

Investigation of Temperature Control Strategies

Cameron Jeffery

Fraser Lee

McGill University Department of Physics

Supervisor: Dominic Ryan March 18, 2025

Abstract

This paper explores and compares a variety of strategies to control the temperature of a mass of aluminum. We investigate three broad families of strategies: open-loop control, on/off control with hysteresis, and proportional-integral-derivative (PID) control - with sub-variants and various parameters tested for within each family. For every control method, an Arduino microcontroller was used to monitor the temperature of an aluminum block, and to switch on and off a heating element, with logic implementing the control strategy. A python codebase was used to analyse the results of each strategy, to search the space of possible strategies, and to program the Arduino with future strategies to be tested. The open-loop approach, while maximally uncomplicated, lacked any active response to the system's state, and thus was only able to gradually converge to a temperature determined by the system's conditions. The on/off controller was a vast improvement on this strategy, as it actively converged on a target temperature - but suffered from overshoot, oscillation and poor control after the target temperature was reached. On/off hysteresis proved to be useless in controlling this particular system, as the lag time between changing input power and detecting a change in temperature - as thermal energy propagated between heater and thermometer - was already long enough to prevent fast oscillations. After a long exploration of the parameter space, our optimised PID controller provided smooth and stable control - quickly converging to the target temperature ($T_{\text{rise } 70^{\circ}\text{C} \rightarrow 115^{\circ}\text{C}} = 119.17\text{s}$) without overshoot ($T_{\text{max}} - 115^{\circ}\text{C} = 0.20^{\circ}\text{C}$), and maintaining it with minimal oscillation ($\text{avg}_{\text{period when settled}}(T - 115^{\circ}\text{C}) = 0.16^{\circ}$). Our findings reaffirm the superiority of PID control if precise, timely, and stable control is required for a system.

Contents

1	Introduction	1
2	Experimental Methods	3
3	Results	6

4 Discussion **11**

5 Conclusion **12**

1 Introduction

This experiment intends to investigate the performance of a real-time computer temperature control systems. The ultimate aim is to compare different control algorithms that regulate the the temperature of an aluminum block using an Arduino-based heater system - a system which forms a full servo control loop that actively senses the system state and applies calculated responses to both achieve and maintain a desired steady temperature for the aluminum block. For this experiment, three categories of temperature control strategies were investigated: open loop operation, on/off control with optional hysteresis, and proportional-integral-derivative (*PID*) control.

In an open loop operation, the heater is applied with a fixed power such that the aluminum block comes to an equilibrium - with power loss to cooling equal to power input from the heater. This process involving no feedback from the thermometer, and the final equilibrium temperature depends directly on both the geometry of the setup and the power input level.

In an on/off controller with hysteresis setup, the heater power will periodically toggle between fully *off* or fully *on* depending on whether the temperature of the aluminum block is above or below the desired temperature. This system can be prone to fast oscillations [1], as the measured temperature transitions from marginally above to marginally below the threshold, prompting investigation of the addition of a hysteresis term. Expressed in degrees, a hysteresis of $h^{\circ}\text{C}$ will have the heater transition from off to on when $t < \text{threshold} - h$, and transition from on to off when $t > \text{threshold} + h$ - decreasing the maximum possible frequency of oscillations.

The most sophisticated approach investigated is PID control, which dynamically adjusts the heater power based on the magnitude of the error $e(t)$ (that is, the difference between the desired temperature and the actual temperature of the aluminum block), the integral of error over time, and the rate of change of the error. Expressed in closed-form, the PID controller is given by:

$$O(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

Where K_p , K_i , and K_d are the proportional, integral, and derivative gain factors, respectively [2]. The output parameter, $O(t)$, represents the control system's decision on how much power to input into the heater - in practice clamped to the range $[0, 1]$, and quantized to 1/60, representing the number of electrical grid cycles per second in which the heater is active (controlled using the Arduino's pulse-width modulation system at 1 Hz). In reality, this system is discretised, and updated per timestep as:

$$\begin{aligned} P_t &= K_p e(t) \\ I_t &\leftarrow I_{t-\Delta t} + K_i e(t) \Delta t \\ D_t &= K_d \frac{1}{\Delta t} (e(t) - e(t - \Delta t)) \\ O(t) &= \text{clamp-and-quantize}(K_t + I_t + D_t) \end{aligned} \tag{2}$$

When the other constants are set to zero, the proportional band ($PB = 1/K_p$) is the range of temperature error over which the control output scales linearly [3].

