u_{j+1} = u_j + (\lambda/b)(y^* - y_{j-D})$ with a delay of D batches, the loop has the characteristic polynomial of Equation (1.9).

$$z^{D+1} - z^D + \lambda\xi = 0 \tag{1.9}$$

It is stable for $\lambda\xi$ below 2, 1, 0.618 and 0.445 at delays of 0, 1, 2 and 3 batches, and with $D = 1$ it is the equation that governs a laboratory-only loop in Volume 5. The intercept form of Equation (1.7) instead compares each late result with the setting that produced it, so with $b = \beta$ it is unaffected by any delay, and a simulation confirms it at $\lambda = 0.7$ with delays of one and two batches. A sufficient condition for stability under any metrology delay with a gain error is known [21].

Three chapters use this section. Chapter 7 corrects pack pressure from each weighed part by the integral form, with $\lambda\xi = 0.55$, and derives the stability bound and the variance factor above. Chapter 9 meets the delay: about six results per head are in flight, the integral law counts each error several times and oscillates at long latencies, and the record-based law, which is Equation (1.7), is unaffected. Chapter 11 learns the resin capacity from earlier switches with an EWMA of weight 0.5. Volume 14 applies run-to-run control to wafers, where it was developed.

> **+ DID YOU KNOW? The Funnel Experiment, Answered in 1990**
>
> Deming's funnel experiment has been used to show that actively controlling a process can increase the variance of its output. A note of 1990 answered that such situations seldom occur in the continuous process industries, where active control to a target had served with excellent results for decades [20].

1.4.6 Iterative Learning Control

Run-to-run control corrects one number per batch. When the error is a whole trajectory that repeats, the correction can be a whole trajectory too. Iterative learning control (ILC) stores an input profile $v_j(t_i)$ on a grid of samples and updates it after each batch from the error of that batch, shifted ahead by a lead of ℓ samples to cover the plant's lag, as shown in Equation (1.10) [7, 22].

$$v_{j+1}(t_i) = Q\left[v_j(t_i) + \frac{\gamma}{\hat{K}}\, e_j(t_{i+\ell})\right] \tag{1.10}$$

For a plant whose sampled transfer function is G, an error component of frequency ω is multiplied by the factor of Equation (1.11) each batch, and the error shrinks monotonically if the factor is below one at every frequency [23].

$$\mu(\omega) = \left|Q(\omega)\left[1 - \frac{\gamma}{\hat{K}}\, G(\mathrm{e}^{\mathrm{j}\omega})\, \mathrm{e}^{\mathrm{j}\omega\ell}\right]\right| < 1 \tag{1.11}$$

On the kettle the heat phase and the first ten minutes of the hold run open loop on a one-minute grid, starting from the feedforward of Equation (1.4) with elements 10 % weaker than the model, so the first batch ends the ramp 6.6 °C low with an RMS error of 4.78 °C. The thermocouple has 0.1 °C of noise. With $\ell = 1$ and no filter the factor is $1 - 0.9\gamma$ at zero frequency and $1 - 0.9\gamma(1-p)/(1+p)$ at the shortest period, with $p = \mathrm{e}^{-1/20}$, so with $\gamma = 1$ a slow error falls tenfold each batch and an error that alternates every minute by only 2 %. The RMS falls to 2.59, 1.43 and 0.84 °C in the next three batches and reaches the noise, about 0.1 °C, by batch 40, as Figure 1.5 shows. Gains up to 2 converge monotonically on this plant, and $\gamma = 0.5$ reaches only 0.55 °C by batch 10.

Learning fails where the lead is wrong. Put the thermocouple in a thermowell that lags 2 min and the one-sample lead falls short: the factor rises to 1.036 at a period of about 20 min. The RMS falls to 1.30 °C at batch 13, then grows back to 2.9 °C by batch 60, because each batch adds a little of the error it cannot remove. A zero-phase low-pass Q, run forward and backward with a pole of 0.3, pulls the factor down to 1.003 and the error settles between 0.71 and 0.80 °C, the price being the short-period detail it no longer learns. A lead of two samples fits the lag better, holds the factor at 1.0005 and keeps falling, to 0.40 °C at batch 100. Learning also copies the noise of each batch into

