INTERACTIVE CONTROL ENGINEERING
Principles and Applications

Volume 5 Delays, Slow Measurements, and State Estimation

CHAPTER 1

Introduction: Delays, Late Measurements, and State Estimation

Pages 1–23

Edited by John D. Hedengren

How to cite this chapter. "Introduction: Delays, Late Measurements, and State Estimation," in *Delays, Slow Measurements, and State Estimation*, John D. Hedengren, Ed., vol. 5 of *Interactive Control Engineering: Principles and Applications*. Advanced Process Solutions, LLC, 2026, ch. 1, pp. 1–23.

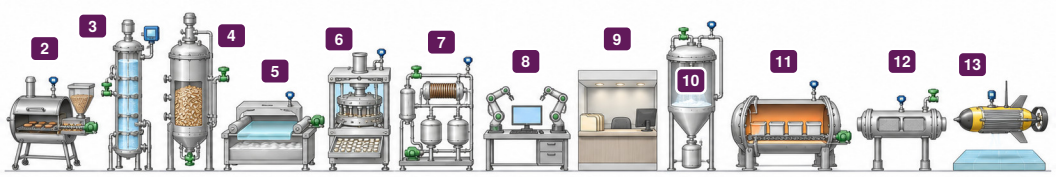
The page numbers in this file are those of the printed volume.

The volume online: contents, other chapters and citation records
https://apmonitor.com/apps/books/v05-delays-estimation/

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

CHAPTER 1

Introduction: Delays, Late Measurements, and State Estimation



The chapters 2 pellet grill 3 distillation column 4 kraft digester 5 float glass line 6 tableting line 7 flow synthesis 8 teleoperator 9 pension fund 10 spray dryer 11 cathode kiln 12 diesel urea doser 13 survey vehicle

Notice that the units from the grill to the flow synthesis skid carry their product along a line before it is measured, while the kiln, the exhaust canister and the survey vehicle each hide a variable that no instrument reads. Illustration, not to scale.

IN PLAIN TERMS **Acting on News That Arrives Late**

The main idea. A controller can only fix what it has seen, and it often sees late. Material takes time to reach the meter, a test takes time to finish, and some things inside a plant are never measured at all.

The challenge. Acting hard on old news makes a loop swing, and waiting makes it slow. A copy of the plant kept in a computer can fill the gap, but only as well as the copy matches the plant.

Why it matters. The same few ideas steer a grill, a pension fund, a robot hand and a vehicle under the sea.

THREE WAYS TO READ THIS CHAPTER

Explore	7 min	This page, Section 1.2 with its timeline and two History boxes, the Did you know? boxes in Sections 1.4.5, 1.4.6 and 1.4.8, Figure 1.10, and Key points in Section 1.8. No equations.
Study	45 min	The whole chapter in order, with each number of the dosing example worked, then Exercises 1 to 10.
Practice	5 min	This page, Section 1.4.9 with Tables 1.3 and 1.4, Section 1.5 with Figures 1.6 and 1.8, and Section 1.8 with Figure 1.10.

1.1 The Question This Volume Answers

A feedback loop can correct only what it has seen, and it sees the plant late. Material must travel from the actuator to the point of measurement, an analyzer must draw a sample and work on it, and a laboratory reports an hour or a day after the bottle was filled. Some variables are never measured at all, such as the temperature inside a brick, the ammonia stored on a catalyst or the position of a vehicle under water. This volume asks what each delay and each missing measurement costs, and how much of that cost a controller can win back with a model.

Two answers recur. The first is prediction. A model of the plant tells the controller what its own past moves will do before the measurement shows it, which is the idea behind the Smith predictor, internal model control and every predictive controller. The second is estimation. A model run beside the plant and corrected by whatever measurements exist supplies the variables that are late or missing. Kalman filters, moving horizon estimators, soft sensors and laboratory bias updates are forms of this idea. Both answers fail in the same way, when the model is wrong in a direction that the measurements cannot reveal.

Twelve chapters test these ideas, each on a plant with an app that runs in a browser. Chapter 2 and Chapter 3 control one variable through a fuel delay and through a slow analyzer. Chapter 4 and Chapter 5 decide a product quality hours or minutes before an instrument can read it. Chapter 6 and Chapter 7 follow material through pharmaceutical lines whose delays change with throughput.

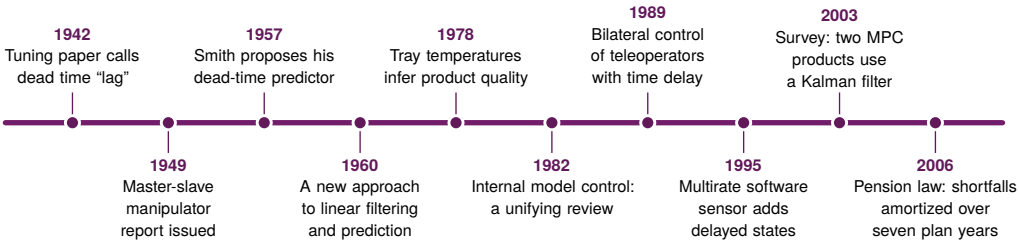


Figure 1.1 Milestones of prediction, estimation and late measurement in the systems of this volume. The dates are from [4] (1942), [5] (1949), [6] (1957), [7] (1960), [8] (1978), [9] (1982), [10] (1989), [11] (1995), [12] (2003) and [13] (2006). Notice that a remote hand and a pension law sit among the plants and the theory: each meets the same late news. Events are evenly spaced, not to a time scale.

Chapter 8 closes a force loop over a delayed channel, and Chapter 9 decides a contribution once a year on a valuation a year old. Chapter 10 and Chapter 11 control a moisture and a temperature that only a model can see, and Chapter 12 and Chapter 13 estimate a stored mass and a position from sensors that cannot report them directly.

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

- State the least error that any controller can leave after a load through a known dead time, and compare a tuned loop against it.
- Build a Smith predictor, say when it pays, and keep its delay right when the delay changes with flow.
- Account for a sampled or late measurement as added dead time, and choose the gain of a controller that acts once per result.
- Test whether a set of measurements can distinguish the disturbances and biases that matter.
- Design a Kalman filter or a laboratory bias update, align a late result with the instant it describes, and choose its weight.

This chapter develops each skill on one small example and names the chapters that use it. PID tuning is not repeated here. Volume 2 develops it, and every loop below is a PI controller tuned by the rules of that volume. Standard texts treat dead-time compensation and state estimation in fuller detail [1, 2].

1.2 A Short History

The milestones of Figure 1.1 follow prediction and estimation through the kinds of system in this volume. Dead time was read from plant tests early. The second tuning method of a landmark paper of 1942 stepped the valve by hand and read the dead time from the response, though the paper called it lag, a word now kept for a first-order response [3, 4]. Prediction and estimation, the two answers of Section 1.1, then appeared within three years of each other, the predictor in 1957 and the filter in 1960. The two History boxes of this section follow each from its start.

Remote handling met the delay from the other end. A report of 1949 described a master-slave manipulator [5], and by 1989 the bilateral control of teleoperators with time delay was a subject of its own [10]. In the process industries the missing measurement was met with inference. A design of 1978 inferred unmeasured product qualities from secondary measurements, and on a simulated debutanizer four to six tray temperatures improved the steady-state control by as much as 400 % [8]. Internal model control was reviewed as a unifying structure in 1982 [9]. A multirate software sensor of 1995 took process outputs sampled at different rates into an iterated extended Kalman filter and carried the delayed ones as added states [11], the first of the three treatments of Section 1.4.7.

> ⧖ HISTORY **1960 Onward: The Filter That Plants Passed By**
>
> A paper of 1960 recast filtering and prediction in the state-transition form and derived an equation for the covariance of the optimal estimation error [7]. The filter, joined to a state feedback, soon became a standard approach in a wide range of fields, with an estimated 400 patents a year based on it. It had little impact in the process industries, whose units are constrained, nonlinear and hard to model, and whose engineers often had no exposure to its concepts or regarded them as impractical. A survey of 2003 found only two predictive control products that used a Kalman filter. The rest fed back a bias, the measured output minus the predicted one, which assumes a step disturbance at the output that stays constant [12]. Section 1.4.7 shows that bias as the simplest filter of all.

Late measurement is not confined to plants. A law approved in 2006 has a pension plan amortize each funding shortfall in level annual installments over seven plan years [13], and Chapter 9 spreads its deficit over a period that the reader chooses.

> ⧖ HISTORY **1957 Onward: From One Delay to Many**
>
> The predictor that Smith proposed in 1957 splits the model into a fast part without delay and a delay, and so removes the delay from the closed-loop characteristic equation when the model is exact. It was proposed for plants that are stable in open loop, and later work extended it to integrating and unstable ones. Plants with several inputs and outputs were harder, because each path from an input to an output may have its own delay. A design of 1973 assumed one delay on every channel. It kept the predictor's properties but is in general not very useful in practice. Later designs for many delays had to give up either the prediction of the output or a characteristic equation free of delay [6]. Chapter 5 uses the single-loop form on its thickness.

