INTERACTIVE CONTROL ENGINEERING
Principles and Applications

Volume 1 Process Dynamics, Modeling, and Identification

CHAPTER 4

A Two-Heater Digital Twin That Runs Beside the Device It Models

Pages 84–111

Edited by John D. Hedengren

How to cite this chapter. "A Two-Heater Digital Twin That Runs Beside the Device It Models," in *Process Dynamics, Modeling, and Identification*, John D. Hedengren, Ed., vol. 1 of *Interactive Control Engineering: Principles and Applications*. Advanced Process Solutions, LLC, 2026, ch. 4, pp. 84–111.

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

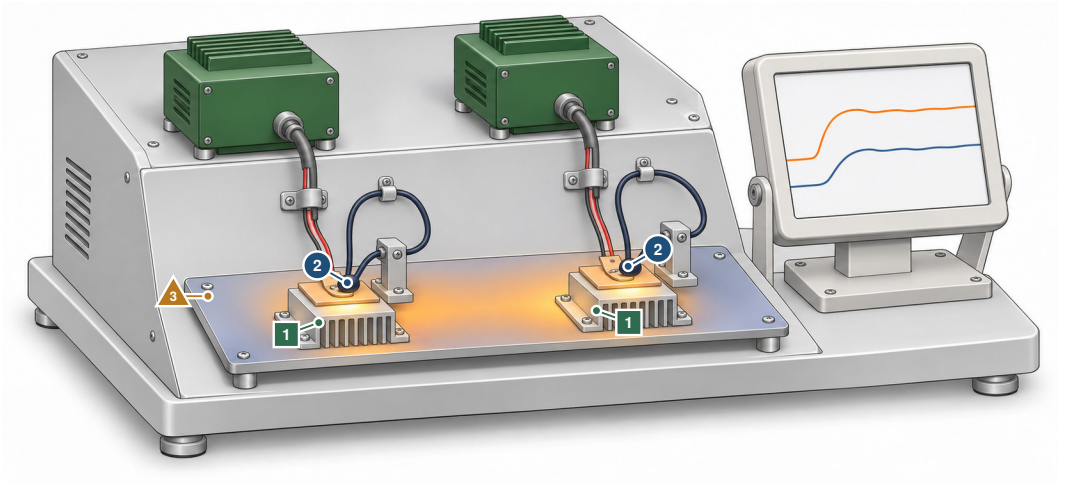
The app for this chapter
https://apmonitor.com/apps/tclab/

The volume online: contents, other chapters and citation records
https://apmonitor.com/apps/books/v01-dynamics-modeling/#tclab

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

CHAPTER 4

A Two-Heater Digital Twin That Runs Beside the Device It Models



The controller moves 1 the power to each of the two heaters

The controller reads 2 the temperature sensor beside each heater

What disturbs it 3 the room air around the board

Notice the warmth spreading across the shared plate, so each sensor reads its own heater and some of its neighbor, on a real board far smaller than drawn and with the power switching on the board itself rather than in separate boxes. Illustration, not to scale.

IN PLAIN TERMS **Two Heaters That Warm Each Other**

The main idea. A small board holds two heaters and two temperature sensors, and a controller sets the power of each heater to hold its sensor at a target. A computer model of the board runs beside it, so real and predicted temperatures share one screen.

The challenge. Each sensor warms more slowly than its heater, so the controller acts on old news. The heaters also warm each other through the board, so a change on one side upsets the other.

Why it matters. Large industrial heaters have the same slow sensors and shared heat, and a test on a model costs far less than a test on the plant.

THREE WAYS TO READ THIS CHAPTER

⊘	**Explore**	11 min	the first page, the history of take-home kits (Section 4.2), the fact on thermistor wiring (Section 4.3), the try-it on the lag (Section 4.4), the try-it on the PID types and the fact on heater travel (Section 4.7) and the key points
🕮	**Study**	60 min	the whole chapter with its three pause points and the twelve exercises at its end
⛰	**Practice**	20 min	the first page, the results (Section 4.7), Technology in Practice (Section 4.8), Limitations (Section 4.9) and the open challenge

The control question Two heaters share one small board, and each thermistor lags its heater by more than two minutes. What does a step test reveal about the instrument and the coupling, what do the three velocity forms of PID do differently, and how far does a twin that is its controller's own model speak for the device beside it?

Learning objectives

- Derive the steady-state gain matrix and the relative gain of the two-heater balance, and the heater powers that hold a pair of setpoints.
- Fit a first-order plus dead-time model to a heater step and explain why its time constant belongs to the thermistor and its dead time to the heater.
- Predict the first-cycle heater move of the three velocity forms on a setpoint step, and explain why they coincide on a load step.
- Compute the heater travel that sensor noise causes through the proportional and derivative terms.
- Explain how a predictive controller with the conduction path in its model holds the second loop where two PID loops cannot, and why the result is a best case.

Prerequisites Sections 1.4.2 (energy balance), 1.4.4 and 1.4.6 (lags in series and apparent dead time), 1.4.8 (two-point fit) and 1.6.2 (velocity form). Section 1.6.3 is needed only for the predictive controller and its estimators.

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

See also Volume 2, Chapter 4 for step tests and tuning on an industrial thermal loop, Volume 2, Chapter 2 for the velocity-form PID in detail and Volume 2, Chapter 3 for a relay test placed by its switching band. Volume 5, Chapter 11 for a temperature that no instrument reaches and Volume 5, Chapter 2 for a temperature loop through a fuel delay.

4.1 Introduction

The Temperature Control Lab (TCLab) is a shield for an Arduino board carrying two transistors used as heaters and two thermistors used as temperature sensors, and it was introduced as a hardware benchmark for modeling and control methods precisely because a physical device imposes constraints that a simulation study does not, including a cycle time that must be met, discrete sampling, communication overhead and model mismatch [1]. It has since carried undergraduate proportional-integral-derivative (PID) experiments and remote laboratory sections in which every student holds the same board [2]. The device is inexpensive and slow enough that a full step response fits inside a class period.

The physical fundamentals are the ones a first course in process dynamics establishes, and Chapter 1 develops them on a heated tank (Section 1.4.2). Each heater is an energy balance on a lumped capacitance driven by electrical dissipation and losing heat to the surroundings and to its neighbor, and each thermistor is a second capacitance connected to its heater through a thermal resistance, so the measured temperature is separated from the state that generates it by a first-order lag [3]. That is the structure of the heater path of the running example behind its thermowell, Equation (1.14), without the transport delay. The experiments that follow from that structure are step testing and first-order plus dead-time identification, steady-state gain and interaction between two loops, the difference between a state and a measurement, sampled control with a finite cycle time, actuator saturation and anti-windup, derivative action on error against derivative action on measurement, and the amplification of measurement noise by a derivative term [4, 5]. Thermal loops with this structure, a slow plant behind a lagging measurement and an actuator that saturates at both ends of its range, are common throughout process plants, so what is learned on a two-transistor board transfers to equipment that is orders of magnitude larger.

Simulation and hardware are usually kept apart when the TCLab is taught. A student first runs a simulation to choose gains, then connects the board and runs the same gains on hardware, and the two runs are carried out in separate scripts and often in separate weeks, so the comparison between prediction and measurement is reconstructed after the fact. However, the model and the device can

be driven from the same controller instance in the same process, so that a single setpoint change produces two trends on one time axis and the mismatch is read directly.

This chapter works with a four-state TCLab simulation and a physical TCLab under one velocity-form PID implementation inside a single browser page. It measures what the three selectable arrangements of that implementation do to a setpoint change and to a load disturbance. Lumped thermal modeling and PID structure are both long settled. The chapter writes down the twin, the serial protocol layer, the latched cutoff and the three difference equations in enough detail to be reproduced elsewhere, and reports closed-loop numbers from the running application that separate setpoint tracking from disturbance rejection across the three arrangements. The experiments run from a manual step test to hardware running beside its own prediction.

Section 4.2 traces laboratory practice from shared rigs to boards a student keeps. Section 4.3 states every constant needed to rebuild the twin, with its limits, numerics and assumptions. Section 4.4 derives the steady-state gain matrix in closed form, checks it against the running application and fits a first-order plus dead-time model to the step response. Section 4.5 gives the three PID difference equations and three options beside them: a model predictive controller (Section 4.5.2), a moving horizon estimator for it (Section 4.5.3) and a relay test that writes the PID gains (Section 4.5.4). Section 4.6 covers the serial protocol and the safety latch. Section 4.7 measures the three PID arrangements on a setpoint step, an ambient load step and sensor noise. It then compares the predictive controller with PID on the twin, which at the default parameters is that controller's own model, so the comparison is a matched-model best case with no measurement from a physical board (Section 4.7.5), and measures the estimator and the relay on the twin alone (Section 4.7.6). Section 4.8 sets the board's devices beside their industrial counterparts, and Sections 4.9 and 4.10 state where the twin stops holding and what transfers to plant practice.

4.2 Background

Laboratory instruction in control has moved along a path that hardware cost determined at every stage. Physical rigs were expensive, so they were shared, scheduled and supervised, and the response was first to instrument them for remote access and then to replace them with simulations. Comparative reviews of the three arrangements, hands-on, simulated and remote, found that they are not substitutes for one another because they serve different objectives, with the hands-on arrangement retaining the design and diagnosis skills that the other two shed [6]. A bibliometric survey of the field counted the growth of virtual and remote laboratories across two decades and mapped the communities that built them [7], and an earlier review of control learning set out the interactive tools that the discipline had accumulated by then [8].

The economics reversed once microcontrollers became cheap. A portable platform assembled from low-cost components was shown to support hands-on work in control and systems theory at a scale that shared rigs never could, with each student keeping a board [9]. The TCLab belongs to this generation of devices. It is a thermal plant rather than an electromechanical one, with time constants of the order of minutes and no stored mechanical energy, so an unstable tuning is observed rather than repaired. The history box below follows kits of this kind to 2018, and the laboratory's role in engineering education has been argued from the objectives it serves rather than from the equipment it owns [10].

> ⧖ HISTORY **Laboratories Students Take Home, 2013 to 2018**
>
> Laboratory hardware for each student was long out of reach because of its cost. A survey of control education published in 2018 put a single piece of laboratory equipment at 5,000 to 10,000 pounds sterling. Hardware once driven from an expensive input and output board inside a computer could by then run from its USB port through a simple microcontroller board. By 2013 a university had built a take-home kit, a small helicopter on three axes driven by two fans, for under 300 pounds of parts each, enough to lend one to every student in the course. The survey expected such kits to grow significantly [11].

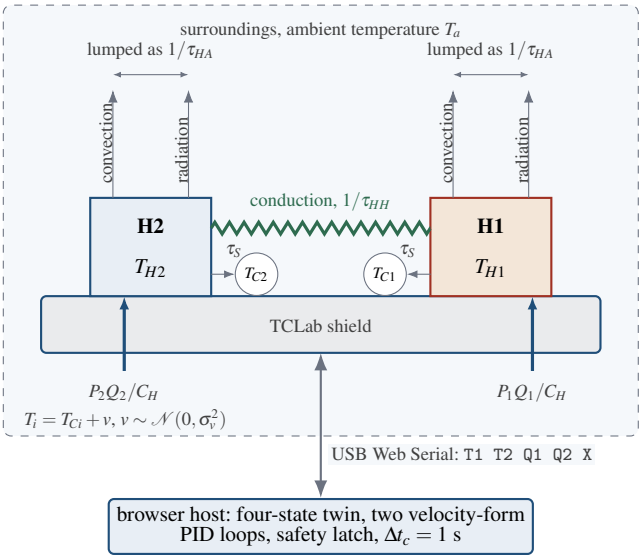


Figure 4.1 Two-heater Temperature Control Lab and the four states of its browser twin. Heater powers Q_1 and Q_2 are commanded from 0 to 100%, the two heater temperatures exchange heat through the board, both lose heat to the surroundings, and each thermistor temperature follows its heater through the lag τ_S.

The browser is the part of this history that is recent. Web Serial gives a page direct access to a USB serial device without a driver, a helper application or an installed runtime, so the same page that carries the simulation can also carry the hardware session. The application of this chapter uses that capability to put the twin and the device under one controller.

4.3 Process Model

The physical TCLab holds two heaters and two sensors on one small board. Heater 1 and its thermistor sit at one end, heater 2 and its thermistor at the other, and the board itself conducts heat between them. Both heaters lose heat to the room by convection from their packages and by radiation from their surfaces. Each thermistor is mounted beside its transistor rather than on it, so the measured temperature follows the heater temperature through a thermal resistance and lags it. Figure 4.1 draws that arrangement and carries every symbol used below.