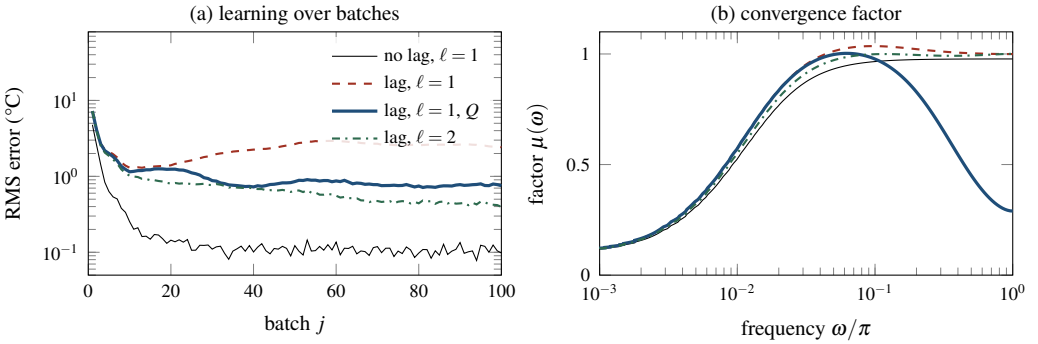


Figure 1.5 Iterative Learning on the Kettle's Heat Phase With $\gamma = 1$ and Elements 10 % Weaker Than the Model. (a) RMS Error per Batch. (b) The Factor of Equation (1.11). Thin Solid: No Thermowell Lag, Lead of One Sample. With a 2 min Thermowell Lag: Lead of One Sample (Dashed), the Same With a Q-Filter (Thick Solid), and a Lead of Two Samples (Dash-Dotted).

the next and has nothing for a batch that does not repeat. Chapter 10 learns a laser power profile along each layer's outline with this update, $\gamma = 0.5$, a lead of 0.6 mm and a Q-filter, adds the feedback effort into the profile, and measures each failure named here: growth without the filter, a corner blurred by too wide a filter, and a layer that breaks the repetition [24].

1.4.7 Hybrid Automata, Switching and Hysteresis

A hybrid automaton [11, 25] is a set of modes, each with its own differential equation, joined by guards that say when the mode changes and resets that say what jumps when it does. The recipe of Figure 1.2 is one. So is a thermostat. Let the kettle hold 80 °C on contactors, with one element always on and the second switched on below $T_{lo} = 79\,^{\circ}\mathrm{C}$ and off above $T_{hi} = 81\,^{\circ}\mathrm{C}$. The kettle then heats toward 100 °C and cools toward 60 °C, and the on and off times follow from Equation (1.1) as Equation (1.12).

$$t_{on} = \tau \ln \frac{100 - T_{lo}}{100 - T_{hi}}, \qquad t_{off} = \tau \ln \frac{T_{hi} - 60}{T_{lo} - 60} \tag{1.12}$$

Both are $20 \ln(21/19) = 2.00$ min, a period of 4.0 min. Scanned every 0.1 min the switch acts one scan late, so the kettle swings from 78.95 to 81.05 °C with a period of 4.2 min, 14.3 starts of the second element an hour. Halving the band halves the swing and doubles the starts, to 27.3 an hour.

The band is what keeps the switching finite. With no band the element switches at every scan, 300 times an hour at a 0.1 min scan and 3000 at 0.01 min, so the rate is set by the scan and not by the plant. In continuous time it would switch infinitely often in a finite time, a Zeno solution. A measurement lag widens the swing beyond the band: a thermowell of 0.5 min gives 78.45 to 81.55 °C at 9.7 starts an hour. A minimum on and off time of 3 min, the protection a compressor needs, gives 78.49 to 81.50 °C at 10 starts an hour. Switching can also destabilize a plant whose every mode is stable, which is one reason for dwell times [26]. Relay oscillation in general is treated in Volume 7.

Safe switching in this volume rests on a few devices. Hysteresis and dwell bound the switching rate. A latch makes a protective switch stay until a condition is cleared and someone resets it. Tracking makes the controller that takes over start from the output in use. Persistence asks a condition to hold for a stated time before it counts. Chapter 12 runs a hysteresis thermostat on each case, stages compressors only after the suction has been outside a 2 K dead band for 15 s, enforces minimum run and rest times, and shows four thermostats that never exchange a signal falling into step through the suction pressure they share. Chapter 13 diverts at once on any of four conditions and returns to forward flow only after two tube volumes have flushed. Chapter 2 cuts the feed 2 K above its setpoint,

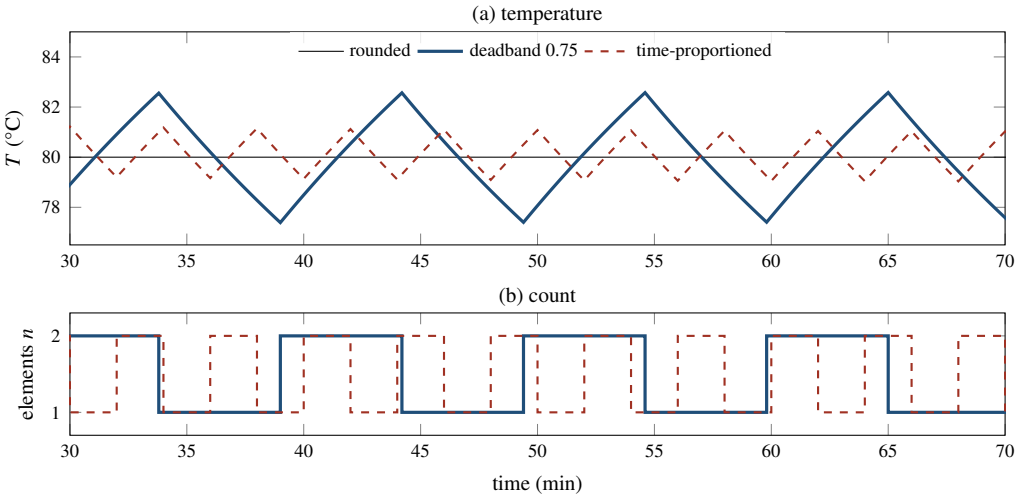


Figure 1.6 The Kettle Held at 80 °C by a PI Loop on Three Contactor Elements. (a) Temperature With the Demand Rounded (Thin Solid), Moved Only Past a Deadband of 0.75 Element (Thick Solid), or Time-Proportioned Over 4 min (Dashed). (b) The Element Count of the Last Two.

resumes it 0.3 K below that threshold, and latches its high-high interlock. Chapter 15 is a hybrid system by nature, smooth flight broken by impacts at which the ball's velocity jumps.