The integral control (second term in Eq. (1)) accumulates the error, $e(t)$ over time and adjusts the output to eliminate any residual offset between setpoint and actual temperature compared to an exclusively proportional controller. The strength of this effect is characterized by the integral action time (IAT), defined as the time it would take for the integral term alone to change the output from zero to full power, given a constant error equal to one proportional band.

In our system, most of the initial temperature rise happens in a regime where $e(t) \gg 1/K_p$, and the output heater power is limited by hardware. Due to the physical dynamics of the system, a rise from $30^\circ \rightarrow 115^\circ$ and from $70^\circ \rightarrow 115^\circ$ - promptly shutting off the power at exactly that temperature - would produce a nearly identical overshoot. These two initial configurations, however, would have a vastly different history, and a different I value at the point that they reach the target temperature. For this reason, we investigated PI controllers both allowing the I parameter to be active at all times, and resetting the accumulator $I_t \leftarrow 0$ when $P > 1$ or $P < 0$.

The derivative control (third term in Eq. (1)) is designed to react to the rate at which

the error is changing. When $e(t)$ is decreasing, K_d moves the shut-off point of the controller earlier in time compared to the counterfactual, counteracting overshoot. Similarly to IAT, the derivative action time (DAT) can be characterized by how strongly this rate-of-change term contributes to the control output. In particular, a temperature change rate of one proportional band per DAT would cause the derivative term alone to shift the output by exactly 1.

The tuning of these parameters allows the controller to be precisely tailored to maintaining the desired thermal conditions for the aluminum block, and this, therefore, should give insight into the robustness and efficacy of the PID control strategy in comparison to other aforementioned control schemes.

2 Experimental Methods

The experimental system consisted of a solid aluminum mass heated by an electric cartridge heater and monitored via a type K thermocouple. Control of the heating process was implemented using an Arduino, which acted as the central controller. The heater was powered by a 14 V AC transformer and gated through a solid-state relay (SSR), which was in turn controlled by a 1 Hz pulse-width modulation (PWM) signal from the Arduino.

Temperature measurements were gathered using a thermocouple amplifier breakout board. These readings were used to determine the deviation from the setpoint temperature and to adjust heater power accordingly in each of the feedback control modes. A Python-based script was used for real-time plotting of the temperature and PWM width as they varied with time, analysis of the quality of various control methods, scheduling and dynamic design of a sequence experiments, and programming the Arduino with instructions for the current control mode and constants to use for each experiment.

The cooling of the aluminum metal block between test runs was assisted by a 12 V DC fan, powered via a bench power supply and controlled through an Adafruit Motor Shield V2. The fan was not integrated into any feedback control loop, but was used solely to accelerate the return of the aluminum block to near-ambient conditions.

A schematic of the experimental setup is diagrammed in Fig. 1:

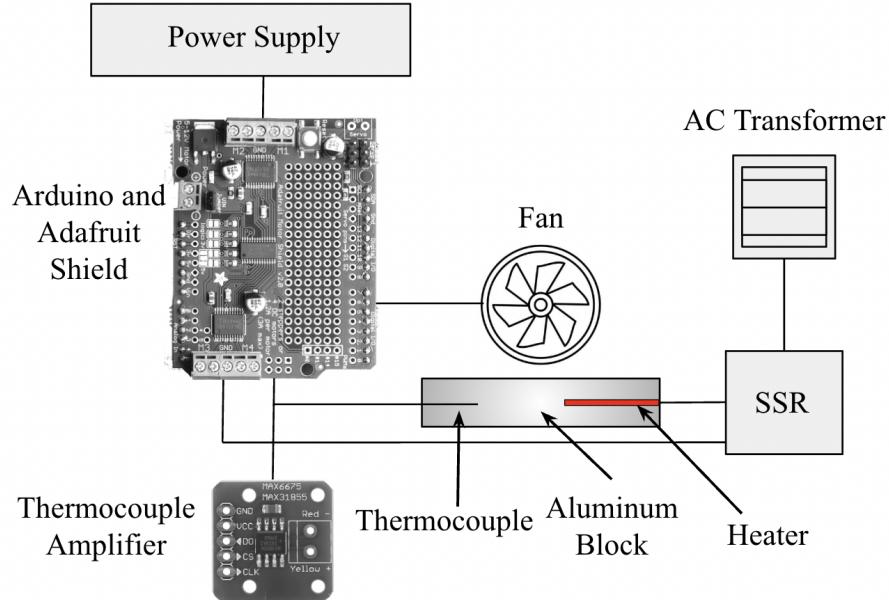


Figure 1: Schematic of the experimental setup for all trial runs. Note the fan was only used to cool the aluminum block in between each control trial.

Throughout the experiment, a safety cutoff was implemented in all code to prevent the system temperature from exceeding 150° C. Each controller was tested under identical initial conditions and setpoint values, typically in the range of 50 – 120° C.

Controllers were evaluated with several parameters:

- T_{rise} : the time between passing an initial threshold of 70°C to entering within $\pm 1^\circ\text{C}$ of the target temperature (for our tests, 115°C).
- T_{settle} : the time between first passing an initial threshold of 70°C, until the *last time in the run* when the temperature enters the tolerance band ($\pm 1^\circ\text{C}$ from the target temperature).
- ΔT_{over} : the overshoot of our target temperature

$$\Delta T_{\text{over}} = \max\{T(t)\} - T_{\text{target}}$$

- E_{ss} : the steady-state error, given by

$$E_{ss} = \frac{1}{N} \sum |T(t_i) - T_{\text{target}}|$$

where the sum is taken exclusively over the portion of data after the system has settled.

For initial tests (*various on/off hysteresis levels, proportional band widths (K_p values), and configurations of K_p and K_i*), our python code was programmed to execute a pre-determined series of experiments. However, for the full PID controller, the space of possible configurations over all three parameters was too large to determine near-optimal configurations by inspection. We instead programmed a bespoke 3-dimensional automatic search of the coordinate space, by:

1. Initially determining a wide range of plausible values for each of the three parameters
2. Testing the controller quality of the 8 “corners” formed by taking either the minimum or the maximum of the range for each of K_p , K_i and K_d .
3. Forming a “cell” covering the entire plausible value-space, and adding this cell to a priority-queue. We then iteratively:
 - (a) Removed the highest priority cell from this queue, and evaluated the controller formed by taking the K_p , K_i and K_d values at its center.
 - (b) By a linear interpolation of adjacent test results, found the axis along which test results were varying the greatest.
 - (c) Divided the cell in two along the plane orthogonal to this axis of greatest variation, and added both new sub-cells to the queue.

The queue was ordered by a combination of the size of cells, and the value of the controller at their center ¹. This allowed our search process to quickly hone in on high quality PID controllers, without completely neglecting any very large under-explored regions. This was

¹For a single-variable controller “value”, we took $v = T_{\text{settle}} + 200 \cdot E_{ss}$, as in our configuration this brought the two parameters into approximately the same order as each-other. The queue was sorted by $\left(\frac{v-v_{\min}}{v_{\min}} + 0.1\right) \cdot \frac{1}{\text{cell volume}}$ as I observed this to provide a good balance of exploration and optimization.

critical, as experiments could typically only be executed on an ~ 8 minute cadence, limiting the total number of data points we were able to collect.

In our initial tests of a PID controller, we derived D_t per time-step, under the assumption that any noise in the $e(t)$ signal would average out over time; an erroneously positive or negative D_{t-1} would necessarily be canceled out by D_t, D_{t+1}, \dots , with the integral of $O(t)$ giving just the true movement of the underlying $e(t)$. We neglected to realize that clamping $O(t)$ to the *zero-one* range, meant it wasn't necessarily the case that result would cancel. Fig 11 compares trials with and without “smoothing” D .

3 Results

The performance of four different temperature control strategies was evaluated by monitoring the temperature of a heating system of the aluminum block over time, along with the corresponding heater power output expressed as a pulse-width modulation (PWM) fraction, and computing the quality statistics described in Experimental Methods 2.