4.3.1 Signal Classification

The two controlled variables are the reported sensor temperatures. The two manipulated variables are the heater duty cycles, which the firmware accepts as percentages and applies by pulse-width modulation. The single disturbance available in the twin is the ambient temperature, which is also the reset condition. An adjustable room temperature is deliberately artificial, since no room obeys a slider, but it supplies the only load disturbance that leaves both setpoints untouched. The heater temperatures are states that no instrument measures, on the twin or on the board. Table 4.1 lists the classification and the ranges the application enforces.

🕮 **Pause.** Try Exercise 4.2 before reading on.

4.3.2 Energy Balances

Each heater is treated as a lumped mass at a single temperature. Its energy balance, in the form of Equation (1.1), carries three terms, which are the electrical power dissipated in the transistor, the loss to the surroundings and the conduction to the other heater. Equations (4.1) and (4.2) give the pair.

Table 4.1 Signal Classification and Ranges

Symbol	Role	Units	Range
T_1, T_2	controlled, reported	°C	unbounded in the twin
Q_1, Q_2	manipulated	%	0 to 100
T_a	disturbance and reset	°C	0 to 45
T_{H1}, T_{H2}	state, unmeasured	°C	unbounded
T_{C1}, T_{C2}	state, measured with noise	°C	unbounded
SP_1, SP_2	setpoint	°C	20 to the cutoff
K_p	controller gain	%/°C	0 to 50
K_i	controller gain	%/(°C s)	0 to 5
K_d	controller gain	% s/°C	0 to 50

$$\frac{\mathrm{d}T_{H1}}{\mathrm{d}t} = \frac{P_1 Q_1}{C_H} + \frac{T_a - T_{H1}}{\tau_{HA}} - \frac{T_{H1} - T_{H2}}{\tau_{HH}} \tag{4.1}$$

$$\frac{\mathrm{d}T_{H2}}{\mathrm{d}t} = \frac{P_2 Q_2}{C_H} + \frac{T_a - T_{H2}}{\tau_{HA}} + \frac{T_{H1} - T_{H2}}{\tau_{HH}} \tag{4.2}$$

where T_{H1} and T_{H2} are the heater temperatures, Q_1 and Q_2 are the duty cycles in percent, P_1 and P_2 are the dimensionless heater capacities, C_H is a fixed capacity constant, τ_{HA} is the heater-to-ambient time constant and τ_{HH} is the heater-to-heater time constant. The conduction term enters the two balances with opposite signs, so the total energy exchanged between the heaters is conserved. The group $P_1 Q_1 / C_H$ has units of °C/s, which fixes the units of C_H at 5720 % s/°C for a dimensionless capacity.

The published physics-based TCLab model separates a convective loss proportional to $T_a - T_H$ from a radiative loss proportional to $T_a^4 - T_H^4$, and the radiative contribution is what makes the measured steady-state slope fall below unity at elevated temperatures [1]. The twin implemented here folds both mechanisms into the single linear coefficient $1/\tau_{HA}$, which is the form used by the reference simulation shipped with the TCLab software package for offline work. The consequence is that the twin is linear in the states and exactly reproducible, and that it will not reproduce the loss of steady-state gain that a real board shows above roughly 55 °C. The default τ_{HA} of 20 s is therefore best read as a fit over the operating band rather than as a convection coefficient.

4.3.3 Sensor States and the Reported Measurement

Each thermistor is a separate state that relaxes toward its own heater temperature with a single time constant. Equations (4.3) and (4.4) are the two sensor relations.

$$\frac{\mathrm{d}T_{C1}}{\mathrm{d}t} = \frac{T_{H1} - T_{C1}}{\tau_S} \tag{4.3}$$

$$\frac{\mathrm{d}T_{C2}}{\mathrm{d}t} = \frac{T_{H2} - T_{C2}}{\tau_S} \tag{4.4}$$

where τ_S is the sensor time constant, taken as identical for both channels. What the interface reports is the sensor state plus zero-mean Gaussian noise, given by Equation (4.5).

$$T_i = T_{Ci} + \sigma_v z, \qquad z \sim \mathcal{N}(0,1), \qquad i = 1,2 \tag{4.5}$$

where σ_v is the noise standard deviation and z is generated by the Box-Muller transform from two uniform variates. The noise is drawn fresh on every read and is not part of the state, so it does not accumulate. Nothing in the twin models thermistor quantization, self-heating or the calibration offset that separates two physical boards.

Table 4.2 Twin Parameters, Controller Defaults and Limits

Symbol	Meaning	Default	Range, visibility
T_a	ambient temperature	21 °C	0 to 45, adjustable
P_1	heater 1 capacity	200	50 to 255, adjustable
P_2	heater 2 capacity	100	25 to 255, adjustable
τ_{HA}	heater to ambient	20 s	5 to 100, adjustable
τ_{HH}	heater to heater	100 s	20 to 300, adjustable
τ_S	sensor lag	140 s	20 to 300, adjustable
σ_v	sensor noise	0.2 °C	0 to 1, adjustable
C_H	heater capacity constant	5720	fixed, hidden
h	integration step	0.1 s	fixed, hidden
Δt_c	nominal control interval	1 s	fixed
K_p	proportional gain	2	0 to 50, adjustable
K_i	integral gain	0.1	0 to 5, adjustable
K_d	derivative gain	1	0 to 50, adjustable
SP_1	loop 1 setpoint	45 °C	20 to cutoff
SP_2	loop 2 setpoint	40 °C	20 to cutoff
T_{cut}	hardware cutoff	70 °C	50 to 90, adjustable
s	simulation speed	1×	1 to 10, adjustable

✦ **DID YOU KNOW? Wires That a Thermistor Ignores**

A thermistor's resistance runs to tens or even hundreds of thousands of ohms. Two ohms of lead wire are 1.96 % of the circuit of a 100 Ω platinum resistance thermometer but only 0.004 % of a 50 kΩ thermistor, so the thermistor's own resistance swamps its wiring [12].

4.3.4 Parameters, Limits and Initial Conditions

Parameters are listed in Table 4.2. Seven of them are operator adjustable and take effect on the running simulation, which makes the twin a small parameter-estimation problem in its own right. Two are hidden. The capacity constant C_H is compiled into the model and is never displayed, and the integration step is fixed.

A reset sets all four states to T_a and both heater outputs to zero, so the twin always starts isothermal with the room. Heater commands are clamped to the interval 0 to 100% before they reach the model, and the twin has no upper temperature bound of its own. With $P_1 = 200$ and $Q_1 = 100\%$ the twin settles at 80.9 °C, which is above both the default hardware cutoff and the ceiling of a real board, so a heater left at full power in simulation reaches a temperature that the device it models cannot.

4.3.5 Numerical Integration and Timing

All four derivatives are evaluated at the old state and advanced with an explicit Euler step of $h = 0.1$ s, as shown in Equation (4.6).

$$x_{n+1} = x_n + h\, f(x_n, Q_1, Q_2), \qquad h = 0.1\,\mathrm{s} \tag{4.6}$$

where $x = [T_{H1}, T_{H2}, T_{C1}, T_{C2}]$. The step is two orders of magnitude below the smallest eigenvalue time constant of the model, so the Euler error is not visible at the resolution of the charts. The simulation speed control from 1× to 10× multiplies the number of 0.1 s substeps taken per timer tick rather than enlarging h, which preserves the integration step at every speed. A speed of 10× compresses a 45 min experiment into about five minutes, which no physical board allows. Unlike the framework apps of Section 1.6.1, where the speed never changes what a run computes, this standalone app does not guarantee identical closed-loop histories when control timing or noise reads differ, and its noise is not seeded.

Two clocks run in the application and they are deliberately different. The physics loop advances simulated time in fixed substeps. The control loop measures wall-clock time between executions, multiplies it by the speed factor and clamps the result to the interval 0.05 s to 3 s, and that clamped value is the Δt used in the integral and derivative terms. The nominal control interval is 1 s of simulated time at any speed. When a physical board is connected the speed factor is forced to $1\times$, because the hardware cannot be accelerated.

4.3.6 Assumptions

The twin rests on six assumptions, each of which breaks down somewhere.

Each heater is isothermal. The transistor package, its leads and the copper around it are treated as one temperature. This fails during the first seconds after a large step, when the junction is far hotter than the package.

The loss to the surroundings is linear in temperature difference. Radiation is not linear, and the error grows with the fourth power of absolute temperature, so the twin overpredicts the steady-state rise at high heater settings [1].

The ambient temperature is uniform and known. On a real board the air near the heaters is warmer than the room, and the ambient reading itself drifts, which is one reported source of run-to-run variation in TCLab benchmarks [1].

The two sensors share one time constant and the two heaters share one loss coefficient. Physical boards differ from one another and the two channels on a single board differ, which is why the twin cannot be expected to overlay a particular device without re-estimating the parameters.

The heater capacities are constant with duty cycle. Pulse-width modulation into a transistor whose gain depends on temperature is not exactly linear in duty cycle.

Measurement noise is white and Gaussian. Thermistor readings quantized by a ten-bit converter are neither, although the resulting difference is small at $\sigma_v = 0.2\,^{\circ}\mathrm{C}$.

4.4 Steady-State Gains and Open-Loop Dynamics

The steady state of Equations (4.1) and (4.2) is a two-by-two linear solve, and the sensor states equal the heater states once the derivatives vanish, so the steady-state gain matrix follows in closed form. The gains are collected in Equation (4.7), evaluated at the default parameters.

$$K = \begin{bmatrix} \partial T_1/\partial Q_1 & \partial T_1/\partial Q_2 \\ \partial T_2/\partial Q_1 & \partial T_2/\partial Q_2 \end{bmatrix} = \begin{bmatrix} 0.599 & 0.050 \\ 0.100 & 0.300 \end{bmatrix} \ ^{\circ}\mathrm{C}/\% \tag{4.7}$$

The diagonal gains differ by a factor of two because the heater capacity $P_2 = 100$ is half of $P_1 = 200$, which reproduces the asymmetry of the physical shield, and the off-diagonal gains inherit the same ratio through the conduction path. The relative gain array of K has $\lambda_{11} = 1.029$, so the diagonal pairing of T_1 with Q_1 is correct and the steady-state interaction is weak [4]. Interaction is nevertheless present in the transient, and Table 4.3 shows how much.

A manual step of Q_1 from 0 to 50% with Q_2 held at zero, taken from the application with the noise disabled and the speed set to $10\times$, gives $T_1 = 50.96\,^{\circ}\mathrm{C}$ and $T_2 = 25.99\,^{\circ}\mathrm{C}$ after 1140 s. The corresponding gains of 0.599 °C/% and 0.100 °C/% agree with Equation (4.7) to three decimal places, which is the check that the running twin implements the equations stated above. A second manual step of Q_2 gives the other column of K.

Table 4.3 Manual Step Tests from the Running Application, $\sigma_v = 0$

Response	K (°C/%)	t_{28} (s)	t_{63} (s)	τ_p, θ_p (s)
T_1 to Q_1	0.599	65	159	141, 18
T_2 to Q_1	0.100	82	177	142.5, 34.5
T_1 to Q_2	0.050	82	177	142.5, 34.5
T_2 to Q_2	0.300	65	159	141, 18

TRY IT IN THE APP 1 · Watch the Lag

Set the sensor noise to 0 in Settings and the speed to 10×. In Manual, type 50 for Q_1 with Q_2 at 0 and watch the two temperature trends. T_1 starts slowly, then climbs, and stands at 50.96 °C about 1140 s after the step. T_2, warmed only through the board, rises to 25.99 °C.

The dynamics are dominated by an instrument, not by the plant. The heater subsystem alone has eigenvalue time constants of 20 s and 14.3 s, obtained from the two-by-two state matrix of Equations (4.1) and (4.2), while each sensor contributes a pole at $\tau_S = 140$ s. A two-point fit of the measured T_1 response, taking $\tau_p = 1.5(t_{63} - t_{28})$ and $\theta_p = t_{63} - \tau_p$ as in Equation (1.18) [4], returns a first-order plus dead-time (FOPDT) approximation whose constants are $\tau_p = 141$ s and $\theta_p = 18$ s, so the identified time constant approximates the thermistor lag and the identified dead time is the heater dynamics collapsed into a delay. The model has no transport delay at all, so this dead time is the effective dead time of Section 1.4.6, the value a method reads from lags in series, as for the two tanks of Chapter 3. Off-diagonal responses fit the same τ_p with roughly twice the apparent dead time, 34.5 s, because heat has to cross the board before it reaches the second sensor. A student who tunes from the step test alone is therefore tuning against the sensor, and speeding up the loop past that lag is not possible with heater power as the only handle. The FOPDT constants are what a tuning correlation consumes, and the ratio $\theta_p/\tau_p = 0.13$ places this loop in the lag-dominant regime where the achievable closed-loop speed is set by the model error rather than by the delay [13].