1.4.8 Integer Actuators

An actuator that comes in whole units, an element, a compressor, a server or a column, cannot take the value a PI loop asks for. Let the kettle's PI, with its 0.1 min period, compute a continuous demand of 1.5 elements at 80 °C and let contactors apply an integer count. Figure 1.6 shows three ways to do it. Rounding the demand changes the count at almost every scan, 300 times an hour, and holds 80 °C within 0.1 °C: it is the thermostat with no band. A deadband that moves the count only when the demand leaves the present count by more than 0.75 element changes it 11.5 times an hour and lets the kettle swing from 77.4 to 82.6 °C, and a deadband of 1.0 gives 5.9 changes and 74.9 to 85.0 °C. Time-proportioning switches the second element on for the demand's fractional part of each 4 min period, 30 changes an hour, and holds 79.0 to 81.2 °C. In each case the integral term finds the mean: with elements 10 % weak the deadband and time-proportioning hold 1.654 and 1.659 elements against the 1.667 required.

Every integer actuator trades swing against switching, and the right point depends on what a switch costs: wear on a contactor, a start on a compressor, money on a server. Chapter 14 orders servers only when its PI's demand leaves a deadband around the present count, adds a cooldown that stops flapping, and back-calculates the integral against the count actually applied, which also has a boot dead time. Chapter 12 stages two fixed compressors around a variable one, so that a continuous actuator absorbs what the integer ones cannot, and compares a scheduler that plans the starts. Volume 9 treats scheduling of integer inputs by optimization.

Table 1.2 Maps From One Cycle to the Next and Their Eigenvalues on the Kettle

Cycle and map	Eigenvalue μ	On the kettle
Batch, EWMA correction	$1-\lambda\xi$	0.7 or 0.3 for $\lambda = 0.3$ or 0.7, $\xi = 1$
Batch, learning update	$\mu(\omega)$ of Eq. (1.11)	0.1 to 0.98 with $\gamma = 1$, up to 1.036 with a short lead
Period, fixed duty	$e^{-t_p/\tau}$	0.8187 for $t_p = 4$ min
Period, minimum dwell	$e^{-2t_d/\tau}$	0.741 for $t_d = 3$ min
Period, thermostat	0	Back on the cycle after one switch
Bounce, impact map	Modulus of the restitution inside the window	Chapter 15

+ DID YOU KNOW? A Small Coil Switches a Large Motor

A contactor is a large relay whose contacts carry a motor's inrush current over many start and stop cycles. One rated at 75 horsepower on 480-volt three-phase power has a coil rated for 120 volts, so a small switch or a programmable controller can start the motor [27]. The kettle's contactors of Section 1.4.8 switch its elements the same way.

1.4.9 Cyclic Steady State

A repeated operation settles, if it settles at all, into a cycle rather than a point. Sampled once per cycle, the state follows a map $x_{c+1} = \Phi(x_c)$, and the cyclic steady state is its fixed point x^*. Near it a deviation is multiplied each cycle by the Jacobian of Φ, so the cycle is stable when every eigenvalue μ lies inside the unit circle, and $|\mu|$ says how fast it is reached [28]. For the time-proportioned element of Section 1.4.8 at a fixed 50 percent duty and a period $t_p = 4$ min, Equation (1.1) gives the map of the temperature at the start of each period, Equation (1.13).

$$T_{c+1} = e^{-t_p/\tau}\,T_c + 14.32 = 0.8187\,T_c + 14.32 \tag{1.13}$$

Its fixed point is a cycle from 79.0 to 81.0 °C, and a deviation falls to 10 % only after 12 periods, 48 min. The thermostat of Section 1.4.7 has the same cycle but a map with $\mu = 0$, because each switch happens at a fixed temperature and forgets how the kettle got there. After a 5 °C drop it needs one long on time, 5.5 min from 75 to 81 °C, and is back on its cycle. The difference goes further. With elements 10 % weak the fixed duty settles at 73.1 to 74.9 °C, while the thermostat keeps 79 to 81 °C at a duty of 0.67. A switch triggered by the state is feedback. A switch triggered by a clock is open loop.

Table 1.2 lists the cycle-to-cycle maps of this section, of Sections 1.4.5 and 1.4.6, and of the chapters that compute their own. The impact map of Chapter 15 is stable only while the paddle is slowing at impact, and over most of that window its eigenvalues have a modulus equal to the ball's restitution. In Chapter 10 heat accumulates from layer to layer, so the layers repeat only as far as the part's temperature does. Chapter 11 rotates three columns through a cycle whose feasibility it checks at each switch, and in Chapter 12 two cases on one suction header close a gap of 0.39 of a cycle to 0.10 by their fifth cut-in, a phase map that attracts.