In the open loop configuration, Fig 2 (a), the heater was operated at a fixed PWM fraction of 0.25. With no feedback mechanism in place, the system temperature rose steadily and asymptotically approached approximately 67.0°C. Since no adjustment is made based on the actual temperature, this strategy is sensitive to environmental conditions, with a final temperature determined by input power and thermal losses.

The on/off control method, Fig. 2 (b) implements a simple feedback loop that turns the heater fully on when the temperature falls below a set lower limit and off when it exceeds an upper threshold. In the example figure above, the target temperature was set to 115.0°C, with a hysteresis band of 0.2°C. As shown in the plot, the heater PWM fraction oscillates between 0 and 1, with abrupt transitions that correspond to the heater toggling states. Although this method achieves regulation, the system never truly stabilizes at a fixed temperature, and the switching introduces thermal cycling.

Various hysteresis widths were tested for the *on/off* controller. The two extrema are displayed in Fig 5.

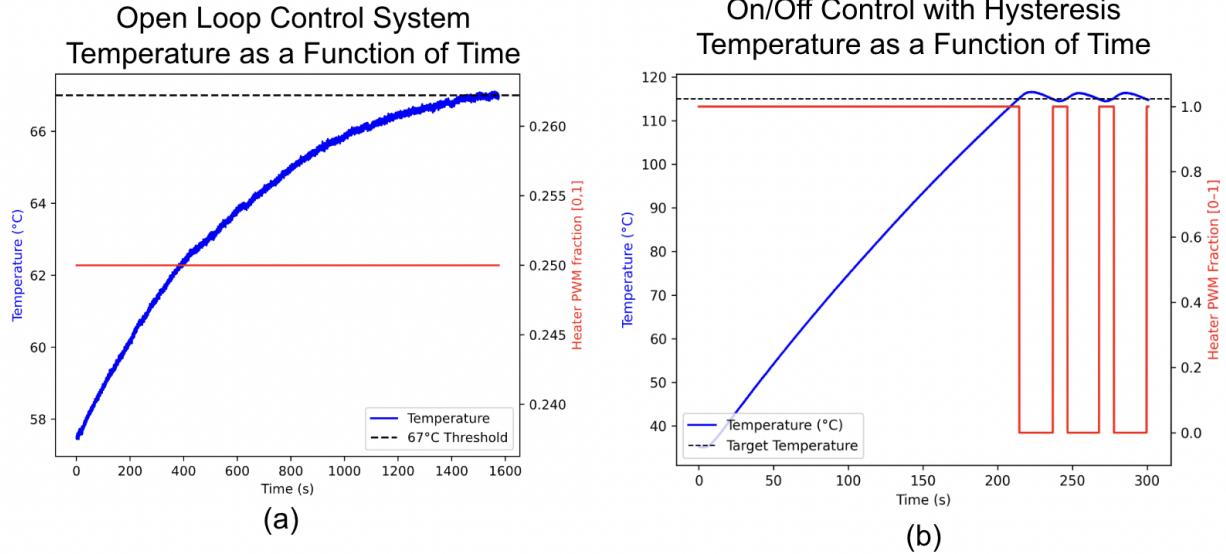


Figure 2: (a) Open loop control system in which a power setting is specified and the aluminum block is allowed to equilibrate. (b) On/Off control with hysteresis with a window of 0.2°C around a target temperature of 115.0°C .