1.3 Notation and Conventions

The chapters share the symbols of the notation page. The setpoint is SP or r, the process variable is PV and the manipulated variable is MV, with controller settings K_c and τ_I. A process model has a gain K_p, a time constant τ_p and a dead time θ_p. In the running example the tank's time constant is τ and the pipe's delay is θ, and both change with the flow Q. A hat marks an estimate, so $\hat{x}$ is the estimate of a state x. A measurement drawn at the instant t_s is reported at $t_s + t_d$, where t_d is the reporting delay, and samples are drawn every T_s. The weight of a bias update is λ.

Two symbols need care across the series. The shared framework uses Q and R as the weights of the predictive controller, so this chapter writes the noise covariances of a Kalman filter as W for the process and V for the measurement, and a scalar pair of variances as q and r_v. The time constant τ_c of an IMC filter is the closed-loop time constant of the SIMC rule, and λ is reserved for the weight of a bias update. It also uses a for the controller action and h for the integration step, so neither letter appears here for anything else. The chapters keep their own notation. Chapter 2 writes θ_t for the auger's transport and θ_m for the delay in its controller's model, Chapter 3 writes τ_a for the analyzer lag and δ for its disturbance state, and Chapter 9 writes θ_v for the age of the valuation. Chapter 8 writes T for the round trip of its channel, and Chapter 13 writes b for the compass bias.

1.4 Delays, Late Measurements, and Estimators

1.4.1 The Running Example

The running example is a dosing station that makes a solution of fixed strength. Water flows at $Q = 1\,\mathrm{m}^3/\mathrm{min}$ through a well-mixed tank of $V_t = 5\,\mathrm{m}^3$, where a pump adds a concentrate in ratio to the measured flow. The dose ratio u, in percent, is the MV. Because the pump follows the flow, each percent adds 1 g/L at any flow, so $K_p = 1\,\mathrm{g/L}$ per percent. The solution then travels through a pipe of $V_p = 5\,\mathrm{m}^3$ to the point where the product is sampled and shipped. The tank outlet concentration x follows Equation (1.1), and the product y at the sample point is the tank outlet that left the tank one

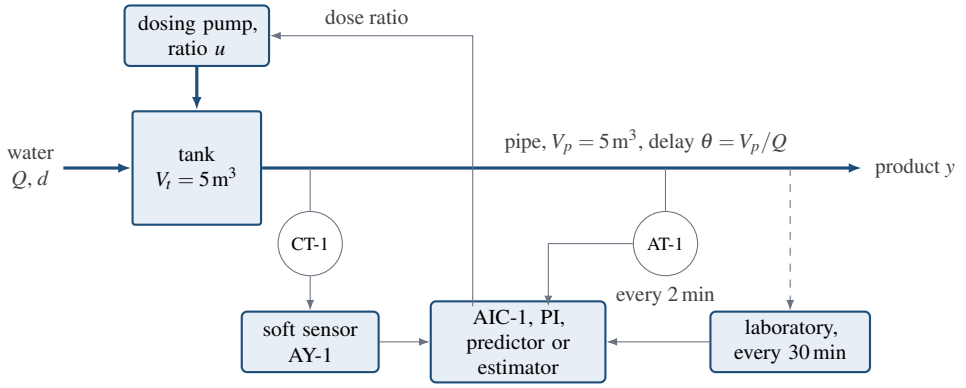


Figure 1.2 The Running Example. A Tank Doses a Concentrate Into a Water Stream, and a Pipe Carries the Product to the Sample Point. The Conductivity Cell CT-1 Reads the Tank Outlet at Once, the Analyzer AT-1 Reads the Sample Point Every 2 min With a Result One Cycle Later, and the Laboratory Reports a Sample From the Same Point 30 min After It Is Drawn.

pipe volume earlier.

$$\tau \frac{\mathrm{d}x}{\mathrm{d}t} = -x + K_p u + d, \qquad \tau = \frac{V_t}{Q}, \qquad y(t) = x(t-\theta), \qquad \theta = \frac{V_p}{Q} \tag{1.1}$$

where d, in g/L, is solute that arrives already dissolved in the water, a load disturbance. At the design flow $\tau = \theta = 5\,\mathrm{min}$, and the design dose of 50 % gives 50 g/L. The plant is first order plus dead time, with a lag and a delay of equal size.

Three instruments see the product, as shown in Figure 1.2. A conductivity cell CT-1 at the tank outlet reads at once, but it reads the dissolved salts of the water as well, so the soft sensor AY-1 that turns it into a concentration is right only while the water stays the same. An analyzer AT-1 at the sample point draws a sample every 2 min and reports it one cycle later. A laboratory receives a bottle from the sample point every 30 min and reports 30 min later, with no bias. The controller AIC-1 is a PI controller in the velocity form of Section 1.6.2, executed every 0.1 min, and the plant is integrated on a step of 0.01 min. Unless a subsection says otherwise, AIC-1 reads an ideal instrument at the sample point, so that the pipe is the only delay. Tuned by the SIMC rule of Volume 2 with $\tau_c = \theta$, it has $K_c = 0.5\,\%$ per g/L and $\tau_I = 5\,\mathrm{min}$.

1.4.2 Dead Time and Achievable Performance

A dead time sets a floor under the error that any controller can achieve. Suppose the water's solute rises by d at $t = 0$. The tank outlet starts to rise at once, but the product shows nothing until θ. Even a perfect controller that recognizes the upset at that instant and cancels it in the tank at once cannot touch the product before 2θ. The product therefore repeats the tank's open-loop rise for one dead time, and the integral of its absolute error (IAE) is at least Equation (1.2).

$$\mathrm{IAE}_{\min} = K_p d \left[\theta - \tau\left(1 - \mathrm{e}^{-\theta/\tau}\right)\right] \approx K_p d \frac{\theta^2}{2\tau} \quad (\theta \ll \tau) \tag{1.2}$$

For the example this is 1.84 g min/L per g/L of upset, and the product rises by at least 0.63 g/L. A setpoint step of size Δr has the floor $\theta\,\Delta r$, since nothing can reach the product sooner.

A PI controller with integral action pays more. Its integral must move the dose by d/K_p in the end, and in the velocity form that move equals K_c/τ_I times the integrated error, so the integrated error (IE) after a load follows Equation (1.3).

$$\mathrm{IE} = \frac{\tau_I}{K_c K_p} d = 2 K_p d\,\theta \quad \text{for SIMC with } \tau_c = \theta,\ \tau_I = \tau \tag{1.3}$$

The IAE equals the IE when the error keeps one sign and exceeds it when the loop overshoots. On the example the SIMC loop leaves an IAE of 10.25 g min/L, 5.6 times the floor, against an IE of exactly 10. The cost of a PI controller grows in proportion to the dead time, while the floor grows with its square while θ is short against τ. Repeating the run at dead times of 1, 2.5, 5 and 10 min with the tank unchanged gives an IAE of 2.00, 5.00, 10.25 and 21.4 g min/L against floors of 0.094, 0.53, 1.84 and 5.68, ratios of 21, 9.4, 5.6 and 3.8. A short delay is cheap in absolute terms, but it is where a PI controller leaves the most on the table relative to what is possible. A long delay costs every controller dearly.

A dead time also caps the bandwidth of the loop, near $1/\theta_p$, as Volume 2 shows from the phase margin. The same cap appears in forms that do not look like a PI loop. In Chapter 8 the loop is a force reflected through a channel with round trip T. Force reflected through a delay can make a contact chatter although the hand, the devices and the wall are each passive [10]. Through a long delay the loop's phase passes -180° at frequencies only $2\pi/T$ apart, so it stays stable at every long delay only if its gain is below one at all frequencies. A wall can then be rendered no stiffer than the hand that holds it, and the largest reflection gain falls from 1.00 at a round trip of 10 ms to 0.211 at 2.6 s. In Chapter 9 the decision interval and the valuation lag play the part of the dead time. Chapter 4 shows the limiting case, in which chips already past the heater when a production change arrives can no longer be reached, and no setting of the heater holds its kappa band through the step.

A fitted dead time is not always a transport delay. Chapter 2 fits a dead time of 4.7 min by the two-point method and 2.8 min by the tangent to a grill whose auger delay is 20 s, because the ignition, burning and the steel add lags in series. Chapter 3 fits 0.21 min to a column that contains no delay at all, the effect of thirty fast modes rounding the foot of the response. Chapter 11 finds no transport delay on its zone loops, so the SIMC rule cannot take $\tau_c = \theta_p$ there, and the scan and the interaction between zones limit the loop instead.