1.4.10 Strategies and Safe Switching in Each Chapter

The sections above give each chapter two questions. How is the product brought to its end: by a trajectory, by an endpoint rule, by a correction from one run to the next, or by learning? And where the plant switches, what keeps the switching safe? Table 1.3 lists the answers each chapter tests, as groundwork for the comparison in Chapter 16. The measured results are left to the chapters.

Table 1.3 What Each Chapter Judges at the End, the Strategies It Compares and What Keeps Its Switching Safe

Ch.	Judged at the end	Strategies compared	Switching kept safe by
2	The whole dose, with the temperature after a cooling failure below a limit throughout	Fixed, temperature-limited and accumulation-limited dosing, one-step optimizer, MPC	Feed cut with hysteresis, latched interlock
3	Amount and purity in the receiver	Constant reflux, constant composition, optimized ramp, MPC	Policy change with tracking
4	Size and fines of the filtered crystals	Three cooling curves, cascade, supersaturation feedback, MPC	Moving output limit, transfer on a change of structure
5	Every vial dry, none collapsed	Fixed recipe, probe-limited, endpoint-aware with a learned resistance, MPC	Endpoint rules held for a set time
6	Cure of the slowest part, exotherm of the hottest core	Air loop, lead/lag selector, observer feedforward, optimized recipe, MPC	Selector with external reset
7	Mass of every part	Phase sequencer, shot-to-shot correction	Switchover by tracking, time limit behind each trigger
8	Four layer thicknesses	Timer, integrated estimate, estimate corrected by a load cell	Ramped gas changes, timeout fault
9	Fill of every vial	Pooled, per-head and record-based corrections	Validity window, step limit
10	Melt pool width along every layer	Feedback, learning, both	Q-filter, profile kept or reset at a new outline
11	Antibody pooled per liter of resin and hour, antibody lost	Time, load, breakthrough and learned-capacity triggers	Switch only when the next column is ready, pressure override
12	Goods temperature, energy, starts	Thermostats, staging, floating pressures, hybrid MPC	Band, minimum run and rest, desynchronization
13	Hold of every parcel	Cascade with feedforward under a supervisor	Diversion at once, return after a flush
14	Latency percentile and cost	Platform rule, PI, forecast floor, MPC	Deadband, cooldown, admission control
15	The period-1 bounce	Open-loop sinusoid, once-per-bounce PI, mirror law	Paddle slowing at impact, inside the window

DID YOU KNOW? A Quarter of Batch Control on Personal Computers

Batch operations run on stand-alone controllers, programmable logic controllers, distributed control systems and personal computers. A quarter of batch process control has been reported to involve personal computers, often in combination with the other platforms [6].

1.5 Instruments and Final Elements in This Volume

Figure 1.7 lists what the controllers of each chapter read and what they move. Temperature and pressure are each read in seven chapters, and a modulating valve or a flow loop is a final element in nine. The quality that decides the product is seldom among the readings, and only three chapters read a composition. Chapter 8 infers four layer thicknesses from a thermocouple and a load cell, and Chapter 2 infers the reagent that has piled up from two temperatures and a feed flow.

Weighing recurs in this volume. A load cell is one or more strain gauges bonded to a metal structure of precisely known elastic properties, so that the force on it becomes a change of electrical resistance [27], as Figure 1.8 shows. A scale closes the loop between shots of Chapter 7, tare cells and a checkweigher the loop between vials of Chapter 9, and a load cell under the coater the bed-mass correction of Chapter 8.

Many final elements here have only two states. A discrete control element has a limited number of

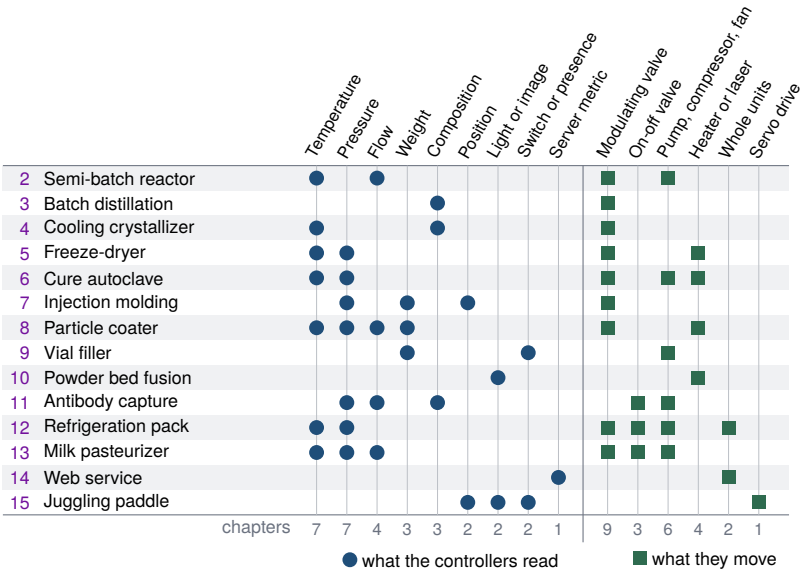


Figure 1.7 What the controllers of each chapter read (round marks) and move (square marks), from each chapter's variables table and technology section. Pressure includes differential pressure, a flow loop counts as a modulating valve, a heater includes the freeze-dryer's thermal unit, and a computed value, such as an inferred thickness, is not marked. Notice the scales, switches and whole units of the repeated and switched plants.