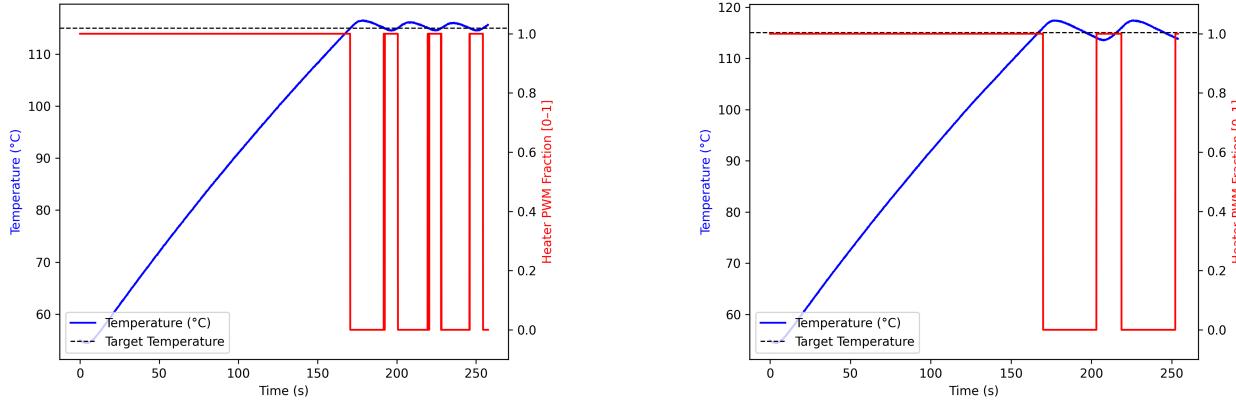


Figure 3: On/Off control with zero hysteresis. 6 power switches over 90 second stable period.

Figure 4: On/Off control with a hysteresis window of 20°C . 9 power switches over 90 second stable period.

Figure 5: Comparison of minimum and maximum tested hysteresis widths for the *on/off* controller. All other parameters are equal.

PI controller test were performed with and without limiting the I accumulator to be zero outside the proportional band. The results are displayed in Fig 8.

Fig 11 compares trials with and without “smoothing” D . The smoothing constant in the maximum case of Fig 11 was 0.999, compared to a more moderate 0.95 in Fig 13 - allowing

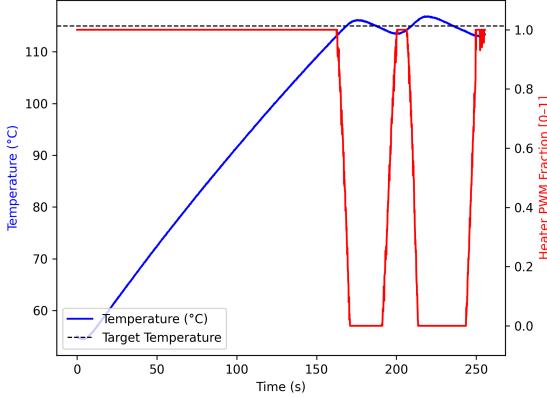


Figure 6: Temperature response plot for a PI controller with I limited to zero outside of the $0 < P < 1$ band. The T_{rise} of this experiment is 121.24s , with an overshoot of $\Delta T_{\text{over}} = 1.86^\circ\text{C}$.

Figure 8: Comparison, limiting the I accumulator to be zero outside of the $0 < P < 1$ band. For both controllers, $K_p = 0.5$, $K_i = 0.05$, target temperature is 115.0°C . As the second test hit termination conditions before stabilizing, the true overshoot of this controller is unknown.

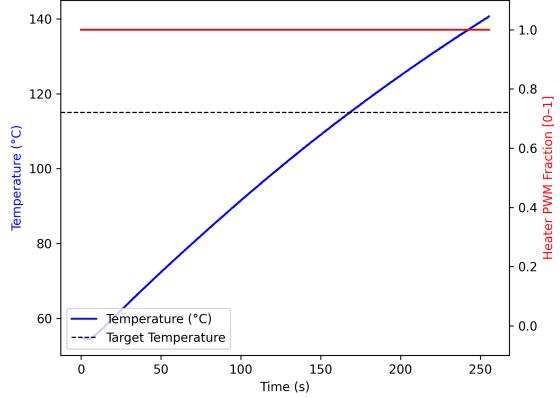


Figure 7: Temperature response plot for a PI controller with the I accumulator active everywhere. The T_{rise} of this experiment is 121.17s , with an overshoot of $\Delta T_{\text{over}} > 25.70^\circ\text{C}$.