1.4.3 The Smith Predictor and Internal Model Control

A controller that knows the delay can stop waiting for it. The Smith predictor runs a model of the plant without its delay, $G_0(s) = K_p/(\tau s+1)$, beside a copy with the delay, and feeds back the measurement plus the difference between the two, as in Equation (1.4) [1].

$$y_p(t) = y(t) + \hat{x}(t) - \hat{x}(t-\theta_m), \qquad \tau\,\frac{\mathrm{d}\hat{x}}{\mathrm{d}t} = -\hat{x} + K_p\,u \tag{1.4}$$

where θ_m is the model's delay. When the model is exact, y and $\hat{x}(t-\theta_m)$ cancel except for the effect of disturbances, and the controller acts on the undelayed model output. Its characteristic equation then contains no delay. A PI controller with $\tau_I = \tau$ and $K_c = \tau/(K_p\tau_c)$ inside the predictor is exactly internal model control (IMC) with a first-order filter of time constant τ_c [9]. The setpoint response is the filter delayed by θ, and the load response follows Equation (1.5).

$$\frac{y(s)}{d(s)} = \frac{\mathrm{e}^{-\theta s}}{\tau s+1}\left(1 - \frac{\mathrm{e}^{-\theta s}}{\tau_c s+1}\right), \qquad \mathrm{IE} = K_p d\,(\theta + \tau_c) \tag{1.5}$$

The SIMC rule of Volume 2 is this design with the delay approximated, so a Smith predictor tuned with $\tau_c = \theta$ removes no more integrated error after a load than the SIMC PI. Its gain lies in the setpoint response and in the freedom to choose τ_c shorter than θ when the model can be trusted.

Table 1.1 and Figure 1.3 measure that freedom on the example. With $\tau_c = 2.5$ min the setpoint IAE falls from 10.93 to 7.46 and the load IAE from 10.25 to 7.50, and the dose moves four times further in the first instant. The load floor stays far off. The predictor cancels the tank's lag in its model, so an upset in the water decays with the open-loop time constant, and the load IE cannot fall below $K_p d\,\theta$ however short τ_c is. Studies of dead-time compensators find the same pattern, a clear gain over PI control when the delay dominates the lag and little gain on loads when it does not [14]. Reaching nearer the floor needs an estimate of the disturbance itself, the subject of Section 1.4.7, or a feedforward from Volume 3.

Table 1.1 Controllers on the Example at $\theta = \tau = 5$ min. IAE in g min/L After Unit Steps, the Product's Peak Rise After the Load, and the Longest True Delay at Which the Loop Stays Stable With the Model's Delay Held at 5 min

Controller	Setpoint IAE	Load IAE	Load peak (g/L)	Stable to (min)
PI, SIMC ($K_c = 0.5$, $\tau_I = 5$)	10.93	10.25	0.73	15.6
Smith predictor, $\tau_c = 5$ min	9.96	10.00	0.70	16.7
Smith predictor, $\tau_c = 2.5$ min	7.46	7.50	0.67	13.1
Smith predictor, $\tau_c = 1$ min	5.96	6.00	0.65	8.9
Least possible, Equation (1.2)	5.00	1.84	0.63	—

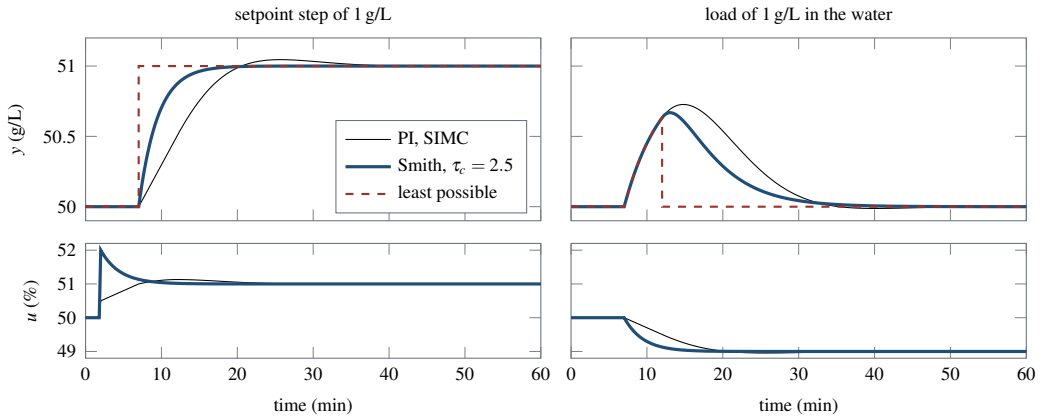


Figure 1.3 A Setpoint Step (Left) and a Load in the Water (Right) at 2 min, Design Flow. Top: Product y. Bottom: Dose Ratio u. Thin Solid: SIMC PI. Thick Solid: Smith Predictor With $\tau_c = 2.5$ min. Dashed: the Least Possible Response, Equation (1.2).

The price of the predictor is its dependence on the model's delay. The last column of Table 1.1 holds the model at 5 min and lengthens the true delay until the loop no longer settles. The SIMC PI tolerates 15.6 min, three times its design, and the predictor with $\tau_c = 1$ min only 8.9 min. A shorter τ_c buys speed with robustness to exactly the error a delay that varies with flow produces, as Section 1.4.4 shows.

Prediction through a delay appears in many forms in this volume. Chapter 5 turns its thickness loop into a Smith predictor with a forming model, and with the model off the same tuning hunts on the delayed gauge with a period of about 54 min. Chapter 2 tunes its loop by the SIMC rule, which it calls an IMC derivation, and its predictive controller carries the pellets in the auger as a shift register of the model's delay, which may be set away from the true one. Chapter 3 derives its PI settings from the IMC formula, and its predictive controller's model leaves out the analyzer lag, so its loop stops settling at a lag of 1.15 min. Chapter 9 projects the fund forward from the last valuation with the recorded contributions, a predictor across a delay of one year, and Chapter 4 predicts the kappa of every slug in the vessel from the heater temperature it received.

1.4.4 Delay That Varies With Flow

The pipe holds a fixed volume, not a fixed time. Material that enters it leaves once a pipe volume of flow has passed, as in Equation (1.6).

$$\int_{t-\theta(t)}^{t} Q(t')\,\mathrm{d}t' = V_p \tag{1.6}$$

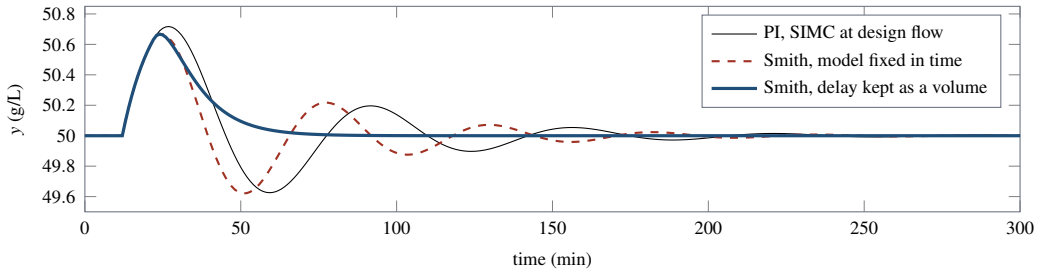


Figure 1.4 A Load of 1 g/L at 2 min With the Flow Halved, So That $\tau = \theta = 10$ min. Thin Solid: SIMC PI Tuned at Design Flow. Dashed: Smith Predictor With Its Model Fixed at 5 min. Thick Solid: Smith Predictor Whose Delay and Time Constant Follow the Measured Flow.

At constant flow this gives $\theta = V_p/Q$. When the flow halves, the delay and the tank's time constant both double to 10 min. A Smith predictor whose model keeps 5 min for both now carries a model error of the kind measured in Table 1.1.

Figure 1.4 repeats the load step at half flow. The SIMC PI stays stable, but its IAE triples from 10.25 to 30.9 g min/L, and the product swings below its setpoint before it settles. The predictor with its model fixed in time rings with a period of about 52 min and an IAE of 26.1. A predictor that measures the flow and keeps its delay as a volume, with its model's time constant and its integral time scheduled on V_t/Q, stays exact. Its response is the design response stretched to twice the time, with an IAE of 15.0, exactly twice the design value. Written in the volume of product that has passed, as Chapter 7 writes its diversion windows, the loop does not change at all.

Real lines mix as they carry. A chain of N equal well-mixed cells of total volume V passes a pulse as the Erlang distribution of Equation (1.7), with a mean residence time $\bar{t} = V/Q$ and a standard deviation of $\bar{t}/\sqrt{N}$ [15].

$$E(t) = \frac{(N/\bar{t})^N t^{N-1}}{(N-1)!}\, \mathrm{e}^{-Nt/\bar{t}} \tag{1.7}$$

One cell spreads a disturbance over a spread as large as its mean, and a hundred cells spread it over a tenth of the mean, close to plug flow. The distribution decides how long material affected by an upset keeps arriving, and so how much product a supervisor must hold back.