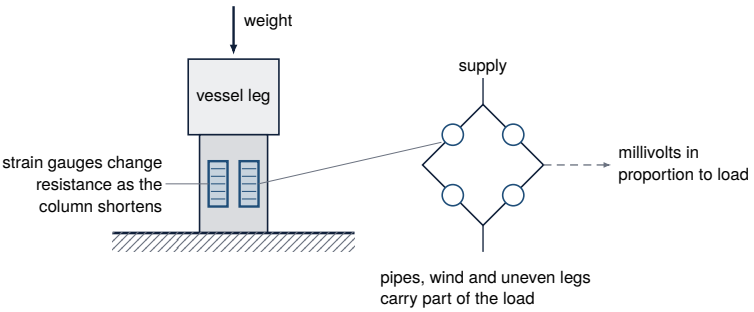


Figure 1.8 Principle of a strain-gauge load cell under a vessel leg. Notice that pipes, wind and uneven legs can carry part of the load past the cell, which then reads less than the true weight.

states, usually on and off, and the solenoid valve, a coil that pulls a ferrous armature against a return spring, is a very common form of on-off valve [27]. The case valves of Chapter 12, the diversion device of Chapter 13 and the valve block of Chapter 11 switch between positions in this way, and the staged compressors and the servers of Chapter 14 come in whole units, the integer actuators of Section 1.4.8.

The controllers sit in the layers that the batch control standards name. Their functional model relates recipe management, scheduling, sequential control, regulatory control and safety interlock systems [29]. Batch operations are controlled from stand-alone controllers, programmable logic controllers, distributed control systems and personal computers [6], and supervisory and lower-level loops run with cycle times from milliseconds to minutes [30]. Figure 1.9 places the loops of this volume on their platforms.

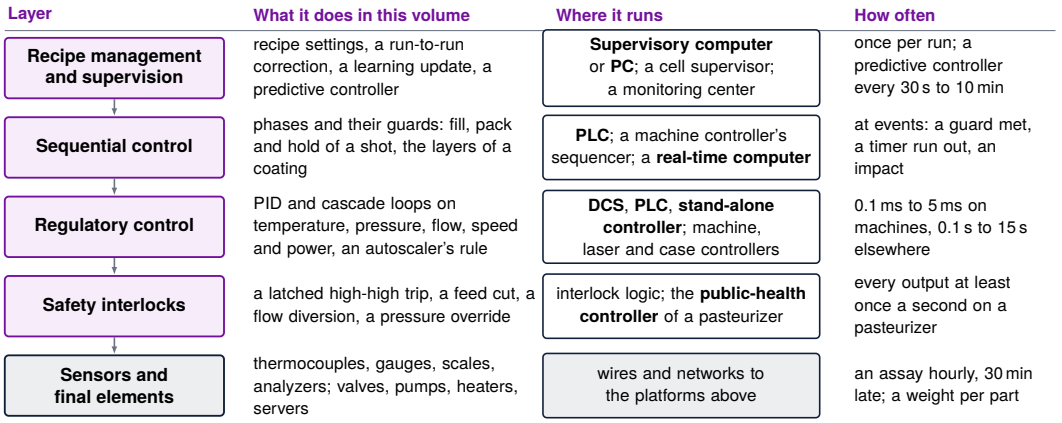


Figure 1.9 The control layers of this volume, named after the functional model of the batch control standards, the platform that hosts each and how often each acts, from [6, 29, 30] and the chapters' technology sections. Notice that a correction between runs acts once per run, however fast the loops beneath it.

Table 1.4 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.

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 1.4 lists what a reader meets.

1.6.1 The Simulation Engine

Each app carries a plant model written as ordinary differential equations $\frac{dx}{dt} = f(x,u,d,p)$ 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 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

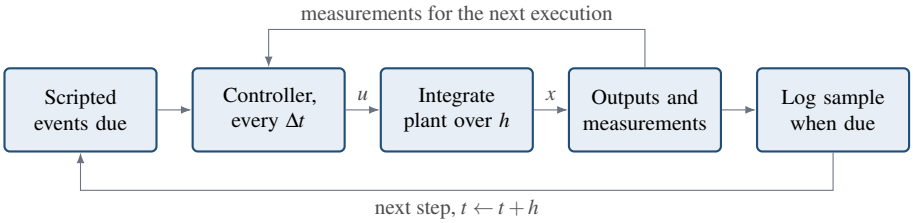


Figure 1.10 Work of the Engine in One Integration Step

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 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 (1.14) to (1.17) for execution k and control period Δt.