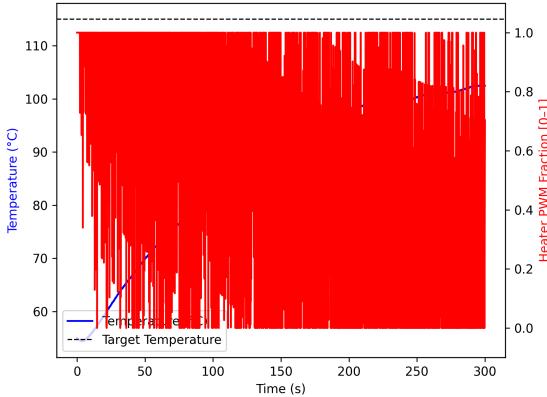


Figure 9: PID Controller without any D -smoothing

Figure 11: Comparison of effect on smoothing the D parameter in two PID runs with otherwise identical controls.

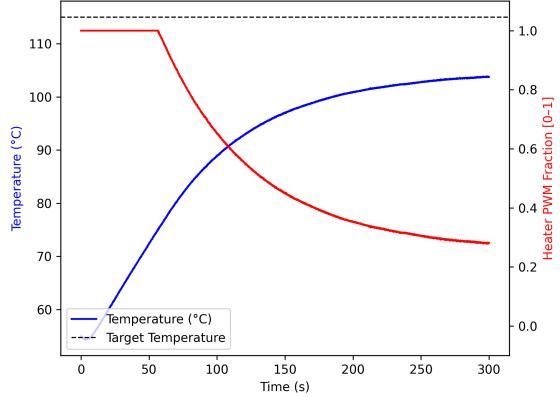


Figure 10: PID Controller with extreme D -smoothing,

$$D_t^{\text{smoothed}} = 0.001 \cdot D_t^{\text{noisy}} + 0.999 \cdot D_{t-1}^{\text{smoothed}}$$

faster reactions at the cost of some minor distortion at the edge of the domain.

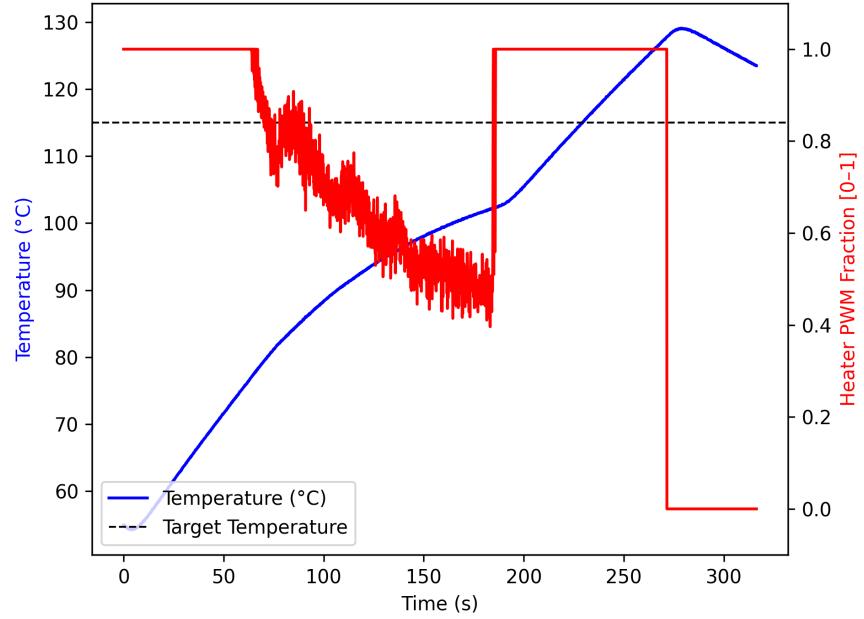


Figure 12: Temperature response plot for a typical PID controller *early* in the search process. $K_p = 0.078$, $K_i = 0.125$, $K_d = 0.5$. Target temperature of 115.0°C . The T_{rise} of this experiment is 180.43s , with an overshoot of $\Delta T_{\text{over}} = 14.07^{\circ}\text{C}$ and a steady-state error of $E_{\text{ss}} = 11.05^{\circ}\text{C}$.

Fig 12 displays one of the initial tests for the PID controller, with far-from-optimal parameters. To contrast, Fig 13 shows the best PID controller found over a search process testing 248 controller iterations, with a total search time of more than 30 hours².

²Fairly good controllers were found much earlier in the automated search process, and controller quality hit diminishing returns quickly.