Pause. Try Exercise 4.1 before reading on.

4.5 Controller Design

Both loops use the velocity form of Section 1.6.2 in code of their own, which keeps an independent integral contribution for anti-windup [14, 15]. The three increments are given by Equations (4.8) to (4.10), with $e_k = \mathrm{SP} - y_k$ and y_k the reported temperature at cycle k.

$$\Delta u_{P,k} = \begin{cases} K_p\,(e_k - e_{k-1}) & \text{Types A and B} \\ -K_p\,(y_k - y_{k-1}) & \text{Type C} \end{cases} \tag{4.8}$$

$$\Delta u_{I,k} = K_i\, e_k\, \Delta t \tag{4.9}$$

$$\Delta u_{D,k} = \begin{cases} \dfrac{K_d\,(e_k - 2e_{k-1} + e_{k-2})}{\Delta t} & \text{Type A} \\ -\dfrac{K_d\,(y_k - 2y_{k-1} + y_{k-2})}{\Delta t} & \text{Types B and C} \end{cases} \tag{4.10}$$

where K_p, K_i and K_d are independent parallel gains rather than a controller gain with reset and rate times, and Δt is the measured control interval. Type A places error on all three terms and is the textbook velocity algorithm. Type B places derivative on the measurement, which is the default in industrial controllers because it removes the derivative spike that a setpoint step would otherwise produce [16]. Type C places both proportional and derivative on the measurement, so only the integral term sees a setpoint change directly. The application defaults to Type B.

The three increments are combined and the demand is clipped to the actuator span, as shown in Equation (4.11).

$$\begin{aligned} s_k &= s_{k-1} + \Delta u_{P,k} + \Delta u_{D,k} \\ c_k &= s_k + I_{k-1} + \Delta u_{I,k} \\ u_k &= \min\bigl(100,\ \max(0,\ c_k)\bigr) \\ I_k &= I_{k-1} + \Delta u_{I,k} + (u_k - c_k) \end{aligned} \tag{4.11}$$

where s_k accumulates the proportional and derivative increments without limit, c_k is the heater demand the algorithm computes and I_k carries the integral contribution and absorbs the whole of the clipping through the term $(u_k - c_k)$, which is applied only when $K_i \neq 0$. In the symbols of Equations (1.25) to (1.27), s_k is S_k, c_k plays the part of v_k with no bias and with the present integral increment included, and the back-calculation acts in a single execution, as a tracking time equal to Δt would. Substituting the third line into the fourth gives $c_k = u_{k-1} + \Delta u_{P,k} + \Delta u_{I,k} + \Delta u_{D,k}$, so whenever $K_i \neq 0$ the recursion is algebraically identical to clipping the running sum and every result of this chapter is unchanged by writing it this way. The two forms differ only at $K_i = 0$, where clipping the running sum would turn symmetric measurement noise into a drift away from the limit (Section 1.6.2).

The correction $(u_k - c_k)$ is the entire anti-windup mechanism [14, 17]. It removes from the integral exactly the part of the demand the limit refused, so an increment computed against a limit is absorbed rather than stored and the heater comes off the limit on the first increment that points back into range. There is no separate tracking time constant, no positional reset state and no rate limit on u. The 1 s nominal cycle limits update frequency but does not impose a bound on heater-command slew rate.

Both loops are reverse acting in the ISA sense, meaning that the controller output falls as the measurement rises, which is the required action because the process gain from heater power to temperature is positive. There is no bias term and no manual reset. Transfer into automatic is handled by seeding at the switch, as shown in Equation (4.12), where the framework apps of Section 1.6.2 instead keep the controller executing in Manual.

$$\begin{aligned} s_{k-1} &\leftarrow \mathrm{clip}(u_{\mathrm{manual}}), \qquad I_{k-1} \leftarrow 0 \\ y_{k-1} = y_{k-2} &\leftarrow y \\ e_{k-1} = e_{k-2} &\leftarrow \mathrm{SP} - y \end{aligned} \tag{4.12}$$

Pre-loading two samples of history with the present measurement and the present error leaves only the integral increment in the first automatic cycle, so the transfer adds no proportional or derivative step. The same seeding is applied whenever the PID type is changed, so switching between Types A, B and C mid-run does not bump the output. Four PID controller objects exist at all times, one for each loop of the twin and one for each loop of the device. They share the gains, the setpoints and the type, and they keep independent histories of s, I, y and e.

4.5.1 Equivalence of the Three Forms at a Fixed Setpoint

With the setpoint constant, $e_k - e_{k-1} = -(y_k - y_{k-1})$ and $e_k - 2e_{k-1} + e_{k-2} = -(y_k - 2y_{k-1} + y_{k-2})$, so Equations (4.8) and (4.10) collapse onto a single expression and the three types produce identical increments. The distinction between them exists only while a setpoint is moving. On the cycle at which the setpoint steps by $\Delta\mathrm{SP}$, the immediate change in output is given by Equation (4.13).

$$\Delta u_k = \begin{cases} K_p\,\Delta\mathrm{SP} + \dfrac{K_d\,\Delta\mathrm{SP}}{\Delta t} + K_i e_k \Delta t & \text{Type A} \\ K_p\,\Delta\mathrm{SP} + K_i e_k \Delta t & \text{Type B} \\ K_i e_k \Delta t & \text{Type C} \end{cases} \tag{4.13}$$

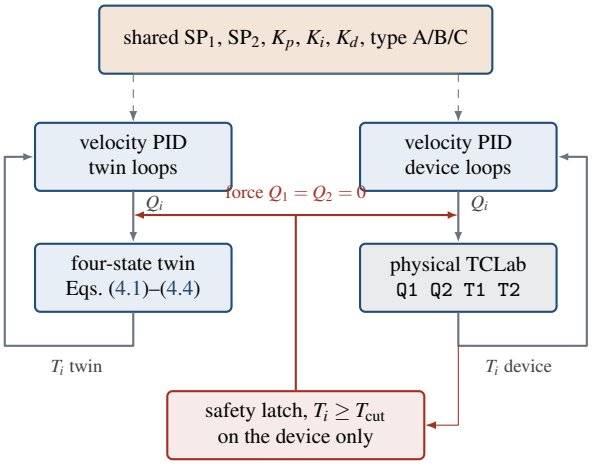


Figure 4.2 One set of gains driving two plants. The twin and the device each keep their own controller history of output, measurement and error, and the latched cutoff is armed by the physical measurements alone.

Type A adds a further $-K_d\,\Delta\mathrm{SP}/\Delta t$ on the following cycle, so the derivative kick is a one-cycle pulse whose net contribution to the output is zero. Type C moves the heater by the integral increment alone, which at the default gains is about 1% of full scale. Figure 4.2 shows the two parallel loops and the point at which the safety latch intervenes.

4.5.2 Model Predictive Control

A model predictive controller is the third entry of the app's mode switch, after Manual and PID auto. It is an option beside the PID loops, and PID auto remains the default closed loop. Where each loop reads one thermistor and moves one heater, the predictive controller moves both heaters from one plan. Every $T_s = 1\,\mathrm{s}$ a quadratic program chooses both duty cycles over $N = 90$ steps and applies the first move, the receding-horizon scheme of Section 1.6.3 [18, 19]. A connected board receives its own instance of the same controller.

The controller's model is Equations (4.1) to (4.4) at the declared defaults of Table 4.2, with $x = [T_{H1}, T_{H2}, T_{C1}, T_{C2}] - T_a$ in deviation from $T_a = 21\,^{\circ}\mathrm{C}$ and $u = [Q_1, Q_2]$. The balances are linear in temperature and power, so every operating point gives the same matrices, and holding u over each step gives Equation (4.14) exactly.

$$x_{k+1} = A_d\,x_k + B_d\,u_k, \qquad \hat{y}_k = C\,x_k + d_k, \qquad C = [0\ \ I_2] \tag{4.14}$$

Its steady-state gains are 0.5994, 0.04995, 0.0999 and 0.2997 °C/%, the matrix K of Equation (4.7) to four figures. No step test was fitted, because the twin's balance is the model. An independent re-derivation of A_d and B_d from the balances matches every entry exactly, and the one residual on the default twin is the 0.1 s Euler step, which departs from the exact hold by at most 0.0045 °C over 900 s at $Q_1 = 50\%$.

The model runs open loop on the duty cycles actually applied, whether the PID loops, the plan, manual operation or the latch set them, and d_k is an output bias filtered by Equation (4.15).

$$d_k = d_{k-1} + \alpha\left(y_k - C\,x_k - d_{k-1}\right), \qquad \alpha = 0.1 \tag{4.15}$$

Here y_k is the pair of reported temperatures less T_a, and the filter's time constant is about 10 s. Held over the horizon, the bias makes the prediction offset free for a constant disturbance such as an

ambient step or a wrong parameter [20]. No preview enters the plan, which minimizes Equation (4.16).

$$J = \sum_{k=1}^{N} q \, \|\hat{y}_k - r\|^2 + S \sum_{m=1}^{12} \|\Delta u_m\|^2 + \sum_{i=1}^{2} \left(\rho_2 \, \sigma_i^2 + \rho_1 \, \sigma_i \right) \tag{4.16}$$

In it r is the setpoint pair less T_a, $q = 1$ per °C^2, and $S = 0.3$ per %^{2} weights the change Δu_m at each of twelve move blocks of $1, 1, 1, 1, 2, 2, 3, 4, 6, 9, 15, 45$ steps, with Δu_1 taken from the duty cycles applied now. The 24 decisions are the block inputs less those applied duty cycles. The constraints are Equation (4.17).

$$\begin{aligned} 0 \le u_{m,i} \le 100\%, \qquad & \sigma_i \ge 0 \\ \hat{y}_{k,i} \le T_{\text{cut}} - T_a - 1\,^\circ\text{C} + \sigma_i & \end{aligned} \tag{4.17}$$

The duty limits are hard, the same limits the PID loops clip to. The ceiling is soft, with one slack per temperature priced at $\rho_2 = 10^5$ per °C^2 and $\rho_1 = 10^4$ per °C so that it outranks any tracking error. It reads the configured cutoff as the latch does and sits 1 °C below it, because a plan held at the cutoff would leave a reading one noise sample from a trip, and the latch remains the protection. S is the largest of seven trial values whose S3 and noise-driven travel stayed within Type B's and whose IAE stayed within 10% of the best, and α the smallest of five whose S4 IAE stayed below Type B's on both loops, a rule set after that sweep had been seen.

A dual active-set method solves each plan [21] and needs a ridge ε on the Hessian diagonal. The shared solver's default, 10^{-9} times one plus the largest diagonal entry, would take the slack's $2\rho_2$ and give $\varepsilon = 2 \times 10^{-4}$, an unintended move weight. The controller passes $\varepsilon = 10^{-9}\,(1 + 2(qNg^2 + S)) = 6.63 \times 10^{-8}$ instead, with $g = 0.5994$ the largest gain. On an unconstrained test, the model lined out at 50 and 35 °C and SP_2 raised by 2 °C, the first move then agrees with a closed-form solution to 3×10^{-7} percentage points on Q_1, a move of 0.081%, while the default ridge shifts it by 9.5×10^{-4} points, 1.2% of the move.

While a plan acts, each PID loop tracks the applied duty, its output set to that value and its histories advanced with an empty integral, so its next execution is its ordinary velocity step. With the noise off at 45 and 40 °C, a switch from MPC to PID auto moved both heaters by 0.0000%, and one from PID auto to MPC moved Q_1 by -0.035%. The first move into the plan is its own decision, not a transfer, so after two minutes of manual operation at $Q_1 = 100$% during the S3 step it was -27.9% against -0.77% under PID. A switch to manual keeps the plan's last request exactly. Whenever the solver reports a status other than optimal, or its first move is undefined or outside 0 to 100%, the plan is discarded, the tracked loops set the heaters for that cycle, the record is marked and a notice posted, and the next valid plan takes over again. A rejection forced 30 s after the S3 step handed over the PID's own step of 0.163%, and the run rejoined the unforced one within 0.0085 °C.