Several chapters follow material in this way. Chapter 7 writes its line as 225 cells, follows material by the volume that has passed the sample point, and diverts while the declared 0.1 and 99.9 % points of the coil's distribution, 0.7956 and 1.2328 coil volumes, overlap a window opened by a failing result. Chapter 6 runs a tracker with the plant's own tanks-in-series structure fed by measured rates, and its dryer turns a short feeder error into one or two packages of altered composition. Chapter 4 carries its chip column as slugs advanced on the measured flow, because a grid of 40 cells smeared the design kappa by 2.64 %, and a production rise of 15 % shortens the residence from 5.00 to 4.348 h. Chapter 11 times its feedforward to the window in which the affected material crosses each zone, since the material needs 1710 min to reach the last one. Chapter 5 tags each parcel of ribbon with the speed and thickness of its birth, so that the gauge, 15.2 min downstream, compares like with like.

1.4.5 Sampled and Multirate Measurements

An analyzer adds its own delay to the pipe's. A result drawn at t_s and reported at $t_s + t_d$ is held until the next one, so its age runs from t_d to $t_d + T_s$ and averages Equation (1.8).

$$\bar{t}_{\text{age}} = t_d + \frac{T_s}{2}, \qquad \theta_e = \theta + t_d + \frac{T_s}{2} \tag{1.8}$$

The effective dead time θ_e is what a tuning rule should be given. For AT-1, with $T_s = t_d = 2$ min, the mean age is 3 min and the loop sees $\theta_e = 8$ min. The SIMC loop tuned for the pipe alone then leaves

a load IAE of 16.1 instead of 10.25, against an IE that is still 10, so the error swings to the other side. The rule applied to 8 min gives $K_c = 0.3125$ % per g/L, an IE of exactly 16 and an IAE of 17.0, slower but with far less swing. Each minute of a result's mean age costs about as much as a minute of pipe.

The worst-case age also sizes the equipment downstream. A change that begins just after a sample is drawn is caught by the next sample and reported one cycle after that, so it is detected after $2T_s$ for an analyzer that reports one cycle late, plus any transport to the instrument. Chapter 7 writes this as $t_{det} = 2t_{run} + t_{tr}$ and requires its holding coil to cover it, so that no material passes the diverter before the analyzer has judged it.

The laboratory is too slow to read between results, so a loop on it acts once per result. Let the controller move the dose by g/K_p times each result's error. A result published now describes material made 35 min ago, before the previous move. The move before that is fully in it, since the tank settles to 99.3 % within the 25 min it has had. The error therefore obeys Equation (1.9), with the result two moves behind.

$$e_k = e_{k-1} - g\,e_{k-2} \quad \Longrightarrow \quad z^2 - z + g = 0 \tag{1.9}$$

The loop is stable for $g < 1$ and free of oscillation for $g \le 1/4$, where the two roots meet at $z = 1/2$. The simulated limit is $g = 1.001$, which the 99.3 % settling explains. With $g = 1/4$ the results after a load of 1 g/L stay within 0.05 g/L only from the one published 5 h after the load. With $g = 0.5$ the product overshoots by a quarter of the load and the results settle after 6 h, but the IAE is lower, 99.7 against 119 g min/L. Either way the cost is about ten times that of the pipe alone.

> **✦ DID YOU KNOW? A Controller That Dozes on Purpose**
>
> One answer to a long dead time is the sample-and-hold PID. It acts on the error for a short moment, then holds its output while the correction travels through the process, and only then looks again. The holds slow it, but they allow bolder tuning [4].

Equation (1.9) recurs wherever a correction acts on information two decisions old. Chapter 9 decides once a year on a valuation a year old, and its deficit obeys the same equation with the fund's growth factor beside the first term, so its roots turn complex for spread periods under 3.85 years. Chapter 10 offers moisture control on the laboratory alone, with samples every 30 min reported 40 min later, and with the shipped tuning it oscillates with a growing amplitude. Chapter 7 runs its quality trim once per analyzer result, and Chapter 6 closes its moisture loop once per dryer cell, every 90 s, on a reading that summarizes 480 s of filling and drying. Chapter 4 has three clocks at once, a titrator every 10 min reported 10 min later, an analyzer every 15 min reported after 20, and a laboratory every 4 h reported an hour later. Combining measurements that arrive at different rates and ages is multirate estimation [11], the subject of the next three subsections.

1.4.6 Observability

An estimator can recover a state only if the measurements carry information about it. A linear model $x_{k+1} = Ax_k$, $y_k = Cx_k$ is observable when no initial state other than zero produces an output that stays at zero, which holds when the matrix of Equation (1.10) has full rank n, the number of states.

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}, \qquad \operatorname{rank} \mathcal{O} = n \tag{1.10}$$

A delay never makes a state unobservable. The pipe is a chain of states whose end is measured, and everything in it reaches the instrument in time. A delay makes information late, while a missing measurement can make it absent.

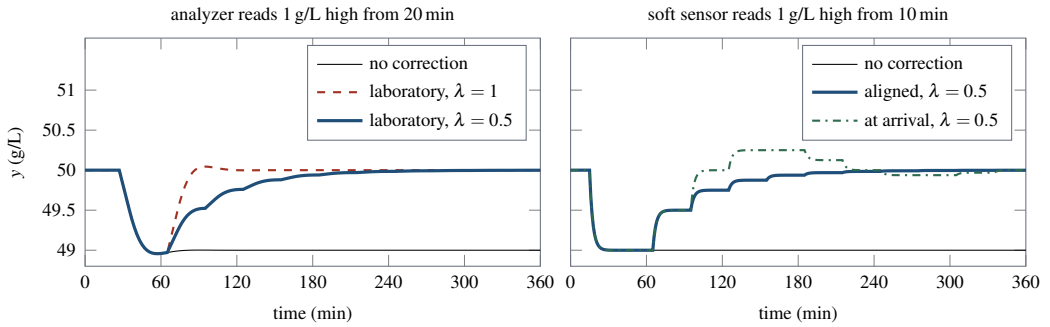


Figure 1.5 Product y When a Measurement Turns Wrong by 1 g/L. Left: Loop on the Analyzer, Whose Bias Is Learned From the Laboratory (Thin Solid: Not Corrected. Dashed: $\lambda = 1$. Thick Solid: $\lambda = 0.5$). Right: Loop on the Soft Sensor, Whose Bias Is Learned From the Laboratory Against the Soft Value of the Sampled Material (Thick Solid) or the Value When the Result Arrives (Dash-Dotted).

The example shows the difference. Sampled once a minute, the tank and five cells of pipe form six states. Add the water's solute d and a bias b of the analyzer, both constant unless they jump. With the analyzer alone, which reads $y + b$, the eight states give a rank of seven. The direction it cannot see is a tank settled at $x = K_p d$ together with $b = -K_p d$, a feed upset that has fully arrived hidden by an equal and opposite drift of the analyzer. Two constant states seen through one measurement are never both observable [16], so two disturbances that persist cannot both be estimated from one measurement. Adding the laboratory, which reads the same product without a bias, restores rank eight. The soft sensor has the same flaw, since its error from the water's salts is a second constant beside d, and the laboratory restores it too.

Figure 1.5 (left) shows what is at stake. At 20 min the analyzer begins to read 1 g/L high, and the loop on the analyzer, retuned for 8 min, drives the product down to hold the analyzer on 50 g/L. The product settles at 49 and stays there, and nothing on the analyzer trend reveals it. A correction from the laboratory pairs each result with the analyzer's reading of the same sample, so that the difference is the bias alone, $\hat{b} \leftarrow \hat{b} + \lambda (y_A - y_L - \hat{b})$. With $\lambda = 1$ the first result that contains the drift, published at 60 min, removes it, and the product is back within 0.1 g/L at 82 min. With $\lambda = 0.5$ the bias is learned by halves and the product returns at 161 min.

> ✦ DID YOU KNOW? **A Plugged Sample Line Reads Calm**
>
> Many analyzers stand away from the process and read a sample carried to them through tubes, filters and regulators. If the sample line plugs, the analyzer measures stale fluid and reports it as the process, with no way of knowing [4].

Observability decides several results in this volume. Chapter 13 finds the compass bias observable from position fixes taken on lines flown in opposite directions, and its estimate of a 1° bias reaches 1.005° by 300 s. Across a canyon, without a fix and without a measurement of the current, the drift is unobservable, and no tuning of the guidance recovers the unimaged strip. Chapter 4 estimates its analyzer's drift against the laboratory exactly as the example does, pairing reports by the instant their samples left the blowline. Chapter 11 corrects only the surface node of its brick model from the exit pyrometer, so the core error is almost untouched. Chapter 12 corrects its stored ammonia along the sensor's sensitivity to each cell's store, and near the bottom of the sensor's V the sensitivities cancel and the observer runs on its model alone. Chapter 10 adds its laboratory bias to the moisture only, so the sticky margin that its overrides read stays 0.9 K optimistic after a change of concentrate, and the laboratory never sees it.