$$e_k = a\,(\mathrm{SP}_k - \mathrm{PV}_k), \qquad P_k = K_c\, e_k, \qquad D_k = -a\,K_c\,\tau_D\,\frac{\mathrm{PV}_k^f - \mathrm{PV}_{k-1}^f}{\Delta t} \tag{1.14}$$

$$S_k = S_{k-1} + (P_k - P_{k-1}) + (D_k - D_{k-1}) \tag{1.15}$$

$$v_k = u_b + S_k + I_{k-1}, \qquad u_k = \mathrm{sat}(v_k) \tag{1.16}$$

$$I_k = I_{k-1} + \left[\frac{K_c}{\tau_I}\,e_k + \frac{u_k - v_k}{T_t}\right]\Delta t \tag{1.17}$$

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 (1.17) drives I toward the value that reproduces the output applied, which is back-calculation [31]. With no integral action,

Equation (1.17) 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 $I_k = u_{\text{man}} - u_b - S_k$, 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 (1.18).

$$x_{k+1} = A\,x_k + B\,u_k + E\,w_k + c, \qquad y_k = C\,x_k \tag{1.18}$$

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 (1.19) over a horizon of N steps.

$$J = \sum_{k=1}^{N} (y_k - r_k)^{\top} Q\,(y_k - r_k) + \sum_{j} \Delta u_j^{\top} R\,\Delta u_j + \sum_{i} \rho_i\, s_i^2 \tag{1.19}$$

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.

Table 1.5 Roadmap of the Volume

Ch.	Plant	The lesson	Level	Time
2	Semi-batch exothermic reactor	Accumulated reagent as a path constraint. Dosing as the safety layer.	intermediate	3 h
3	Batch distillation column	No operating point. An open-loop reflux trajectory against a composition loop.	intermediate	3.5 h
4	Seeded cooling crystallizer	The shape of the trajectory decides the product, not the tuning.	intermediate	3 h
5	Pharmaceutical freeze-dryer	Hottest vials set the limit, slowest the endpoint. An estimator that learns.	advanced	2.5 h
6	Composite cure autoclave	Lead/lag selection among twelve thermocouples, an exotherm the selector cannot stop.	advanced	2.5 h
7	Injection molding machine	Bumpless switchover inside each shot, correction between shots.	introductory	2 h
8	Fuel particle coater	A recipe as a state machine. Phases ended by an estimate and its failures.	intermediate	3 h
9	Four-head vial filler	Run-to-run correction per head with results in flight.	introductory	2 h
10	Laser powder bed fusion	Learning a power profile layer by layer, and why the Q-filter matters.	introductory	2 h
11	Three-column antibody capture	When to switch a column whose capacity is unmeasured. A cyclic schedule.	advanced	2 h
12	Supermarket refrigeration pack	Hysteresis thermostats that synchronize, staged compressors.	intermediate	2.5 h
13	Milk pasteurizer	Product history, regeneration and a diversion supervisor.	introductory	2 h
14	Autoscaled web service	Integer servers with a boot dead time, flapping and retry storms.	intermediate	2.5 h
15	Juggling robot paddle	A hybrid orbit stable through its impact map, with or without a sensor.	advanced	2.5 h

- *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.

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 16.

1.8 Chapter Roadmap

Table 1.5 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 kind of operation that Section 1.1 sorts them into.

One batch along a trajectory	Repeated runs corrected from the last	Switched actuators or dynamics
Sections 1.4.2 to 1.4.4	Sections 1.4.2, 1.4.5, 1.4.6	Sections 1.4.7 to 1.4.9
(2) unreacted reagent as a path limit	(7) switchover inside a shot, correction between shots	(12) thermostats that fall into step
(3) an open-loop reflux trajectory against a loop	(8) a recipe whose phases end on an estimate	(13) a diversion supervisor on product history
(4) the shape of the cooling curve decides the product	(9) a correction per head with results in flight	(14) integer servers with a boot dead time
(5) hottest vials set the limit, slowest the end	(10) a power profile learned layer by layer	(15) a hybrid orbit stable through its impacts
(6) a selector among twelve thermocouples	(11) when to switch a column of unmeasured capacity	

Figure 1.11 Method map of the volume: each chapter by the kind of operation of Section 1.1, with its lesson from Table 1.5 and the subsections it rests on. Notice that the recipe sections serve both the single batch and the repeated run.

Chapters 7, 9, 10 and 13 form a first course: a sequencer with a switchover, a run-to-run correction, a learning update and a supervisor, each on a plant that is quick to run. They rest on Sections 1.4.2, 1.4.5, 1.4.6 and 1.4.7. Chapters 2 to 6 are best read after Sections 1.4.3 and 1.4.4, in any order, and Chapter 8 after them. Chapters 12, 14 and 15 rest on Sections 1.4.7 to 1.4.9, and Chapter 11 draws on every section. A reader who knows batch operation can take a shorter path through Chapters 2, 3, 9, 10 and 15.