The safety latch of Section 4.6.1 sits above the plan. On a trip both heaters go to zero whatever the plan requested, the interface returns to manual, and no plan is solved while the alarm is latched. The three models of Section 1.4.9 are kept apart. The plant is the twin at whatever parameters are set, the instrument adds the noise of Equation (4.5), and the controller's model is Equation (4.14) at the declared defaults. The controller reads only the reported temperatures, the setpoints, the cutoff and the applied duty cycles, so an edited twin and every physical board are model mismatch, and each of ten deliberately altered copies that reads the twin's true state or its operator-set parameters fails the application's tests. A plan took 326 and 329 µs on average over 1000 plans in two Node measurements, at one-minute load averages of 3.50 and 1.95. In headless Chrome on the built application the controller used 0.19 and 0.20% of one core at 1× and 0.47 and 0.63% at 10×, at load averages of 3.45 and 2.77.

4.5.3 Moving Horizon Estimation

The output bias of Equation (4.15) shifts the prediction but keeps the declared gains and speed, so a weaker heater or a faster loss is planned for with the declared plant. A moving horizon estimator

(MHE) may therefore feed the predictive controller instead, fitting an augmented balance to the latest readings with its parameters held inside bounds [22]. The Estimator switch under MPC selects it. The output bias remains the default and keeps running under either choice. The estimated heater temperatures are states no instrument reaches, the same problem as the brick core of Volume 5, Chapter 11 on a far smaller scale. The augmented balance replaces Equations (4.1) and (4.2) with Equation (4.18).

$$\begin{aligned} \frac{\mathrm{d}T_{H1}}{\mathrm{d}t} &= g_1 \frac{P_1 Q_1}{C_H} + \ell\,(T_a + b - T_{H1}) - \frac{T_{H1} - T_{H2}}{\tau_{HH}} \\ \frac{\mathrm{d}T_{H2}}{\mathrm{d}t} &= g_2 \frac{P_2 Q_2}{C_H} + \ell\,(T_a + b - T_{H2}) + \frac{T_{H1} - T_{H2}}{\tau_{HH}} \end{aligned} \tag{4.18}$$

Here g_1 and g_2 are heater gains as multiples of the declared capacities, ℓ is the loss coefficient $1/\tau_{HA}$ and b is the offset of the room from the declared $T_a = 21\,^{\circ}\mathrm{C}$, while every other constant and Equations (4.3) and (4.4) keep the defaults of Table 4.2. At $g_i = 1$, $\ell = 0.05/\mathrm{s}$ and $b = 0$ the balance is the twin's. The parameters $\theta = [g_1, g_2, \ell, b]$ are random walks joined to the four temperatures less T_a in the state z. The product $\ell\,(T_a + b - T_H)$ is linearized once per step, at the estimate held when the step enters the window, and held exactly over the measured Δt, which gives A_j and c_j, with an error of at most $0.012\,^{\circ}\mathrm{C}$ per step when ℓ and b are off by 0.01/s and $1\,^{\circ}\mathrm{C}$. Every second the estimator solves Equation (4.19) over the last 20 steps, $s = k - 20$.

$$\begin{aligned} &\min \ \|z_s - \bar{z}_s\|^2_{\bar{P}_s^{-1}} + \sum_{j=s}^{k-1} \|w_j\|^2_{W^{-1}} + \sum_{j=s}^{k} \|y_j - C z_j\|^2_{R^{-1}} \\ &\text{s.t. } z_{j+1} = A_j z_j + c_j + G w_j, \qquad \theta_{\min} \le \theta_j \le \theta_{\max} \end{aligned} \tag{4.19}$$

In it C selects T_{C1} and T_{C2} and G drives the two heater states and the four parameters. R is $(0.2\,^{\circ}\mathrm{C})^2$ per channel, the declared σ_v, and W holds standard deviations per step of $0.05\,^{\circ}\mathrm{C}$ on each heater, 0.01 on each gain, 5×10^{-4}/s on ℓ and $0.05\,^{\circ}\mathrm{C}$ on b. The arrival cost carries the previous estimate forward as $\bar{z}_s$ with $\bar{P}_s$ from a Riccati recursion along the stored models. The bounds are the adjustable ranges of Table 4.2, never the values set, so g_1 lies in 0.25 to 1.275, g_2 in 0.25 to 2.55, ℓ in 0.01 to 0.2/s and b in -21 to $24\,^{\circ}\mathrm{C}$. The problem has 128 decisions and 168 bound rows for the solver of the plan [21], and 20 steps was the longest window of 10, 15, 20 and 30 whose estimate and plan stayed under 5 ms at the 99th percentile, a rule fixed before any run.

The plan of Equations (4.16) and (4.17) then starts from the estimated temperatures with Equation (4.18) frozen at $\hat{g}_1$, $\hat{g}_2$, $\hat{\ell}$ and $\hat{b}$, which is linear and is discretized exactly, and with every setting of the default path except the output bias, whose role $\hat{b}$ takes. The controller under this estimator is thus adaptive. Its 90 s prediction matches the twin at the estimated parameters to $1.3 \times 10^{-6}\,^{\circ}\mathrm{C}$. The estimator reads only the reported temperatures, the applied duty cycles and the declared constants, and eleven altered copies that pass the twin's state or the operator's settings to the estimator, the plan or the relay, seven of them by routes no search of the source finds, each fail the application's tests. A rejected window is replaced by a clipped Kalman update and a rejected plan by the tracked loops, and no measured run rejected either. On an unchanged plant a change of estimator moved the first duty by at most 0.019% from the unswitched run, but after a plant change the MHE restarts its parameters at their declared values, and 600 s after P_1 fell to 150 the duties then departed by up to 4.2 and 9.0% over 10 min with the noise off. In two Node.js measurements an estimate took a median of 1.79 and 1.80 ms and a 99th percentile of 2.80 and 2.87 ms, and in headless Chrome at $10\times$ the estimate and plan together reached 7.2 and 7.3 ms at the 99th percentile, about 7% of the 100 ms that one second of twin time lasts, on a loaded computer.

4.5.4 Relay Autotuning

The PID gains can also be measured by the relay experiment of [23], started by the Relay autotune button from wherever the loop stands under PID auto, and never a standing mode. Heater 1 switches

between $u_0 \pm d$, with u_0 the mean heater 1 duty over the last 60 s and $d = \min\big(20,\ 0.9\min(u_0, 100 - u_0)\big)\%$, each time T_1 crosses SP_1 by more than $\varepsilon = 3\sigma_v = 0.6\,^{\circ}\mathrm{C}$, while heater 2 holds its mean over that window and both loops track. Once two cycles of amplitude a agree within 5%, the describing function and the Tyreus-Luyben rule write the editable gains of Equation (4.20), where P_u is the measured period and D the fraction of it spent at $u_0 + d$.

$$K_u = \frac{4d\sin(\pi D)}{\pi\sqrt{a^2 - \varepsilon^2}}, \quad K_p = \frac{K_u}{3.2}, \quad K_i = \frac{K_p}{2.2\,P_u}, \quad K_d = 0 \tag{4.20}$$

The derivative stays at zero because it supplies about 45% of the noise-driven travel (Section 4.7.3). On this loop the experiment measures the band as much as the plant, as the relay on the blending tank of Volume 2, Chapter 3 also does. The chain of lags has no dead time, so the sampled loop reaches -180° only through the 1 s hold, at $K_u = 543.5\,\%/^{\circ}\mathrm{C}$ and $P_u = 17.2\,\mathrm{s}$, while a relay with hysteresis cycles where the phase falls short of -180° by $\psi = \arcsin(\varepsilon/a)$, predicted at 38.06 %/°C and 85.1 s. From the 45 and 40 °C steady state with the noise off it converges in 184 s at 37.97 %/°C and 84.1 s with $\psi = 41.8^{\circ}$, and a band of 0.3 or 1.2 °C moves K_u to 53.41 or 26.20. A change of mode or of SP_1, a trip or a reading 1 °C below the cutoff cancels it with the gains unchanged. Near the cutoff heater 1 then receives u_0, a step up by d from the low level, and in 30 noisy runs at setpoints of 68 to 69 °C 8 readings in 3 runs reached 70 °C afterward, where a board's latch acts as under PID.

4.6 Hardware Interface

Connecting hardware begins with a port request, which the browser grants only in response to a direct operator action. The transport then tries 115 200 baud and falls back to 9600 baud, opening the port, waiting 1500 ms for the board to finish its reset, and writing `VER`. A reply line matching the firmware identity pattern confirms that the device is a TCLab and not some other serial peripheral. On success the transport immediately writes `Q1 0`, `Q2 0` and `X`, so the board is known to be cold-started before any control runs.

The steady-state protocol is four request and reply exchanges per control cycle. The controller writes `T1` and `T2` and parses the two temperatures, computes the two heater commands from them, then writes `Q1 <value>` and `Q2 <value>` with two decimal places and reads back the value the firmware actually applied. Every command waits for one newline-terminated reply with a 1800 ms timeout, and all operations are serialized through a promise chain so that two cycles can never interleave on the wire. A timeout, a closed stream or a removed cable is treated as a disconnect, which zeroes both plants and reports the cause.

The ordering of that exchange is what makes the comparison fair. Reading before computing gives the device loop the same within-cycle measurement the twin loop uses, so neither loop carries a sample of dead time the other does not. The remaining asymmetry is the transport itself, because the four round trips and the firmware's own pulse-width modulation period sit between the measurement and the applied power on the hardware side and are absent on the simulated side. Against a 141 s process time constant the residual is small but is not zero, and it is one of the reasons the two trends do not overlay exactly.

4.6.1 The Safety Latch

The latch is the one part of the application that treats hardware and simulation differently. Its input is the physical measurement only. Whenever either reported device temperature reaches the configured cutoff, which defaults to 70 °C and is adjustable from 50 °C to 90 °C, the alarm latches and five things happen in the same cycle. Both simulated heater outputs are set to zero, the four controllers are reset, the interface is forced back to manual with zero demand, the device is sent `Q1 0`, `Q2 0` and `X`, and the recorded sample is written with zero heater columns.

TCLab firmware 2.1.0 also supplies its own device latch, tripping at 90 °C and permitting release at or below 85 °C. The application honors that status independently of its configurable cutoff and sends

the firmware acknowledgment command `A`. The connection bar also provides a firmware-loading action.

Clearing the latch requires the hazard to have passed. Acknowledgment is accepted only when both device temperatures are strictly below the cutoff, so the alarm cannot be dismissed while the board is still hot, and a board that is already above the cutoff when it is first connected latches immediately rather than accepting commands. Acknowledgment restores the mode the trip interrupted, with both controllers reseeded at zero output, so a loop that was in automatic returns to automatic without a proportional or derivative step. Integral action can change the zero output on the next execution. The cutoff itself cannot be edited while a device is connected or while an alarm is latched, and setpoints are clamped to the cutoff, so no target can be entered that would trip the protection it enforces.

Six further paths lead to the same shutdown, which are USB removal, an unexpected stream close, a command timeout, the page being hidden, the page being unloaded and the stop control. Each of them writes the three shutdown commands on a best-effort basis before releasing the port. The firmware's own command timeout remains the last independent layer, and no browser guarantee substitutes for it.

The simulated plant has no cutoff of its own. A twin driven to 100% heater power settles at 80.9 °C and stays there, which is intentional because the latch protects a transistor and there is no transistor in the twin, but it does mean that a simulation-only session can be run into a region that the hardware session would refuse.

4.7 Results

The experiments of this section are the five defined in Table 4.4. Reported reference trajectories use the shipped physics and controller functions with a fixed 1 s control interval and ten 0.1 s Euler steps between executions. Controller output is held between executions, and its input measurement is the value stored with that output. This protocol removes browser scheduling jitter from comparisons of PID types without changing the application law. It is a controlled timing reference, not a guarantee about an arbitrary browser's control interval. The Type B step is also checked by operating the built application with its browser timers advanced at their scheduled cadence and exporting the recorded data. Its first 890 post-step samples agree with the reference to CSV rounding (0.001 degrees for temperature and 0.01% for heater duty). The browser peak is 60.342 °C.

Closed-loop experiments begin at ambient with zero output and then run for 1800 s at the default setpoints before the disturbance or comparison window. At that instant both noise-free temperatures are within 0.01 °C of their targets. Step and load results use the next 890 s, and noise results use 600 s. IAE sums absolute error over that window, and heater travel sums successive command changes, including the initial setpoint-step change. Noise is zero except in the declared noise experiment. The physical hardware was not attached during these checks.