Table 1.2 The Weight of a Bias Update as a Steady Kalman Gain: the Ratio of Drift to Noise It Assumes, the Results Needed to Learn 90 % of a Constant Error, and the Standard Deviation of the Estimate Relative to That of One Result

λ	q/r_v	Results to 90 %	$\sigma_{\hat{b}}/\sqrt{r_v}$
0.1	0.011	22	0.23
0.2	0.050	11	0.33
0.5	0.50	4	0.58
0.8	3.2	2	0.82
1.0	∞	1	1.00

1.4.7 Kalman Filtering and Moving Horizon Estimation

A Kalman filter is the estimator that weighs a model against noisy measurements in proportion to their uncertainties [7]. For a model $x_{k+1} = Ax_k + Bu_k + w_k$ and measurements $y_k = Cx_k + v_k$, with process noise of covariance W and measurement noise of covariance V, each period predicts and then corrects as in Equation (1.11).

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1}, & P_k^- &= AP_{k-1}A^\top + W \\ L_k &= P_k^- C^\top \left(CP_k^- C^\top + V\right)^{-1}, & \hat{x}_k &= \hat{x}_k^- + L_k\left(y_k - C\hat{x}_k^-\right) \end{aligned} \tag{1.11}$$

with $P_k = (I - L_k C)P_k^-$. When the noise statistics are constant, L_k converges to a steady gain, and most apps of this volume use that steady form. An unmeasured disturbance enters as an extra state that the model lets drift as a random walk, and the filter's estimate of it is what makes a predictive controller free of offset [17].

The simplest such filter is the bias update of the previous subsection. A bias that drifts as a random walk with variance q per result, read through noise of variance r_v, has the steady Kalman gain λ of Equation (1.12).

$$\hat{b}_k = \hat{b}_{k-1} + \lambda\left(y_k - \hat{y}_k - \hat{b}_{k-1}\right), \qquad \frac{q}{r_v} = \frac{\lambda^2}{1-\lambda}, \qquad \frac{\sigma_{\hat{b}}^2}{r_v} = \frac{\lambda}{2-\lambda} \tag{1.12}$$

The weight that chapters choose by hand therefore states a belief about how fast the bias moves against how noisy each result is. A constant error ε is learned as $\varepsilon[1-(1-\lambda)^n]$ after n results, and the noise passed into the estimate falls as λ falls. Table 1.2 gives both sides of the trade, and a run of the update on the example's laboratory with noise of 0.1 g/L gives standard deviations of 0.100, 0.055 and 0.030 g/L at λ of 1, 0.5 and 0.2, against 0.100, 0.058 and 0.033 from the equation.

A late measurement must update the state it describes. Three treatments are in use [18]. The model can carry the delay as extra states, a shift register of past values, so that the late measurement is an ordinary output of the augmented model. The filter can store its recent states and inputs, update at the instant the measurement describes and predict forward again, which is exact and costs a store of history. Or it can apply the measurement at its arrival as if it were new, which is cheap and wrong by whatever happened in between.

A moving horizon estimator (MHE) poses the same task as a least-squares fit over a window of the last N_e samples, as in Equation (1.13) [19].

$$\begin{aligned} \min_{x_{k-N_e},\, w} \; & \|x_{k-N_e} - \bar{x}\|^2_{\bar{P}^{-1}} + \sum_j w_j^\top W^{-1} w_j + \sum_j (y_j - Cx_j)^\top V^{-1}(y_j - Cx_j) \\ \text{s.t. } & x_{j+1} = Ax_j + Bu_j + w_j, \qquad x_{\min} \le x_j \le x_{\max} \end{aligned} \tag{1.13}$$

The first term, the arrival cost, summarizes the data before the window through a prior $\bar{x}$ and its covariance $\bar{P}$. With the filter's own prediction and covariance as the arrival cost and no bound active,

the window's solution is the Kalman filter's estimate. The MHE differs only when a bound binds, such as a concentration that cannot be negative, and it pays for that with an optimization at every sample [20]. A clamp applied after a recursive update, as several chapters apply to their biases, keeps the estimate in range but does not let the bound inform the rest of the state.

Chapter 2 runs an extended Kalman filter, Equation (1.11) linearized at every step, with a heat bias as a random walk, and trusts its probe to $r_v = 1\,\mathrm{K}^2$, above its noise, because a tighter filter passes the noise into the feed. Chapter 3 runs a steady filter with a disturbance state and offers an MHE whose reflux records differ from the filter's by at most 2.6×10^{-8}, because no bound ever became active. Chapter 5 runs a steady filter on its annealing zones. Chapter 13 stores 24 s of history, applies each acoustic fix at the instant it describes behind a chi-square gate, and finds that applying the fixes at their arrival more than triples the error.

1.4.8 Soft Sensors and the Laboratory Bias Update

A soft sensor infers a variable that is slow to measure from variables that are fast [8, 21]. Placed upstream of a delay, it takes the delay out of the loop. AY-1 reads the tank outlet, so the loop on it controls a first-order plant with no dead time, and the PI controller of the predictor's inner loop, $K_c = 2$ and $\tau_I = 5\,\mathrm{min}$, applies directly. A load in the water then costs an IAE of 2.50 g min/L, a quarter of the SIMC loop on the product, because the soft sensor sees the upset five minutes before the pipe delivers it. A soft sensor is a Smith predictor whose model output has been replaced by a measurement.

Its weakness is that it is only a model. When the water's salts change at 10 min, AY-1 reads 1 g/L high, the loop drives the product to 49 g/L, and without a correction it stays there, as the thin line of Figure 1.5 (right) shows. The laboratory corrects it with a bias update of the form of Equation (1.12), but only if each result is compared with the soft sensor's value for the same material, as in Equation (1.14).

$$\hat{b} \leftarrow \hat{b} + \lambda \left[y_L(t_s) - \hat{y}_{\mathrm{soft}}(t_s - \theta) - \hat{b} \right] \tag{1.14}$$

The bottle drawn at t_s holds product that left the tank at $t_s - \theta$, and its result arrives at $t_s + t_d$. Stored soft values make the alignment possible. Aligned, the bias error falls as $(1-\lambda)^n$, so with $\lambda = 1$ one result removes it and the product is back within 0.1 g/L at 71 min, and with $\lambda = 0.5$ at 157 min.

Compared instead with the soft value at the result's arrival, the update reads the effect of its own last correction as new error. The result was drawn before that correction reached the product, while the soft value already includes it. The bias error then obeys $\varepsilon_k = \varepsilon_{k-1} - \lambda\,\varepsilon_{k-2}$, the same equation as the laboratory loop of Section 1.4.5. With $\lambda = 0.5$ the product overshoots to 50.25 g/L and returns only at 216 min. With $\lambda = 1$ the roots lie on the unit circle, and the product swings between 49 and 51 g/L with a period of six results, three hours, indefinitely. Aligning a result with the instant it describes turns a loop with a delay into a filter without one.

> ✦ DID YOU KNOW? **Comparing at Arrival Undoes the Update**
>
> From the simulator: with the soft sensor reading 1 g/L high, the bias update with $\lambda = 1$ leaves an IAE of 50.0 when each result is compared with the material it sampled, and 313 when it is compared at its arrival, against 463 with no correction (Table 1.3).

Chapter 10 infers its powder moisture from the air temperatures with the plant's own drying model and corrects it with $\lambda = 0.5$, each result compared with the estimate stored when the sample was drawn, and not with the estimate of the moment. It also reports the trade of Table 1.2, with less powder off specification at $\lambda = 1$ than at 0.2. Chapter 7 runs a copy of its line as a soft sensor with a bias weight of 0.5 against predictions stored at each sample, and notes that a bias corrects a slow mismatch but not a short slug of unmeasured material. Chapter 4 uses weights of 0.8 for its analyzer drift and 0.1 for its predictor, with a gate that holds back an outlying report until later reports agree. Chapter 5 infers thickness at a camera 2.2 min after forming, against 15.2 min for the gauge, and cuts

Table 1.3 What Each Measurement Costs on the Example. Product IAE in g min/L After a Load of 1 g/L in the Water, and After a Measurement Error of 1 g/L Over a Run of 6 or 8 h