KEY POINTS

- A batch has no steady state, and its product is judged once, after the end.
- Each phase ends on a clock, a measured event or an estimate.
- Correcting the next run from the last pays only against disturbances that persist.
- A switch triggered by the state is feedback, and one triggered by a clock is open loop.

Where else this applies. Wafers corrected run to run in Volume 14, Chapter 6, and a tap changer behind a deadband in Volume 17, Chapter 8.

References

[1] D. Bonvin, "Optimal operation of batch reactors—a personal view," *Journal of Process Control*, vol. 8, no. 5–6, pp. 355–368, 1998.

[2] D. Bonvin, B. Srinivasan, and D. Hunkeler, "Control and optimization of batch processes," *IEEE Control Systems Magazine*, vol. 26, no. 6, pp. 34–45, 2006.

[3] J. H. Lee and K. S. Lee, "Iterative learning control applied to batch processes: An overview," *Control Engineering Practice*, vol. 15, no. 10, pp. 1306–1318, 2007.

[4] L. Rayleigh, "On the distillation of binary mixtures," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 4, no. 23, pp. 521–537, 1902.

[5] H. A. Miers and F. Isaac, "The refractive indices of crystallising solutions, with especial reference to the passage from the metastable to the labile condition," *Journal of the Chemical Society, Transactions*, vol. 89, pp. 413–454, 1906.

[6] D. Bonvin and G. François, "Control and optimization of batch chemical processes," in *Coulson and Richardson's Chemical Engineering*. Elsevier, 2017, pp. 441–503.

[7] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[8] J. W. Hou, "Optimal cure cycle design for autoclave processing of thick composites laminates: A feasibility study," Old Dominion University, for NASA Langley Research Center, Norfolk, VA, Progress Report, NASA Contractor Report NASA-CR-181261, Nov. 1985, NASA Technical Reports Server 19870018301.

[9] M. Buhler, D. E. Koditschek, and P. J. Kindlmann, "A family of robot control strategies for intermittent dynamical environments," *IEEE Control Systems Magazine*, vol. 10, no. 2, pp. 16–22, 1990.

[10] E. Sachs, A. Hu, and A. Ingolfsson, "Run by run process control: Combining SPC and feedback control," *IEEE Transactions on Semiconductor Manufacturing*, vol. 8, no. 1, pp. 26–43, 1995.

[11] T. A. Henzinger, "The theory of hybrid automata," in *Proceedings 11th Annual IEEE Symposium on Logic in Computer Science*, 1996, pp. 278–292.

[12] U.S. Food and Drug Administration, "Guidance for industry. PAT: A framework for innovative pharmaceutical development, manufacturing, and quality assurance," U.S. Department of Health and Human Services, Food and Drug Administration, Rockville, MD, Tech. Rep., Sep. 2004.

[13] C. Sonntag, A. Devanathan, and S. Engell, "Hybrid NMPC of a supermarket refrigeration system using sequential optimization," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 13 901–13 906, 2008.

[14] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-145, 2011.

[15] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use, "ICH harmonised guideline Q13: Continuous manufacturing of drug substances and drug products," Adopted 16 November 2022, 2022. [Online]. Available: https://database.ich.org/sites/default/files/ICH_Q13_Step4_Guideline_2022_1116.pdf

[16] A. Ingolfsson and E. Sachs, "Stability and sensitivity of an EWMA controller," *Journal of Quality Technology*, vol. 25, no. 4, pp. 271–287, 1993.

[17] 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.

[18] B. Srinivasan, S. Palanki, and D. Bonvin, "Dynamic optimization of batch processes: I. characterization of the nominal solution," *Computers & Chemical Engineering*, vol. 27, no. 1, pp. 1–26, 2003.

[19] E. Del Castillo and A. M. Hurwitz, "Run-to-run process control: Literature review and extensions," *Journal of Quality Technology*, vol. 29, no. 2, pp. 184–196, 1997.

[20] J. F. MacGregor, "A different view of the funnel experiment," *Journal of Quality Technology*, vol. 22, no. 4, pp. 255–259, 1990.

[21] R. P. Good and S. J. Qin, "On the stability of MIMO EWMA run-to-run controllers with metrology delay," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 1, pp. 78–86, 2006.

[22] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[23] M. Norrlöf and S. Gunnarsson, "Time and frequency domain convergence properties in iterative learning control," *International Journal of Control*, vol. 75, no. 14, pp. 1114–1126, 2002.

[24] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *International Journal of Control*, vol. 73, no. 10, pp. 930–954, 2000.

[25] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 28–93, 2009.

[26] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Systems Magazine*, vol. 19, no. 5, pp. 59–70, 1999.

[27] 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.

[28] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, ser. Applied Mathematical Sciences. Springer, 1983.

[29] International Society of Automation, "ISA88: Batch control, committee scope and purpose," www.isa.org, standards committees, 2026, accessed 22 November 2026.

[30] 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.

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

[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 Theory and calculation · 4 min

The kettle's dose is Equation (1.2), in which one minute at 80 °C counts as one minute and each 10 °C hotter counts ten times as much.

(a) Find the dose of a hold of 6 min at 85 °C. *Checkpoint:* 18.97 min.