The application itself has no scenario selector. Its progress panel lists fourteen scenarios that the reader sets up by hand and that it judges from the recorded history, as Section 1.6.4 describes. Scenarios 1 to 3, Watch the lag, See the interaction and Find the asymmetry, hold one heater near 50% in Manual with the other off, for at least 300 s or, in scenario 2, until T_2 has risen at least 1 °C. Scenario 4, Step test, sets the sensor noise to zero and records a manual hold of Q_1 at 20% with Q_2 off, followed by 40%. Each hold lasts at least 600 s and must end settled, with the least-squares trend of T_1 over its last 120 s below 0.1 °C per minute. Scenario 5, Close the loops, runs the default gains and targets under PID for at least 600 s with both final errors below 1 °C. Scenario 6, Compare the PID types, steps the loop 1 target by at least 4.5 °C under each type and follows each step for 60 s. Scenario 7, One tuning, two loops, runs at least 300 s under PID with $P_2 = 100$ and again with $P_2 = 200$. Scenarios 8 to 11 need a board: a connection, 300 s of PID on the board with a target change, a physical T_1 of 60 °C in Manual, and a latched alarm at a cutoff no higher than 55 °C. Scenario 12, Make the interaction worse, lowers the coupling time to 25 s or less and follows a PID target change for 120 s. Scenario 13, PID against MPC, makes the same loop 1 step under each law

Table 4.4 Experiment Definitions

Label	Mode	Action
S1	manual	Q_1 0 to 50%, Q_2 held at 0
S2	manual	Q_2 0 to 50%, Q_1 held at 0
S3	PID A, B, C	settle at SP = 45/40, then SP_1 to 55 °C
S4	PID A, B, C	settle at SP = 45/40, then T_a 21 to 26 °C
S5	PID B	hold 45/40 with $\sigma_v = 0$ and with $\sigma_v = 0.2$ °C

Table 4.5 Scenario S3, SP_1 Stepped from 45 to 55 °C

Metric	Type A	Type B	Type C
First-cycle change in Q_1, %	31.0	21.0	1.0
Peak Q_1, %	89.8	90.0	86.6
Time for T_1 to first reach SP, s	84	84	105
Peak T_1, °C	60.32	60.34	59.86
Overshoot, % of the step	53.2	53.4	48.6
IAE over 890 s, °C s	1564	1570	1638
Q_1 travel, %	189.4	172.1	157.5
Peak T_2 deviation, °C	1.67	1.68	1.57

and follows each for 120 s. Scenario 14, Estimate the plant, tune by relay, runs at least 300 s under the MPC with the MHE after P_1, τ_{HA} or T_a has moved from its default, and follows a converged relay experiment for 120 s under PID. Eight scenarios are required. The four hardware scenarios, 13 and 14 are optional. Some predicates combine present settings with earlier history: scenario 7 does not require a fresh 300 s after P_2 changes, and scenario 12 can reuse an earlier target change. Scenarios 5, 6, 7, 9 and 12 judge automatic mode rather than the control law, so a run under the MPC also completes them, and scenario 6 then credits a Type A step that the plan, not the PID, drove.

4.7.1 Setpoint Step

Scenario S3 raises SP_1 by 10 °C at the default gains $K_p = 2$, $K_i = 0.1$ and $K_d = 1$, with loop 2 holding 40 °C throughout. Results are given in Table 4.5.

The first-cycle changes agree with Equation (4.13). Type A moves 31.0% of full scale, comprising 20% proportional action, a 10% derivative pulse and 1.0% integral action at the fixed 1 s interval. Type B omits the pulse and moves 21.0%, while Type C moves 1.0% through the integral term alone. On the following cycle Type A reverses 9.0% after its derivative pulse. Its total travel is 189.4% against 172.1% for Type B, a difference of 17.3 points. Type C requires 157.5% of travel because it has no proportional move at the step.

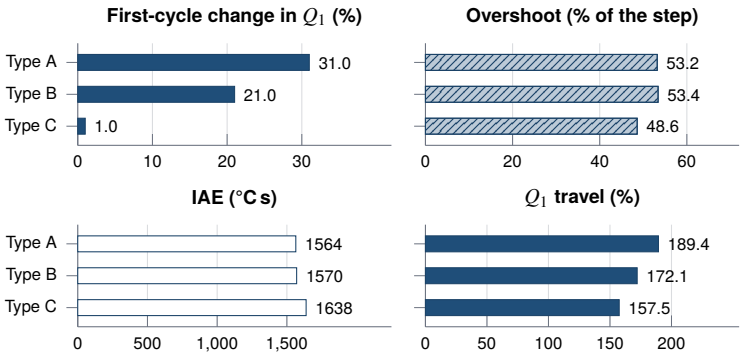


Figure 4.3 Scenario S3 of Table 4.5 under the three velocity forms of PID: the first-cycle heater move and the heater travel (solid bars), the overshoot (hatched) and the integrated absolute error (open). Notice that the first-cycle move falls from 31.0 to 1.0% while the overshoot stays between 48.6 and 53.4% of the step, and the IAE of Type C is only 4.3% above that of Type B.

Table 4.6 Scenario S4, Ambient Stepped from 21 to 26 °C

Metric, loop 1	Type A	Type B	Type C
Peak T_1 deviation, °C	1.178	1.178	1.178
IAE over 890 s, °C s	273.0	273.0	273.0
Q_1 travel, %	29.2	29.2	29.2
Final Q_1, %	28.78	28.78	28.78

TRY IT IN THE APP 6 · Compare the PID Types

With the sensor noise at 0, run PID auto at the default targets until both loops settle. Raise the loop 1 target from 45 to 55 °C under Type A, then repeat under Types B and C. Watch heater 1 on the first cycle after each step: it jumps 31.0, 21.0 and 1.0%, yet T_1 overshoots by about half the step every time (Table 4.5).

Type C first reaches the new target at 105 s, against 84 s for Type B, and increases IAE from 1570 to 1638 °C s, or 4.3%. The peak overshoots span 48.6 to 53.4% of the step, so removing the proportional setpoint response delays the rise without eliminating overshoot. Loop 2 departs from its target by up to 1.68 °C, illustrating transient interaction that the steady-state relative gain array alone does not describe. Figure 4.3 draws the table as bars.

4.7.2 Ambient Load Disturbance

Scenario S4 holds both setpoints fixed and steps the ambient temperature from 21 °C to 26 °C, which is a load change on both loops at once. Results are given in Table 4.6.

The three columns coincide to the last stored digit under this common sampling protocol. Their final heater position of 28.78% approaches the 28.60% steady requirement at $T_a = 26\,°C$ with $T_1 = 45\,°C$ and $T_2 = 40\,°C$. Residual transient error remains at 890 s. Types A, B and C have identical increments at a fixed setpoint and with matching initialized histories. Differences caused by changing type during a run or by different timer histories are initialization or sampling effects, rather than different load rejection laws.

Pause. Try Exercise 4.3 before reading on.

4.7.3 Derivative Action Against Sensor Noise

Scenario S5 holds the default setpoints for 600 s after the 1800 s warm-up. Noise-free heater travel is 0.24% on loop 1 and 0.14% on loop 2. Four declared pseudorandom streams at the default $\sigma_v = 0.2\,°C$

Table 4.7 Scenario S3 Under PID and Model Predictive Control on the Twin

Metric	Type A	Type B	Type C	MPC
IAE T_1, °C s	1563.8	1570.0	1637.9	384.6
IAE T_2, °C s	520.9	522.2	494.8	13.8
Peak T_1, °C	60.32	60.34	59.86	55.08
Overshoot, % of the step	53.2	53.4	48.6	0.8
First reaches SP_1, s	85	85	106	91
Within 0.5 °C for good, s	769	770	671	74
Q_1 travel, %	189.4	172.0	157.5	115.6
Q_2 travel, %	54.9	55.0	52.1	22.7
Peak T_2 deviation, °C	1.67	1.68	1.57	0.20

Noise off, matched model. Under MPC, 2690 plans, none rejected, 2.50 solver iterations on average and 26 at most. Windows end at 2690 s and times count from 1800 s, one sample earlier than in Table 4.5, so Type B reads 172.0 rather than 172.1% of travel and 85 rather than 84 s, while IAE agrees at the precision of Table 4.5.

produce 484 to 525% travel on loop 1 and 485 to 526% on loop 2. The standard deviation of loop 1's recorded error is 0.200 to 0.205 °C, against 0.005 °C in the noise-free reference. Exact noisy histories vary between visits because the browser application uses unseeded randomness. The current CSV records the same noisy value that the controller acted on. Equations (4.8) and (4.10) account for the size of it. With derivative and error taken as in Type B, the noise part of the per-cycle increment is $-(K_p + K_d/\Delta t)\, v_k + (K_p + 2K_d/\Delta t)\, v_{k-1} - (K_d/\Delta t)\, v_{k-2}$, whose standard deviation at the default gains and $\Delta t = 1$ s is $\sigma_v\sqrt{26} = 1.02\,\%$. The expected movement is $\sqrt{2/\pi}$ times that standard deviation, or 0.81 % per cycle and 488 % over 600 cycles, against the measured 484 to 525%. Setting K_d to zero leaves $\sigma_v\sqrt{8} = 0.57\,\%$ and a predicted 271 %, so the derivative term supplies roughly 45% of the noise-driven travel and the proportional term supplies most of the remainder. The four noise streams with $K_d = 0$ give 272 to 296% travel on loop 1. This confirms the direction and approximate size of the derivative contribution. The application implements no derivative filter, which is the usual industrial remedy [14], so lowering K_d is an available tuning handle. The nominal control interval is fixed by the application, so changing it requires a controller implementation change. The omission is deliberate, because it makes the cost of an unfiltered derivative measurable rather than asserted.

4.7.4 Operating Scenario in the Application

Figure 4.4 shows the current built application during the noise-free Type B step. The default targets are held for 1800 s, then loop 1's target rises from 45 to 55 °C. The capture is taken 300 s after that change. The twin and a connected board share tuning and targets while retaining separate controller histories. Exported samples permit direct comparison of their measurements and applied heater commands.

4.7.5 Comparison With Model Predictive Control

The predictive controller was run against the PID arrangements on the twin under the reference protocol above, with the same program, duration and noise for both, PID at the default gains, the plan at its fixed settings and no retuning for any case. With the default parameters the twin is the controller's own model apart from its Euler step, so every number here is a matched-model best case. A physical board will differ from that model in its gains, its losses and, above about 55 °C, its radiation, and no measurement of the predictive controller on a board is reported. Table 4.7 gives scenario S3.

The plan reaches the new target with 0.8% overshoot against 53.4% for Type B and cuts the loop 1 IAE from 1570.0 to 384.6 °C s with less travel, 115.6 against 172.0% on Q_1. It holds Q_1 at 100% from 1808 to 1829 s and takes it off well before T_1 arrives, because it carries the 140 s thermistor lag that the PID loops meet only as error. The larger difference is on loop 2, as shown in Figure 4.5.