Measurement and controller	Delay seen by the loop	IAE
Load of 1 g/L in the water, design flow unless stated		
Least possible through the pipe	pipe, 5 min	1.84
Soft sensor at the tank, exact, PI with $K_c = 2$	none	2.50
Instrument at the sample point, Smith, $\tau_c = 2.5$	pipe, 5 min	7.50
Instrument at the sample point, SIMC PI	pipe, 5 min	10.25
Same at half flow: Smith in volume, Smith in time, PI	pipe, 10 min	15.0, 26.1, 30.9
Analyzer, PI tuned for the pipe, for 8 min	pipe and mean age, 8 min	16.1, 17.0
Laboratory alone, $g = 0.5$ per result	pipe and mean age, 50 min	99.7
Measurement reads 1 g/L high, laboratory bias update		
Analyzer from 20 min: none, $\lambda = 0.5$, $\lambda = 1$	over 6 h	325, 68.0, 39.5
Soft sensor from 10 min: none, $\lambda = 0.5$, $\lambda = 1$	over 8 h	463, 80.0, 50.0
Soft sensor, compared at arrival: $\lambda = 0.5$, $\lambda = 1$	over 8 h	89.7, 313

Table 1.4 The Delay or Missing Measurement of Each Chapter and What Recovers It

Ch.	Delay or missing measurement	What recovers it
2	Pellets ride an auger and burn before they heat, and the lid opens	Lid-aware hold of the feed, and a predictive controller whose filter carries the fuel in transit
3	Analyzer lag of up to 2 min on the distillate	Detuning by phase margin, and a predictive controller with a disturbance state
4	Four hours from heater to blowline, and three instruments on three clocks, one drifting	H-factor feedforward, a predictor of every slug reconciled with the analyzer and the laboratory
5	Thickness read 15.2 min after forming, and stress settled before the polarimeter	Smith predictor, inferred thickness at the camera, thermal-history estimator with a bias
6	Composition seen minutes after the feeders, and a feeder blind while it refills	Refill feedforward, a cascade sampled once per dryer cell, a material tracker
7	Delay and spread that change with throughput, and an analyzer that reports late	Diversion by volume and residence-time quantiles, a soft sensor with a laboratory bias
8	A channel with a round trip of up to 2.6 s, jitter and loss	Reduced reflection, wave variables, passivity control, model mediation
9	A valuation one year late, decided once a year	Spread period, a projected estimate, a predictive plan with a buffer
10	Moisture from the laboratory after 40 min, and the sticky margin never measured	Psychrometric soft sensor with a laboratory bias, feedforward, overrides
11	The brick's core never measured, and a day of transport	Two-node soft sensor and core cascade, feedforward timed per zone
12	Stored ammonia never measured, and a sensor that reads ammonia as nitrogen oxides	Model-based observer, storage setpoint scheduled on temperature
13	Position never measured under water, and acoustic fixes late and sometimes wild	Extended Kalman filter with fixes applied at their instant behind a gate

the rejected glass of a forming upset from 734 to 212 m^2. Chapter 11 infers the unmeasured core of each brick from a two-node model and holds that estimate with a cascade, and Chapter 6 trims its weight and hardness calibrations from each at-line test.

1.4.9 What Each Delay and Missing Measurement Costs

The section reduces to two questions that can be asked of any plant. What does each delay or missing measurement cost, and what recovers it? Table 1.3 answers both for the example, and Table 1.4 names the delay, the missing measurement and the remedy of every chapter. The final chapter returns to the second table with the measured result of each remedy.

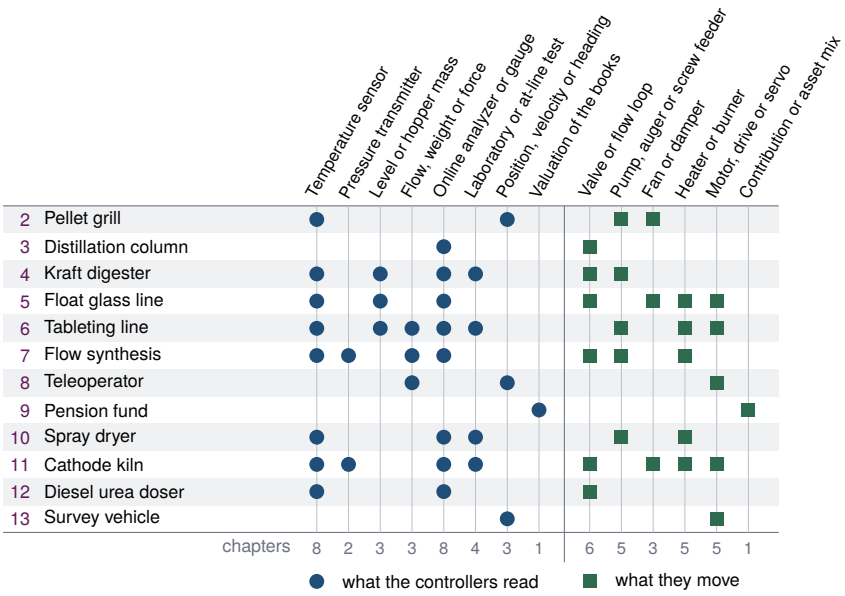


Figure 1.6 What the controllers of each chapter read and move: a round mark where a controller reads the instrument and a square mark where it moves the element, as the chapter's variables table and its technology section name them. A flow set through a flow station counts as a valve or flow loop, and a computed value is marked under the instruments it is computed from. Notice that the analyzers and the laboratory, the slow readings, appear in as many chapters as the thermometers.

Three patterns run through the table. Where the delay is a known transport, in Chapters 2 to 7, the remedy is to predict or to track material, and its quality depends on knowing the delay as the flow changes. Where the measurement is late but exists, a model or a faster instrument takes its place between results, and the slow measurement corrects its bias. Where a variable is never measured, in Chapter 11, Chapter 12 and Chapter 13, the estimate is only as good as the model, and observability decides which errors the remaining sensors can catch. Chapter 8 stands apart, because no model of the channel removes its delay and the remedy is to stay passive. Volume 9 develops the predictive controllers that combine these elements in one optimization.

1.5 Instruments and Final Elements in This Volume

A delay often sits in the instrument rather than in the plant. Figure 1.6 lists what the controllers of each chapter read and what they move. A temperature and an online analyzer or gauge lead the readings, with eight chapters each, and four chapters also wait for a laboratory or at-line test. The three with a laboratory in the loop, Chapter 4, Chapter 10 and Chapter 11, wait from forty minutes to six hours for a result. The final elements are spread more evenly: a valve or flow loop in six chapters, and a pump or feeder, a heater and a motor or drive in five each. In Chapter 9 the instrument is a valuation of the books and the final element a contribution.

An analyzer is slow by its principle. A chromatograph separates the species of a sample in time, so cycle times of minutes are not uncommon even in an online instrument, and some dead time is inevitable [4]. Many analyzers read a sample carried to them by a sample system, and problems with process analyzers often stem from those systems [4]. Such an analyzer stands behind the lag of Chapter 3 and, for liquids, the at-line result of Chapter 7. The top of Figure 1.7 shows a near-infrared analyzer, the kind that reads the blend of Chapter 6 in its feed frame at once, but only through a calibration. The bottom shows a pyrometer, which reads the surface of the bricks leaving

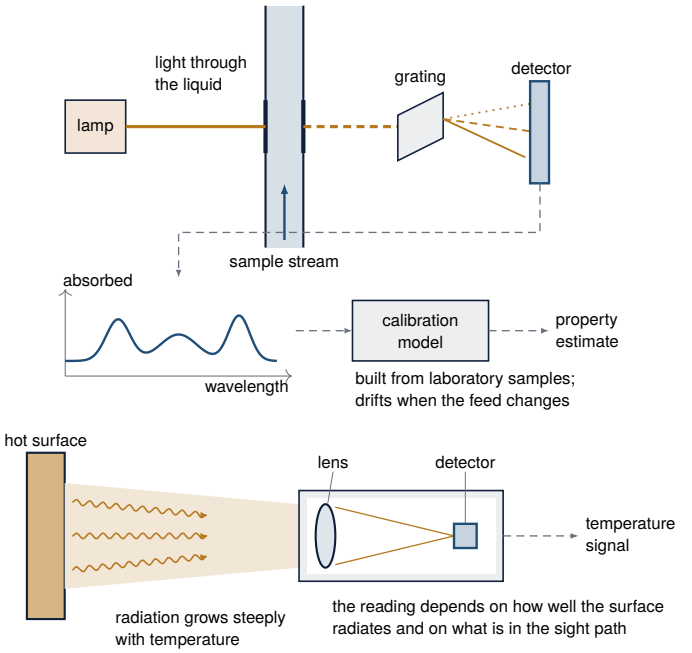


Figure 1.7 Top: principle of a near-infrared analyzer on a sample stream. Bottom: principle of an infrared pyrometer. Notice that the analyzer turns a spectrum into a property through a calibration model, a soft sensor built into the instrument, and that the pyrometer sees only the surface it is aimed at.

Chapter 11 and nothing of their core. Even a transmitter adds dead time when it reports sparingly, and battery-powered wireless transmitters may update only once in several minutes [4].