(b) Find the dose of a heat-up tracked exactly from 20 to 80 °C at 4 °C/min. *Checkpoint:* 1.09 min.

(c) Find how long a hold at 75 °C must last to deliver 30 min of dose. *Checkpoint:* 94.9 min.

1.2 Concept check · 2 min

The kettle's thermocouple reads 1 °C high. The hold can end on a clock set to the nominal 26.3 min or on the dose estimate with the predicted cooling dose, as in Section 1.4.2. Which statement is right?

(a) ending on the dose estimate removes the error, because the estimate integrates the measured temperature

(b) neither guard removes it, because both rely on the same reading, so only a test of the product shows it

(c) ending on the clock removes the error, because a clock does not depend on the thermocouple at all

(d) either guard removes it, because the PI loop's integral action cancels a constant measurement bias

1.3 Theory and calculation · 5 min

The kettle of Equation (1.1) is heated on a ramp by a PI loop with $\tau_I = \tau = 20\,\text{min}$ and $K_c K = 5$.

(a) Find the closed-loop time constant and the lag on a ramp of 1.5 °C/min. *Checkpoint:* 4 min and 6 °C.

(b) Find the feedforward of Equation (1.4), with $\hat{K} = K = 40\,°\text{C}$ per element, when that ramp's setpoint passes 50 °C. *Checkpoint:* 1.5 elements.

1.4 Theory and calculation · 5 min

The product's limit is lowered to 87 °C, and the fast policy of Section 1.4.4 keeps the same 2 °C back-off.

(a) Find the time to heat from 20 to the switching temperature with all three elements on. *Checkpoint:* 15.60 min.

(b) Find the holding input at the switching temperature. *Checkpoint:* 1.625 elements.

(c) Find the time of free cooling from the switching temperature to 40 °C. *Checkpoint:* 23.57 min.

1.5 Theory and calculation · 6 min

An EWMA controller, Equation (1.7), runs with $\lambda = 0.4$ and a model gain $b = 1$.

(a) With $\beta = 1$, a step of -2 in d arrives. Find the error left after three corrections. *Checkpoint:* -0.432.

(b) Find the steady offset under a drift of -0.2 per batch. *Checkpoint:* -0.5.

(c) Find the standard deviation of the result when the only disturbance is white noise with a standard deviation of 0.5. *Checkpoint:* 0.559.

(d) The true gain is $\beta = 1.5$. Find the factor by which a step shrinks each batch, and the largest λ that keeps the loop stable. *Checkpoint:* 0.4 and 1.33.

1.6 Concept check · 2 min

A filler's results arrive one batch late, and its correction is the integral form $u_{j+1} = u_j + (\lambda/b)(y^* - y_{j-1})$ with $\lambda\xi = 0.8$. Which statement is right?

(a) it is unstable, because 0.8 exceeds the limit of 0.618 for a delay
(b) it is stable and monotone, because 0.8 is below the limit of 2
(c) it is unstable, because any delay destabilizes an integral correction
(d) it is stable, with an error that oscillates and shrinks by 0.89 per batch

1.7 Theory and calculation · 4 min

A learning update, Equation (1.10), with $\ell = 1$ and no filter acts on the kettle sampled every minute, with the model gain exact, so that the factor of Equation (1.11) is $1 - \gamma$ at zero frequency and $1 - \gamma(1-p)/(1+p)$ at the shortest period, with $p = \mathrm{e}^{-1/20}$.

(a) Find both factors for $\gamma = 0.5$. *Checkpoint:* 0.5 and 0.9875.

(b) Find the factor at zero frequency for $\gamma = 1.5$ and say what it means for a constant error. *Checkpoint:* -0.5.

1.8 Theory and calculation · 4 min

The kettle holds 80 °C with one element always on and the second switched by a thermostat with $T_{lo} = 78\,^{\circ}\mathrm{C}$ and $T_{hi} = 82\,^{\circ}\mathrm{C}$. Ignore the scan and any measurement lag.

(a) Find the on and off times of the second element. *Checkpoint:* 4.01 min each.

(b) Find the period and the number of starts of the second element per hour. *Checkpoint:* 8.03 min, 7.47.

1.9 Concept check · 2 min

The kettle's elements have aged and deliver 10 % less heat. Its hold runs either on the thermostat of Section 1.4.7, with a band from 79 to 81 °C, or with the second element on for a fixed half of every 4 min period. Which statement is right?

(a) the thermostat still holds 79 to 81 °C, because each switch happens at a measured temperature
(b) the fixed duty still holds 79 to 81 °C, because a clock does not depend on the elements
(c) both still hold 79 to 81 °C, because both had that cycle with nominal elements
(d) neither holds the band, because only integral action can compensate a change of gain

1.10 Theory and calculation · 5 min

The second element is time-proportioned at a fixed 50 % duty with a period of 6 min.

(a) Find the eigenvalue of the map from one period's starting temperature to the next, and the number of periods that shrink a deviation to 10 %. *Checkpoint:* 0.741 and 8.

(b) Find the cyclic steady state, the lowest and highest temperature of the cycle. *Checkpoint:* 78.50 °C and 81.50 °C.