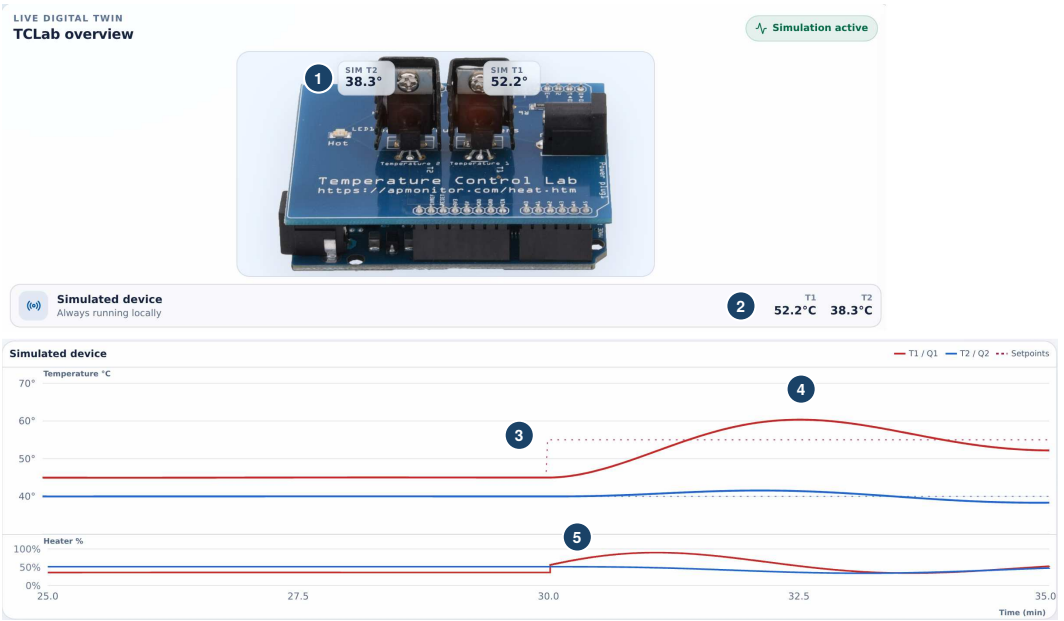


Figure 4.4 Current TCLab Control Studio during the noise-free Type B setpoint step, cropped and composited from one capture: the live twin panel above its trends. Loop 1's target changes from 45 to 55 °C after 1800 s at the default targets, and loop 2 stays at 40 °C. The capture follows 300 s later. Callouts: (1) the board with the twin's readings on its two heaters, (2) the readout row, (3) the loop 1 target step, (4) the overshoot of T_1, (5) the heater trends, with heater 1 stepping up at the target change. The trends draw the response without printing its values. The reference Type B experiment reaches 60.34 °C at its peak and moves heater 1 by 21.0% on the first execution after the target change (Table 4.5).

The plan has lowered Q_2 by 1% at 1802 s, before T_2 moves, because its model carries the conduction from heater 1. Under PID, loop 2 cannot act until that heat reaches its own thermistor, so T_2 is 0.1 °C off at 1829 s and Q_2 is first 1% below its value before the step at 1844 s. The largest T_2 deviation is 1.68 °C at 2093 s under PID against 0.20 °C at 1876 s under the plan, and the loop 2 IAE falls from 522.2 to 13.8 °C s, 2.6% of the PID value, with 22.7 against 55.0% of Q_2 travel. On the reverse step, SP_2 from 40 to 45 °C, T_1 moves 0.42 °C under PID and 0.13 °C under the plan.

Table 4.8 collects the other cases. On the ambient step of S4, which the model keeps at 21 °C, the plan wins by less, because a bias filtered at $\alpha = 0.1$ learns a load only as fast as that filter, and its final Q_1 of 28.60% is the steady requirement. Under the default noise, with four seeded streams of the comparison's own generator rather than those of S5, the plan moved Q_1 28 to 32% in 600 s against 485 to 551% under PID, while the standard deviation of the loop 1 error stayed between 0.196 and 0.213 °C under both. Noise reaches the plan only through the bias filter and the move weight, whereas the unfiltered PID terms act on every sample. With the twin's parameters changed and the model kept, every run under the plan still ends on both setpoints but its lead shrinks. A coupling time of 20 s, within the range of scenario 12, raises its loop 1 IAE from 384.6 to 652.3 °C s, and T_2 then peaks 2.21 °C off against 4.58 °C under PID.

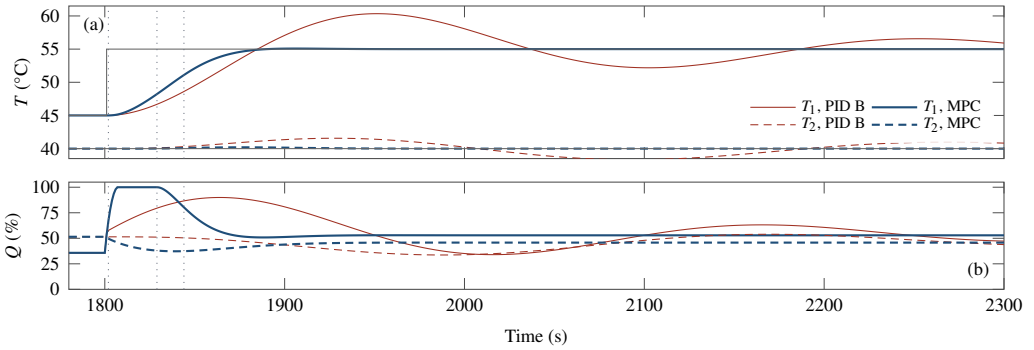


Figure 4.5 Scenario S3 on the twin under PID Type B and under the predictive controller, noise off, with SP_1 stepped from 45 to 55 °C after 1800 s. Thin lines are PID Type B and thick lines the predictive controller. (a) Reported T_1 (solid) and T_2 (dashed) with the setpoints as thin gray lines. (b) Heater duty cycles Q_1 (solid) and Q_2 (dashed). The dotted verticals mark 1802 s, when the plan has lowered Q_2 by 1%, 1829 s, when T_2 under PID is 0.1 °C off its setpoint, and 1844 s, when PID has lowered Q_2 by 1%.

Table 4.8 Further Cases on the Twin, PID Type B Against MPC

Case	Measure	PID B	MPC
S4, T_a 21 to 26 °C	IAE T_1, T_2	273.0, 296.2	202.9, 260.2
	travel Q_1, Q_2, %	29.2, 30.5	7.7, 14.3
SP_2 40 to 45 °C	IAE T_1, T_2	130.6, 814.0	9.1, 245.5
$\sigma_v = 0.2$ °C	travel Q_1, %	485–551	28–32
	travel Q_2, %	475–518	29–37
S3, $\tau_{HH} = 20$ s	IAE T_1, T_2	1301.5, 938.8	652.3, 328.7
S3, $P_2 = 200$	IAE T_1, T_2	1432.7, 558.7	387.4, 36.8
S3, $P_1 = 150$, $\tau_S = 180$ s	IAE T_1, T_2	1485.3, 551.7	760.8, 16.1
SP_1 to 68 °C	peak T_1, IAE T_1	71.45, 2319.2	68.04, 1664.8
SP_1 to 70 °C	peak T_1, IAE T_1	72.94, 2514.9	69.00, 2698.5

IAE in °C s over the 890 s after the change, noise off except the σ_v rows (600 s, four seeds, a range). The last three S3 rows change the twin's parameters while the controller keeps its model. The twin has no latch, so the 70 °C PID row exists only on the twin.

+ DID YOU KNOW? A Quieter Heater Under the Plan

From the simulator: at the default sensor noise of 0.2 °C, PID Type B moved heater 1 by 485 to 551% in 600 s, while the predictive controller moved it by 28 to 32%, with about the same spread of the loop 1 error under both (Table 4.8).

The cutoff case needs its context. With SP_1 at 70 °C, the highest setpoint the default cutoff admits, PID Type B peaks at 72.94 °C and first reaches 70 °C 171 s after the step, while the plan holds 69.00 °C, its ceiling, with zero slack. The plan's larger IAE, 2698.5 against 2514.9 °C s, exists only because the twin has no latch. On a board with the same cutoff the PID reading at 171 s would trip the latch and end the run in manual at zero duty, leaving no IAE to compare, so the difference is a property of the twin and not a result for the device. With SP_1 at 68 °C the plan peaks at 68.04 °C against 71.45 °C and has the smaller IAE, while PID passes 70 °C 174 s after the step.

Table 4.9 Estimators on Changed Twins, IAE of T_1 and T_2

Law	matched	fan	room +5	$P_1 = 150$
PID B	1612/538	1591/824	1871/829	1631/571
MPC, bias	430/41	1442/721	600/287	907/86
MPC, MHE	469/76	859/157	471/111	660/96
frozen, one offset	437/48	2200/1721	469/117	3054/2807
frozen, two offsets	445/57	1321/267	487/138	745/61
RMS error, bias/MHE	.046/.055	.170/.062	.077/.057	.086/.057

IAE in °C s over 601 to 2400 s, $\sigma_v = 0.2$ °C, mean of three seeds. The frozen rows plan with the MHE's arithmetic and the parameters held at their declared values. RMS error is the estimated against the true thermistor temperature, in °C. No plan or window was rejected.

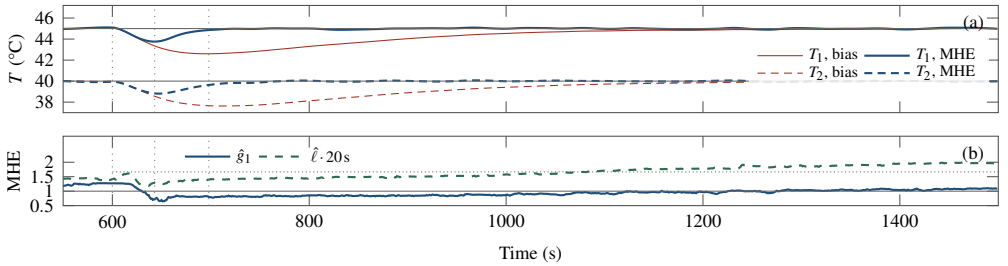


Figure 4.6 Fan case on the twin under the predictive controller with each estimator, first seed, $\sigma_v = 0.2$ °C, with τ_{HA} lowered from 20 to 12 s at 600 s. Thin lines are the output bias and thick lines the MHE. (a) Noise-free T_1 (solid) and T_2 (dashed) with the setpoints as thin gray lines. (b) The MHE's $\hat{g}_1$ (solid) and $\hat{\ell}$ scaled by the declared 20 s (dashed), whose true values are 1 (thin gray) and 1.67 (thin gray dotted). The dotted verticals mark the change and the largest T_1 sag under the MHE (643 s) and under the bias (698 s).

4.7.6 Estimator and Relay Comparisons

The estimators are compared on the twin from the 45 and 40 °C steady state, with a plant change at 600 s, SP_1 stepped to 55 °C at 1500 s and the end at 2400 s. A fan shortens τ_{HA} from 20 to 12 s, the room warms by 5 °C or P_1 falls from 200 to 150, while every controller keeps its declared model, and a fourth case leaves the twin matched. Entries are means over three seeded streams at the default noise, with IAE from 601 to 2400 s on the noise-free thermistor temperatures, which only the measuring script reads. Two frozen rows ask what the parameters earn, running the same window and plan with the parameters at their declared values and either the room offset b or one offset per heater (the offset-free form for two outputs) left free, as listed in Table 4.9.

The parameters earn the fan case, where the MHE cuts the bias path's IAE from 1442 and 721 to 859 and 157 °C s and the frozen forms reach only 2200 and 1721 or 1321 and 267. Most of that gain is in rejecting the change, as shown in Figure 4.6. In the first seed the largest T_1 sag is 1.24 °C at 643 s under the MHE against 2.40 °C at 698 s under the bias, and the T_1 IAE to 1500 s is 103 against 646, while after the step it is 764 against 808 and the T_2 IAE 63 against 37. The loss is taken up first as a weaker heater ($\hat{g}_1$ reaches 0.65 at 651 s), then as a larger $\hat{\ell}$. On the warmer room the frozen form with one offset matches the MHE, 469 and 117 against 471 and 111, so that gain belongs to estimating the room as a room. On the matched twin the wander of the parameters costs 9% on T_1 and 85% on T_2, while the frozen form stays near the bias at 437 and 48. With P_1 lowered the parameters earn T_1, 660 against 745, and cost T_2, 96 against 61.

Under the default noise the estimator does not identify the gain and the loss apart. One operating point fixes each heater's steady effect on its own temperature, which depends on g and ℓ together,

and the posterior correlation of $\hat{g}_1$ with $\hat{\ell}$ is 0.82 to 0.86. On the matched twin with the first seed, $\hat{g}_1$ ends at 1.000, 1.046, 1.147 and 1.265 for σ_v of 0, 0.05, 0.1 and 0.2 °C, and at 0.2 °C it rests on its bound of 1.275 in 14% of the seconds. Every seed ends between 1.25 and 1.27, a bias whose cause is not established, and the wrong pair gives T_1 a steady gain from Q_1 of 0.56 to 0.57 °C/% against 0.599, but T_2 0.070 to 0.073 against 0.100 and a heater time constant of 14.3 to 14.6 s against 20, so T_1 benefits and T_2 pays. With the noise off the two estimators tie on the matched twin at a T_1 IAE of 395, and the MHE wins the fan, room and heater cases at 794, 425 and 593 against 1434, 578 and 895 °C s. Overshoot stayed at or below 0.17 °C under both estimators, and the bias filter holds no parameter, so no case shows a bound that it violates and the MHE respects. Over 13 cases near and away from the cutoff, the MHE path came at most 0.23 °C closer to it than the bias path, and both kept at least 1.00 °C, where PID Type B passed the cutoff in three.