The estimator usually runs one layer above the loops it serves. A distributed control system places a supervisory level over localized controllers, with cycle times from milliseconds to minutes [22], and a survey of predictive control draws basic control executing about every second, constraint control about every minute and a local optimizer about every hour [12]. Figure 1.8 places the loops of this volume on their platforms. The apps run from the millisecond of Chapter 8 to the yearly decision of Chapter 9.

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 1.5 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

Layer	What it does in this volume	Where it runs	How often
Optimizer or decision calendar	an economic optimum for the cook or the kiln, a yearly contribution, a predictive plan	**Local optimizer** over the loops; a **plan year** with its valuation date	30 min to 2 h in the plants; once a year in the fund
Estimator, predictor and supervisor	a Kalman filter, a soft sensor with its bias, a Smith predictor, a material tracker	**Supervisory level** of a DCS; an **application node** beside it; the **vehicle's computer**	each loop scan, up to 1 min; a predictive move every 1 s to 1 min
Base loops	velocity-form PI loops on the estimates, a force loop on each side of a channel	**Localized controllers** of a DCS; a **microcontroller**; a vehicle's **control unit**	1 ms on the teleoperator; 0.1 s to 5 s in most apps; 1.5 min in the kiln
Sensors and final elements	thermometers, analyzers, a laboratory, a valuation; valves, pumps, heaters, drives	wires, a sample line, a bottle to the laboratory, an actuary's report	results from minutes to 6 h late; a valuation a year late

Figure 1.8 The control layers of this volume, the platform that hosts each and how often each acts. Platforms are from [22] and [12] and from the chapters' technology sections, and the periods from the chapters. Notice that the estimator sits between the loops and the optimizer, and that the slowest measurement is often slower than every loop it serves.

Table 1.5 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.

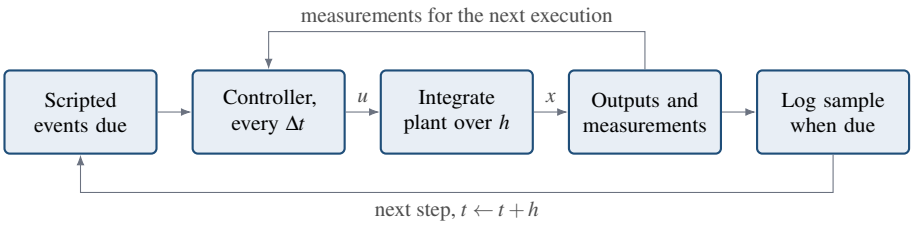


Figure 1.9 Work of the Engine in One Integration Step

executions. The log takes one sample at a fixed interval of simulated time for the trends, the scorecard and the exported file. The order of work in one step is shown in Figure 1.9. 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 (1.15) to (1.18) for execution k and control period Δt.

$$e_k = a\,(\mathrm{SP}_k - \mathrm{PV}_k), \qquad P_k = K_c\,e_k, \qquad D_k = -a\,K_c\,\tau_D\,\frac{\mathrm{PV}^f_k - \mathrm{PV}^f_{k-1}}{\Delta t} \tag{1.15}$$

$$S_k = S_{k-1} + (P_k - P_{k-1}) + (D_k - D_{k-1}) \tag{1.16}$$

$$v_k = u_b + S_k + I_{k-1}, \qquad u_k = \mathrm{sat}(v_k) \tag{1.17}$$

$$I_k = I_{k-1} + \left[\frac{K_c}{\tau_I}\,e_k + \frac{u_k - v_k}{T_t}\right]\Delta t \tag{1.18}$$

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 (1.18) drives I toward the value that reproduces the output applied, which is back-calculation [23]. With no integral action, Equation (1.18) is skipped. Clamping the running sum would instead discard each increment that crosses a limit and keep each one that returns, so symmetric noise would push the output off the limit even in a proportional-only loop.

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

In Manual the controller keeps executing and sets $I_k = u_{\mathrm{man}} - u_b - S_k$, so the output it would send equals the manual value u_{man}. The first automatic output differs from it by one period's change in P and D. In Auto the engine copies each controller output to the manual station, so a switch to Manual is also bumpless. The standalone studios with PID loops use velocity-form code of their own, with the limit on the output and anti-windup on the integral alone, in which a gain change applies only to later increments.

1.6.3 The Predictive Control and Estimator Options

More than 100 apps offer model predictive control (MPC) as an option beside the structure they open with, which stays the default. Volume 9 develops the theory. In most of them the controller plans with

a small linear model in discrete time as shown in Equation (1.19).

$$x_{k+1} = A\,x_k + B\,u_k + E\,w_k + c, \qquad y_k = C\,x_k \tag{1.19}$$

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 (1.20) over a horizon of N steps.

$$J = \sum_{k=1}^{N} (y_k - r_k)^{\top} Q\,(y_k - r_k) + \sum_{j} \Delta u_j^{\top} R\,\Delta u_j + \sum_{i} \rho_i\, s_i^2 \tag{1.20}$$

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 [24]. 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 [17]. 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 [19].

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

1.6.4 Scenarios, Progress and Data

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

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

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

Show Model opens the equations, live values of the variables, the editable parameters and the assumptions. The trends plot SP and PV above the MV for each loop, with actuator limits and marks at setpoint moves, disturbances, mode changes and scripted events. The scorecard reports for each loop the integral of the absolute error (IAE), overshoot, settling time, MV travel and share of time at

a limit, with the app's own measures, and it does not combine them into one score. IAE and travel accumulate from Reset, so two runs are compared at the same clock reading.

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

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

1.7 Working With AI Agents While Studying

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

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

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

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

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

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 14.

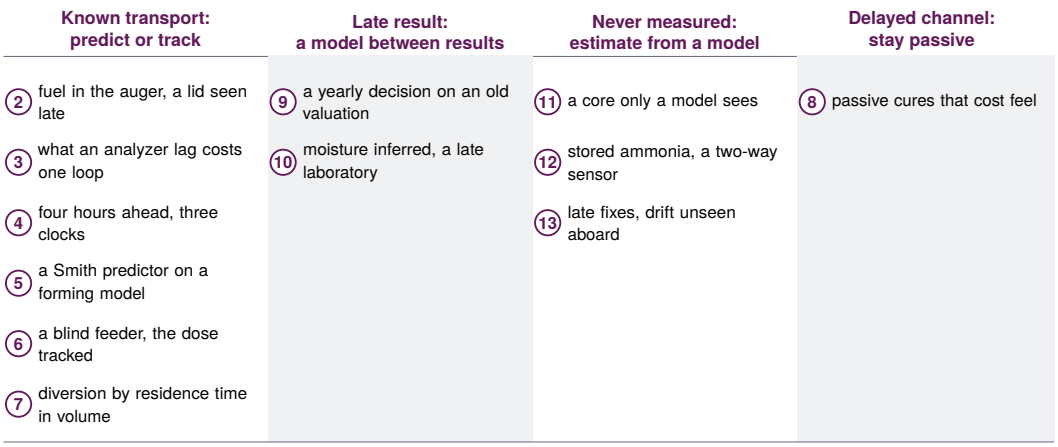


Figure 1.10 Method map of the volume. Each chapter is placed by the three patterns of Section 1.4.9, with the teleoperator apart, and labeled with its lesson, shortened from Table 1.6. The placement is the editor's reading of the two. Notice that most chapters predict through a known transport, and that the three whose variable is never measured rely on an estimator.

Table 1.6 Roadmap of the Volume

Ch.	Plant	The lesson	Level	Time
2	Wood pellet grill	Identification through a fuel delay, a lid the loop answers too late, and prediction of fuel in transit.	intermediate	3.5 h
3	Thirty-two stage column	What an analyzer lag costs one loop, and a predictive controller whose model omits it.	introductory	2.5 h
4	Continuous kraft digester	Feedforward through four hours, and three clocks reconciled into one estimate.	advanced	2 h
5	Float glass line	A Smith predictor on a forming model, and a stress forecast before it can be measured.	advanced	2.5 h
6	Continuous tableting line	A blind feeder, a cascade sampled per cell, and tracking of the dose.	intermediate	2 h
7	Continuous flow synthesis	Diversion by residence time measured in volume, and a soft sensor with a bias.	intermediate	2.5 h
8	Bilateral teleoperator	The gain limit of force through a delay, and passive cures that cost feel.	advanced	2.5 h
9	Defined-benefit pension fund	A decision once a year on a valuation a year old, with a small valve.	introductory	2.5 h
10	Milk powder spray dryer	A soft sensor of moisture corrected by a late laboratory, and an inferred constraint.	intermediate	2.5 h
11	Cathode calcination kiln	A core temperature only a model sees, held by a cascade on the estimate.	intermediate	3 h
12	Diesel urea dosing	An observer of stored ammonia behind a sensor that reads both ways.	advanced	2.5 h
13	Underwater survey vehicle	A Kalman filter with late fixes, and drift that nothing aboard can see.	advanced	2 h

1.8 Chapter Roadmap

Table 1.6 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 1.10 places the same chapters by the kind of information problem each centers on.