Over 40 noise seeds every relay experiment converged, in 95 to 962 s, at K_u of 23.5 to 43.9 %/°C (median 30.0) and P_u of 60 to 84 s, and wrote K_p of 7.4 to 13.7. On S3 with the noise off, the gains from the noise-free experiment, $K_p = 11.87$ and $K_i = 0.0641$, give IAE of 1126 and 87 °C s against 1570 and 522 for the default tuning, and all 40 noisy gain sets also won the step on T_1 (650 to 989 °C s). The same gains lose the warm-up from ambient, a T_1 IAE over 1800 s of 5002 against 3689 °C s, as did 39 of the 40 others, because the smaller K_i approaches slowly. Under the default noise the written gains move the heaters 2.0 to 3.7 times as far as the default tuning.

4.8 Technology in Practice

The twin of Section 4.3 reads two thermistor states through white noise and moves two heaters the instant they are commanded. Figure 4.7 draws the two loops with the devices of the board.

What Measures It

Each channel reads a thermistor, a resistance element whose high resistance makes the wires to it negligible [12]. The ten-bit converter of the Assumptions turns that resistance into a reading, and Equation (4.5) stands for its noise. A plant loop would read a platinum resistance thermometer or a thermocouple, usually inside a thermowell. A thermowell adds a first-order lag through its own mass and heat capacity, and an air gap between element and well adds a second lag [12]. The thermistor mounted beside its transistor is that arrangement in miniature: the sensor lag τ_S of Equation (4.3) is seven times the heater's own τ_{HA} and sets the fitted time constant of Table 4.3. No range, accuracy or response time is given here, because none was found in an open source that names no maker.

> **TECHNOLOGY** **Sensor: A Thermistor**
>
> A thermistor is a small piece of metal oxide whose resistance rises or falls with temperature. It is highly sensitive and nonlinear, so it serves where high accuracy is unimportant and the range is narrow, as in many consumer devices and in building heating and cooling. Plants with wide temperature ranges use platinum resistance thermometers instead. Reading a resistance means passing a current through it, and that current heats the element and adds a positive error. Pulsing the current and sampling only while it flows reduces the heating, at the price of a slow sample rate, which most temperature measurements can tolerate [12].

What Moves It

Each heater is a transistor driven by pulse-width modulation, with the duty cycle setting the share of full power [12]. An industrial electric heater whose element can only be fully on or fully off is driven the same way, by a PID function block feeding a time-proportioned output, as in a handbook example for an electric oven [12]. The model's heater is linear in duty cycle and has no switching period, which Section 4.6 counts among the delays the twin leaves out.

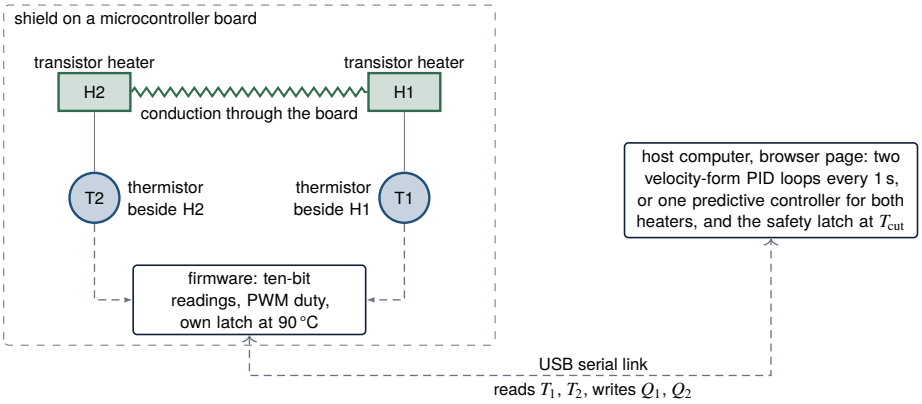


Figure 4.7 The two loops of the chapter on the board itself. Each thermistor sits beside its transistor heater, the firmware converts the readings and applies the duty cycles, and the control law runs on the host behind a serial link. Notice that heat from either heater reaches both thermistors through the board, so the two loops share one plant although each reads one sensor and moves one heater.

> **TECHNOLOGY** **Final Element: A Transistor Switched On and Off**
>
> Electronic power control switches a transistor fully on and fully off in rapid succession, a method called pulse-width modulation. The duty cycle, the share of each period spent on, sets the fraction of full power delivered, and a drive that accepts a control signal acts like any other final control element. In large motor drives the square-edged pulses radiate electrical noise, so power conductors are routed away from signal wiring [12]. On this board the transistor is itself the heater: the firmware turns the commanded percentage into a duty cycle, and the power the transistor dissipates warms the package that the thermistor reads.

Where the Controller Runs

The control law runs in a browser page on the host computer, not on the board. The firmware reads the thermistors, applies the duty cycles and keeps its own latch, and each cycle the host reads both temperatures, computes both commands and writes them over the serial link (Section 4.6). That sequence is the cycle time of a standards glossary: sensor signals read into memory, the algorithm executed and the actuator signals sent [24]. Plant loops run with cycle times from milliseconds to minutes [24], and the 1 s nominal control interval sits inside that range, short against the 141 s fitted time constant. In a plant the velocity-form PID of Equation (4.11) would be the PID function block of a single-loop controller or a distributed control system.

> **IN PRACTICE** **Judge Derivative Action on the Heater Trend**
>
> An instrument handbook sums up the three actions for the person tuning a loop. Proportional action reproduces process noise on the output, derivative action amplifies it dramatically, and integral action is able to ignore it. Derivative action helps most on processes dominated by large lags, where a small amount may allow more aggressive proportional and integral settings. The open-loop tuning rules overlook noise, and a small apparent dead time makes them predict gains too large to be practical. The handbook's advice is that no algorithmic tuning method replaces the informed judgment of the person tuning the loop [12].

What Is Modernizing It

Three developments reach this loop. The first is the learned model with physics built in: a neural network constrained to keep the rising response of temperature to heater power was trained on 250 samples from a two-heater kit, with heater inputs between 10 and 50 %, and then served a predictive

controller allowed up to 65 %, beyond its training range [25]. The second is the digital twin in the process industry, which joins sensor data to physics-based or data-based models. About 40 % of the 31 studies in one review combined the two kinds of model [26]. The third is the learning-based controller tested on a twin before a board, the subject of the open challenge at the end of Section 4.10.

4.9 Limitations

The loss to the surroundings is linear, so the twin will not reproduce the falling steady-state gain that a physical board shows above roughly 55 °C, and a parameter set fitted at 40 °C will overpredict at 65 °C.

The twin is not calibrated to any particular device. Board-to-board variation in TCLab hardware is large enough that benchmark comparisons are recommended on one device under similar ambient conditions [1], and the application's estimator does not separate heater gain from heat loss under the default noise (Section 4.7.6), so matching a specific board remains manual work. Chapter 11 fits nested energy balances, one of them with a radiative term, to recorded TCLab runs.

The physical loop reads its temperatures at the start of the same cycle in which it acts, so it carries no extra sample of dead time, but the serial round trips and the firmware pulse-width modulation period separate the measurement from the applied power in a way the simulated loop does not reproduce.

The PID controller has no derivative filter, no bias term, no feedforward path and no rate limit on the heater command, so the only PID structures available are the three velocity forms, beside manual operation and the one predictive controller.

The predictive controller's model is the twin's balance at the declared defaults, so it leaves out radiation and the fall in gain above about 55 °C, every parameter an operator sets and every difference between a particular board and the defaults. With the default estimator its output bias absorbs a constant difference slowly, at $\alpha = 0.1$, and nothing in it corrects a wrong gain. Its horizon and weights are fixed, and so is its model unless the MHE is chosen. Every result for it here is from the twin, matched or deliberately changed, none is from a board, and its soft ceiling keeps a plan below the cutoff only when the model is right, so on a board the latch remains the protection.

No board comparison of the estimator or the relay was run. Under the default noise the estimator does not separate heater gain from loss, its pair underrates the coupling, and τ_{HH} and τ_S stay declared. The relay's band sets its K_u as much as the loop does.

The safety latch is armed by the physical measurements alone, so the twin and the device do not execute the same logic once the cutoff is reached. That asymmetry is intended, because the latch protects a transistor and the twin has none, but it is a caveat on the claim that one set of code drives both plants. A simulation-only session can be driven to 80.9 °C, and no browser mechanism is a substitute for the firmware's own command timeout or for watching the board.

Web Serial is available only in Chromium-family browsers on desktop operating systems and only from a secure context, so the hardware half of the application is unavailable on iOS and on pages served over plain HTTP. The simulator remains fully functional in that case, which is the reason the two halves were built to run independently.

4.10 Discussion

Temperature loops of exactly this shape, a slow first-order plant behind a lagged sensor with a saturating actuator, regulate reboilers, extruder barrels, bioreactor jackets and battery packs, and the steam-heated exchanger of Volume 2, Chapter 4 is tuned from step tests in the same way. A two-transistor board is worth studying because the structure transfers while the cost does not. A department can issue several hundred TCLab kits for the price of one pilot-scale rig, and each student then owns the apparatus for the semester rather than sharing it for three hours. Reported experience with the device in online and remote sections indicates that this arrangement survives the loss of a physical laboratory [2].

Industrial digital twins are justified by the tests that are too expensive or too dangerous to run on the plant. The same argument holds at the scale of a classroom board, where running the loop unstable, forcing the actuator onto its limit and tripping the protection are all instructive and all cheaper in simulation. Placing the twin beside the device rather than in front of it makes the comparison the object of study, which is the habit that transfers to a plant where the model is the only thing available before startup.

OPEN CHALLENGE Trusting a Learned Controller With Real Heaters

In one study, controllers built on large language models set the power of two heaters on kits of this kind while a fan cooled one. Trials on the physical boards met varying initial conditions, varying response times and poor reproducibility under changing room conditions, so the study built a digital twin for repeatable tests. Use in a plant would need deterministic fallback policies, formal verification of the form of each output and the standards IEC 61511 and NIST SP 800 for functional safety and cybersecurity [27]. Progress would be a learned controller whose behavior on the board is bounded before it is given the heaters.

4.11 Conclusion

This chapter studied an application that runs a four-state Temperature Control Lab simulation and a physical TCLab from one set of controller code, with the two responses plotted on a common time axis. The twin integrates two heater temperatures and two thermistor temperatures with explicit Euler steps of 0.1 s, and its steady-state gain matrix, measured from the running application as 0.599 and 0.050 °C/% on the first row and 0.100 and 0.300 °C/% on the second, agrees with the closed-form solution of the energy balances. A two-point fit of the same response returns $\tau_p = 141$ s against $\theta_p = 18$ s, so the identified time constant belongs to the thermistor rather than to the heater.

Three velocity forms of PID are implemented and measured, and a 10 °C setpoint step separates them by a first-cycle heater movement of 31.0%, 21.0% and 1.0%, while an ambient load step of 5 °C leaves them indistinguishable because the three difference equations coincide when the setpoint is constant. Sensor noise of 0.2 °C gives 484 to 525% heater travel over 600 cycles after settling, compared with 0.24% without noise, and the derivative term accounts for roughly 45% of that travel, which is the case for the derivative filter that the application deliberately omits.

A linear predictive controller on the twin's own balance, offered beside the PID loops, holds loop 2 within 0.20 °C of its setpoint on the 10 °C step where Type B departs 1.68 °C, a matched-model result on the twin with no device measurement behind it. On a moving horizon estimate it cuts the T_1 IAE after a faster heat loss from 1442 to 859 °C s, with gain and loss unidentified under noise, and relay-tuned PI gains win the step and lose the warm-up.

A fuller study would extend the moving horizon estimator to the coupling and sensor time constants, and add a filtered derivative, feedforward from the second heater and a comparison across several physical boards. What the application offers throughout is the same controller on a model and on the device that model was built for.

KEY POINTS

- The board answers a heater step slowly because each sensor lags the heater beside it.
- A two-point fit returns $\tau_p = 141$ s and $\theta_p = 18$ s, so the time constant belongs to the thermistor.
- The three velocity forms of PID differ only while the setpoint moves: a 10 °C step moves the heater 31.0, 21.0 and 1.0% on the first cycle, and an ambient step leaves them identical.
- Sensor noise of 0.2 °C drives 484 to 525% of heater travel over 600 cycles, and the unfiltered derivative supplies roughly 45% of it.