Chapter 3 and Chapter 2 are the first to read. The first isolates one loop and the cost of a measurement lag, and the second adds a true transport delay and a disturbance the loop sees too late. Chapter 5 then shows the Smith predictor on a real line. Chapters 4, 6 and 7 follow material along process lines whose delays move with throughput, and can be read in any order once Sections 1.4.4 and 1.4.5 are familiar. Chapter 10 and Chapter 11 form a path on soft sensors, and Chapter 12 and Chapter 13 a path on observers and Kalman filters, best read after Sections 1.4.6 to 1.4.8. Chapter 8 stands on its own for readers from robotics, and Chapter 9 for readers from finance, where the valuation is the late measurement.

KEY POINTS

- A controller sees the plant late, and a dead time sets a floor under the error that no controller can pass.
- A Smith predictor feeds back what its own moves will do, and it is only as good as its delay, which in a pipe is a volume, not a time.
- A sampled result adds its mean age to the dead time, and a loop that acts once per result two moves old is free of oscillation only for a gain of a quarter or less.
- A bias update is a steady Kalman filter, and each result must be compared with the estimate of the material it describes.

Where else this applies. Prediction across two dead times on a steam heat exchanger in Volume 2, Chapter 4, analyzer biases in a crude tower's predictive controller in Volume 11, Chapter 4, and predictive controllers that join prediction and estimation in Volume 9.

References

[1] J. E. Normey-Rico and E. F. Camacho, "Dead-time compensators: A survey," *Control Engineering Practice*, vol. 16, no. 4, pp. 407–428, 2008.

[2] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*. John Wiley & Sons, 2006.

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

[4] 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.

[5] R. C. Goertz, "Master-slave manipulator," Argonne National Laboratory, Argonne, IL, Report ANL-4311, Mar. 1949.

[6] J. E. Normey-Rico, T. L. M. Santos, R. C. C. Flesch, and B. C. Torrico, "Control of dead-time process: From the Smith predictor to general multi-input multi-output dead-time compensators," *Frontiers in Control Engineering*, vol. 3, p. 953768, 2022.

[7] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[8] B. Joseph and C. B. Brosilow, "Inferential control of processes: Part I. steady state analysis and design," *AIChE Journal*, vol. 24, no. 3, pp. 485–492, 1978.

[9] C. E. Garcia and M. Morari, "Internal model control. a unifying review and some new results," *Industrial & Engineering Chemistry Process Design and Development*, vol. 21, no. 2, pp. 308–323, 1982.

[10] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Transactions on Automatic Control*, vol. 34, no. 5, pp. 494–501, 1989.

[11] R. D. Gudi, S. L. Shah, and M. R. Gray, "Adaptive multirate state and parameter estimation strategies with application to a bioreactor," *AIChE Journal*, vol. 41, no. 11, pp. 2451–2464, 1995.

[12] 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.

[13] U.S. Congress, "Pension Protection Act of 2006," Pub. L. 109-280, 120 Stat. 780, Aug. 17, 2006, GPO text PLAW-109publ280, 2006.

[14] A. Ingimundarson and T. Hägglund, "Performance comparison between PID and dead-time compensating controllers," *Journal of Process Control*, vol. 12, no. 8, pp. 887–895, 2002.

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

[16] K. R. Muske and T. A. Badgwell, "Disturbance modeling for offset-free linear model predictive control," *Journal of Process Control*, vol. 12, no. 5, pp. 617–632, 2002.

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

[18] T. D. Larsen, N. A. Andersen, O. Ravn, and N. K. Poulsen, "Incorporation of time delayed measurements in a discrete-time Kalman filter," in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 4, 1998, pp. 3972–3977.

[19] 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.

[20] E. L. Haseltine and J. B. Rawlings, "Critical evaluation of extended Kalman filtering and moving-horizon estimation," *Industrial & Engineering Chemistry Research*, vol. 44, no. 8, pp. 2451–2460, 2005.

[21] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven soft sensors in the process industry," *Computers & Chemical Engineering*, vol. 33, no. 4, pp. 795–814, 2009.

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

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

[24] 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.

1.9 Exercises

1.1 Theory and calculation · 5 min

A mixing tank with $\tau = 8\,\text{min}$ and $K_p = 1\,\text{g/L}$ per percent feeds a pipe that delays its product by $\theta = 4\,\text{min}$. The water's solute rises by $d = 2\,\text{g/L}$.

(a) Find the least IAE of the product that any controller can achieve after this load. *Checkpoint:* 1.70 g min/L.

(b) Find the SIMC PI settings with $\tau_c = \theta$ and the integrated error they leave after the same load. *Checkpoint:* $K_c = 1\,\%$ per g/L, $\tau_I = 8\,\text{min}$, $\text{IE} = 16\,\text{g min/L}$.

1.2 Concept check · 2 min

A Smith predictor on a first-order plus dead-time plant uses a PI with $\tau_I = \tau$ and the IMC filter time constant $\tau_c = \theta$. Compared with the SIMC PI tuned with $\tau_c = \theta$ on the same plant, what does it do after a step load at the plant's input?

(a) It removes the load in half the integrated error
(b) It leaves the same integrated error
(c) It leaves no error once the delay has passed
(d) It leaves twice the integrated error

1.3 Theory and calculation · 4 min

A tank of $6\,\text{m}^3$ discharges through a pipe of $3\,\text{m}^3$ to a sample point. The flow falls from 1.5 to $0.75\,\text{m}^3/\text{min}$.

(a) Find the tank's time constant and the pipe's delay at both flows. *Checkpoint:* 4 and 2 min at 1.5, 8 and 4 min at 0.75.

(b) A bottle is drawn at the sample point at 10:00 at the low flow. At what time did its material leave the tank, and which soft-sensor value should a bias update compare with its result? *Checkpoint:* 9:56, the value logged at 9:56.

1.4 Theory and calculation · 3 min

An analyzer draws a sample every 6 min and reports it 6 min after the draw. The process behind it has $K_p = 0.5$, $\tau_p = 12$ min and $\theta_p = 3$ min. Find the mean age of the held reading, the effective dead time the loop sees, and the SIMC PI settings with $\tau_c = \theta_e$.

1.5 Theory and calculation · 6 min

A controller acts once per laboratory result and moves its output by g/K_p times the result's error. The plant settles between results.

(a) Each result describes material made before the previous move but after the one before it. Write the error recursion, and find the range of g for stability and the largest g without oscillation. *Checkpoint:* $e_k = e_{k-1} - g\,e_{k-2}$, $0 < g < 1$, $g \leq 1/4$.

(b) A faster laboratory reports each result before the next move, so it describes material made after the previous move. Find the same two limits and the gain that removes a load in one result. *Checkpoint:* $0 < g < 2$, $g \leq 1$, $g = 1$.

1.6 Concept check · 2 min

An analyzer at the end of a pipe reads the product plus a slowly drifting bias, and the water entering the tank carries an unmeasured solute that also drifts. Which addition lets an estimator tell the two apart once both have settled?

(a) An analyzer with a shorter cycle at the same point
(b) A Kalman filter with a longer horizon on the same analyzer
(c) An unbiased laboratory result on the same product
(d) A second identical analyzer beside the first

1.7 Theory and calculation · 4 min

A laboratory bias update uses the weight $\lambda = 0.2$. Treat it as a steady Kalman filter for a bias that drifts as a random walk.

(a) Find the ratio of the drift variance to the result's noise variance that this weight assumes. *Checkpoint:* 0.05.

(b) Find how many results it needs to learn 90 % of a constant error, and the standard deviation of the estimate in units of one result's. *Checkpoint:* 11 results, one third.

1.8 Concept check · 2 min

A soft sensor at a tank outlet is corrected by laboratory results drawn after a pipe, each available one sampling interval after its draw. The update compares each result with the soft sensor's value at the moment the result arrives, with $\lambda = 1$. After a step error of the soft sensor, what does the bias do?

(a) It removes the error with the first result that contains it
(b) It converges as $(1-\lambda)^n$, here in one step
(c) It grows without bound from the first result on
(d) It oscillates without decaying, with a period of six results

1.9 Theory and calculation · 3 min

An at-line analyzer runs a method of 4 min on each sample, and the sample needs 1 min to reach it. Product flows at 1.5 m^3/min past the sample point toward a diverter valve. Find the longest time from the start of a change to the result that reports it, and the volume of hold-up needed between the sample point and the valve.

1.10 ○ Concept check · 2 min

A moving horizon estimator uses the same linear model and noise weights as a steady Kalman filter, with the filter's prediction and covariance as its arrival cost. Its states carry bounds, none of which is active. How does its estimate compare with the filter's?

(a) It equals the filter's estimate

(b) It is smoother, because it fits a window

(c) It lags the filter's by half the window

(d) It is biased toward the arrival cost