Where else this applies. Volume 14, Chapter 5 couples one hundred heater zones on one plate through the same conduction, and Volume 2, Chapter 4 tunes a plant thermal loop from step tests.

References

[1] J. Park, R. A. Martin, J. D. Kelly, and J. D. Hedengren, "Benchmark temperature microcontroller for process dynamics and control," *Computers & Chemical Engineering*, vol. 135, p. 106736, 2020.

[2] B. Yu, "Teach online controls laboratory using a low-cost temperature control lab hardware," in *2022 ASEE Annual Conference & Exposition Proceedings*. ASEE Conferences, 2022.

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

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

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

[6] J. Ma and J. V. Nickerson, "Hands-on, simulated, and remote laboratories: A comparative literature review," *ACM Computing Surveys*, vol. 38, no. 3, p. 7, 2006.

[7] R. Heradio, L. de la Torre, D. Galan, F. J. Cabrerizo, E. Herrera-Viedma, and S. Dormido, "Virtual and remote labs in education: A bibliometric analysis," *Computers & Education*, vol. 98, pp. 14–38, 2016.

[8] S. Dormido Bencomo, "Control learning: present and future," *Annual Reviews in Control*, vol. 28, no. 1, pp. 115–136, 2004.

[9] B. Taylor, P. Eastwood, and B. L. Jones, "Development of a low-cost, portable hardware platform to support hands-on learning in the teaching of control and systems theory," *Engineering Education*, vol. 9, no. 1, pp. 62–73, 2014.

[10] L. D. Feisel and A. J. Rosa, "The role of the laboratory in undergraduate engineering education," *Journal of Engineering Education*, vol. 94, no. 1, pp. 121–130, 2005.

[11] J. A. Rossiter, B. Pasik-Duncan, S. Dormido, L. Vlacic, B. Jones, and R. Murray, "A survey of good practice in control education," *European Journal of Engineering Education*, vol. 43, no. 6, pp. 801–823, 2018.

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

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

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

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

[16] C. C. Hang, K. J. Åström, and W. K. Ho, "Refinements of the Ziegler-Nichols tuning formula," *IEE Proceedings D Control Theory and Applications*, vol. 138, no. 2, pp. 111–118, 1991.

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

[18] 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.

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

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

[21] D. 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.

[22] 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.

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

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

[25] N. T. Nguyen and J. C. Tique, "Physics-constrained Taylor neural networks for learning and control of dynamical systems," arXiv preprint 2410.02258, 2024.

[26] M. Mayr, G. C. Chasparis, and J. Küng, "Learning paradigms and modelling methodologies for digital twins in process industry," arXiv preprint 2407.02275, 2024.

[27] J. Vyas and M. Mercangoz, "Autonomous control leveraging LLMs: An agentic framework for next-generation industrial automation," arXiv preprint 2507.07115, 2025.

4.12 Exercises

4.1 Concept check · 2 min

A two-point fit of the twin's response of T_1 to a step in Q_1 returns $\tau_p = 141\,\mathrm{s}$ and $\theta_p = 18\,\mathrm{s}$. Which element of the twin does the fitted time constant mostly describe?

(a) The heater-to-ambient loss, since it sets how fast heater 1 warms

(b) The conduction between the heaters, since it is the longest path on the board

(c) The thermistor lag, since the measurement is the slowest state in the loop

(d) The control interval, since the fit reads one sample per second

4.2 Concept check · 2 min

Which of these signals of the twin is a state that no instrument measures, on the twin or on a physical board?

(a) The reported temperature T_1 that loop 1 controls

(b) The heater duty cycle Q_1 that loop 1 manipulates

(c) The ambient temperature T_a that sets the reset condition

(d) The heater temperature T_{H1} that drives the thermistor

4.3 Concept check · 2 min

Scenario S4 steps the ambient temperature from 21 to 26 °C with both setpoints fixed. Table 4.6 gives identical results under PID Types A, B and C. Why?

(a) With a constant setpoint the error increments are the negative measurement increments, so the three laws coincide

(b) The derivative term is zero during a load change, so only the proportional and integral terms act

(c) The anti-windup correction absorbs the difference between the three forms at every cycle

(d) A change of ambient reaches the controller only through the integral term, which all three share

4.4 Theory and calculation · 7 min

Take the twin at its defaults: $P_1 = 200$, $P_2 = 100$, $C_H = 5720\,\%\,\mathrm{s}/^\circ\mathrm{C}$, $\tau_{HA} = 20\,\mathrm{s}$ and $\tau_{HH} = 100\,\mathrm{s}$. Write $a = 1/\tau_{HA}$ and $b = 1/\tau_{HH}$.

(a) Set the derivatives of Equations (4.1) and (4.2) to zero and solve for the four entries of the steady-state gain matrix K. *Checkpoint:* $K_{11} = 0.599$, $K_{12} = 0.0500$, $K_{21} = 0.0999$ and $K_{22} = 0.300$, all in °C/%.

(b) Compute the relative gain λ_{11} and say which pairing it recommends. *Checkpoint:* $\lambda_{11} = 1.029$, the diagonal pairing.

(c) Find the duty cycles that hold $T_1 = 45\,^\circ\mathrm{C}$ and $T_2 = 40\,^\circ\mathrm{C}$ at $T_a = 21\,^\circ\mathrm{C}$, and again at $T_a = 26\,^\circ\mathrm{C}$. *Checkpoint:* $Q_1 = 35.75$ and $Q_2 = 51.48\,\%$, then $Q_1 = 28.60$ and $Q_2 = 37.18\,\%$.

4.5 Theory and calculation · 6 min

The step tests of Table 4.3 start at $t = 0$ with the noise off.

(a) Find the two time constants of the heater subsystem alone, from the state matrix of Equations (4.1) and (4.2) with $a = 0.05/\mathrm{s}$ and $b = 0.01/\mathrm{s}$. *Checkpoint:* 20 s and 14.3 s.

(b) Apply the two-point rule of Equation (1.18) to the direct response ($t_{28} = 65\,\mathrm{s}$, $t_{63} = 159\,\mathrm{s}$) and to the cross response ($t_{28} = 82\,\mathrm{s}$, $t_{63} = 177\,\mathrm{s}$), and give θ_p/τ_p for the direct one. *Checkpoint:* $\tau_p = 141\,\mathrm{s}$, $\theta_p = 18\,\mathrm{s}$ direct, $\tau_p = 142.5\,\mathrm{s}$, $\theta_p = 34.5\,\mathrm{s}$ across, ratio 0.13.

Explain where the apparent dead time comes from, given that the model has no transport delay.

4.6 Theory and calculation · 7 min

Loop 1 runs Type B at the default gains $K_p = 2$, $K_i = 0.1$ and $K_d = 1$ with $\Delta t = 1\,\mathrm{s}$, held at a fixed setpoint. The reported temperature carries white Gaussian noise v_k with $\sigma_v = 0.2\,^\circ\mathrm{C}$.

(a) From Equations (4.8) and (4.10), write the noise part of the per-cycle heater increment as a combination of v_k, v_{k-1} and v_{k-2}, and find its standard deviation. *Checkpoint:* Coefficients -3, $+4$ and -1, standard deviation 1.02 %.

(b) The mean absolute value of a zero-mean Gaussian variable is $\sqrt{2/\pi}$ times its standard deviation. Predict the heater travel over 600 cycles. *Checkpoint:* 0.81 % per cycle, about 488 %.

(c) Repeat with $K_d = 0$, and find the share of the travel that the derivative term supplies. *Checkpoint:* 0.57 % per cycle standard deviation, 271 %, a share of about 45 %.

4.7 App challenge · 7 min

Use scenario 1, Watch the lag. In Settings set the Sensor noise to 0 and apply it, press Reset, and set the Sim speed to 10×. Keep the app's tab in front during the run. In Manual, type 50 in the Q_1 field with Q_2 at 0. Let the session clock run about 20 min (two minutes of real time) and export the CSV file. Its time column is in minutes. Take the step time as the first row in which `sim_q1_pct` reads 50.00.

(a) Read `sim_t1_c` and `sim_t2_c` 1140 s after the step, and compute the gains of T_1 and T_2 to Q_1. *Checkpoint:* 50.96 °C and 25.99 °C, each within 0.03 °C. Gains 0.599 and 0.100 °C/%.

(b) Using the final rise of $0.599 \times 50 = 29.97\,^\circ\mathrm{C}$, find the times after the step at which T_1 first passes 28.3 and 63.2% of it, and fit τ_p and θ_p. *Checkpoint:* $t_{28} = 65\,\mathrm{s}$ and $t_{63} = 159\,\mathrm{s}$, each within 3 s. $\tau_p = 141\,\mathrm{s}$, $\theta_p = 18\,\mathrm{s}$.

Say why T_2 rises at all, and why its rise is so much smaller.

4.8 App challenge · 9 min

Use scenario 6, Compare the PID types, with the default gains, Type B and setpoints of 45 and 40 °C. In Settings set the Sensor noise to 0, press Reset, set the Sim speed to 10× and select PID auto. Run until the session clock passes 30 min. Then raise the loop 1 setpoint to 55 °C (type 55 in the field and press Enter, which commits it), run 15 min more and export the CSV file. The step row is the first row in which `setpoint_t1_c` reads 55.00.

(a) Find the change in `sim_q1_pct` from the row before the step row to the step row, and predict from Equation (4.13) what Types A and C would give. *Checkpoint:* 21.0 %, within 0.3 %. Type A 31.0 % and Type C 1.0 %.

(b) Find the peak of T_1, its time after the step row, and the overshoot as a percentage of the step. *Checkpoint:* 60.34 °C within 0.05 °C, about 150 s after the step, 53.4 %.

(c) Find the largest departure of T_2 from 40 °C after the step, and when it occurs. *Checkpoint:* 1.68 °C within 0.03 °C, about 290 s after the step.

Reconcile (c) with $\lambda_{11} = 1.029$.

4.9 App challenge · 9 min

Use scenario 13, PID against MPC. Repeat the run of the previous exercise with the mode switch at MPC instead of PID auto and the Estimator at its default, Bias: noise 0, Reset, 10×, 30 min, then the loop 1 setpoint from 45 to 55 °C with Enter, 15 min more, and export the CSV file. The step row is the first row in which `setpoint_t1_c` reads 55.00.

(a) Find the peak of T_1 and the overshoot. *Checkpoint:* 55.08 °C within 0.05 °C, 0.8 %.

(b) Find the largest departure of T_2 from 40 °C after the step. *Checkpoint:* 0.20 °C within 0.03 °C, about 75 s after the step.

(c) Find how many seconds after the step row `sim_q2_pct` first stands 1 % below its value in the row before the step. *Checkpoint:* About 1 s, against about 43 s under PID Type B.

Explain why the plan lowers Q_2 before T_2 moves, and why this run says little about a physical board.

4.10 Application area · 3 min

An engineer has fitted the twin's parameters to a board at about 40 °C and now wants to hold T_1 at 65 °C with Q_2 off.

(a) What steady Q_1 does the default twin predict for 65 °C at $T_a = 21\,^{\circ}\mathrm{C}$? *Checkpoint:* 73.4 %.

(b) Will the board, held at that duty, settle above or below 65 °C, and how could the bend be located? *Checkpoint:* Below.

4.11 Application area · 3 min

The predictive controller keeps both predicted temperatures under a soft ceiling 1 °C below the hardware cutoff, and the latch trips on any single board reading at or above the cutoff.

(a) With Gaussian noise of $\sigma_v = 0.2\,^{\circ}\mathrm{C}$, about what fraction of readings reach the cutoff when the true temperature sits on the cutoff, and when it sits 1 °C below it? *Checkpoint:* One half, and about 3×10^{-7}.

(b) Why may the alarm be acknowledged only once both board temperatures are below the cutoff, and why does the twin have no latch? *Checkpoint:* The hazard must have passed. The twin has no transistor.

4.12 Review the agent · 5 min

An AI assistant reads a relay experiment on loop 1 of the twin, run with the noise off from the 45 and 40 °C steady state, which returned $K_u = 37.97\,\%/^{\circ}\mathrm{C}$ and $P_u = 84.1\,\mathrm{s}$. It writes: "The ultimate gain of loop 1 is 38 %/°C, so a proportional controller above that gain will oscillate without end. The default $K_p = 2$ has a gain margin of 19, and the Tyreus-Luyben gains $K_p = 11.87$ and $K_i = 0.0641$ use a third of the available margin." Find the flaw and say how it could be checked.