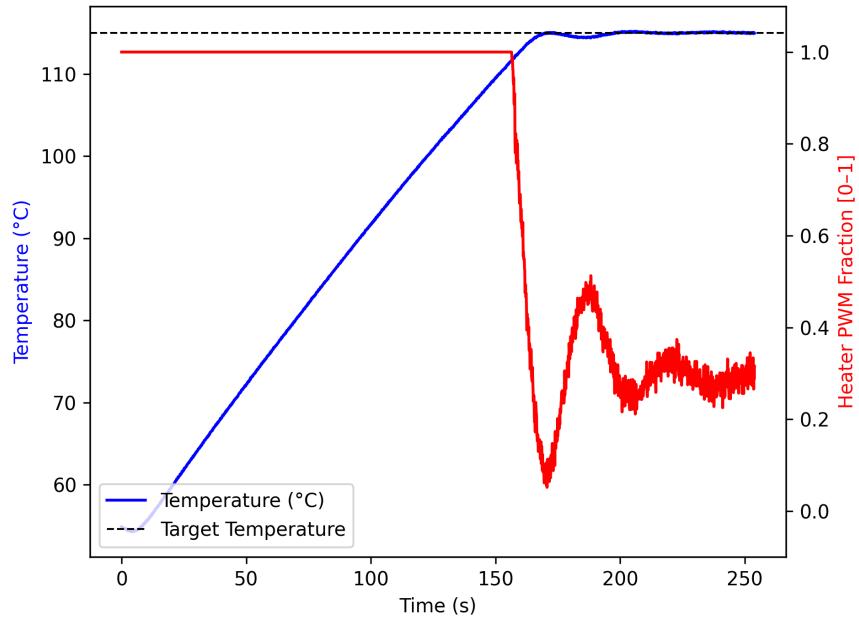


Figure 13: Optimized PID control temperature response plot with target temperature of 115.0°C . $K_p = 0.364540$, $K_i = 0.014467$, $K_d = 0.062500$. The T_{rise} of this experiment is 119.17s , with an overshoot of $\Delta T_{\text{over}} = 0.20^{\circ}\text{C}$ and a steady-state error of $E_{\text{ss}} = 0.16^{\circ}\text{C}$. Found after over 30 hours of machine search.

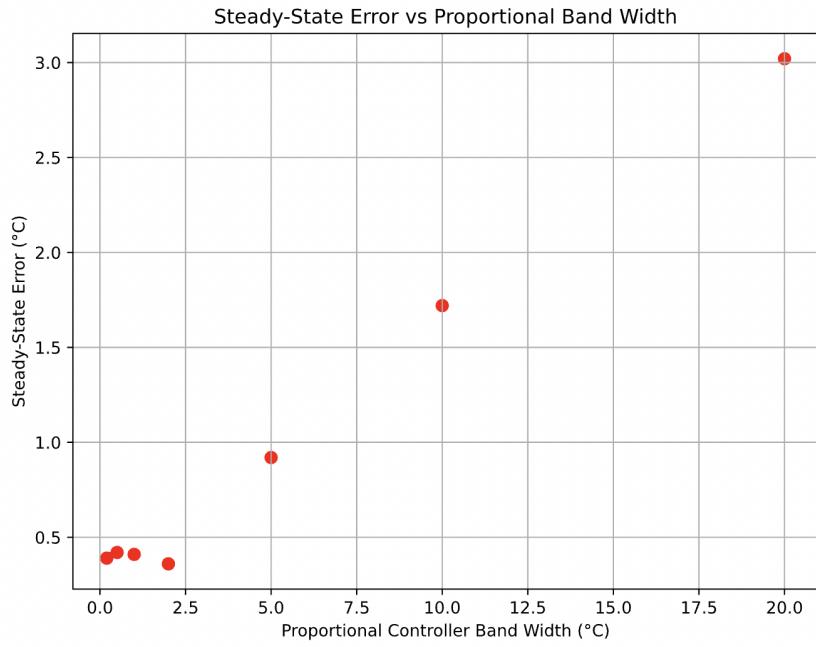


Figure 14: Proportional band width trial aggregate, comparing the proportional controller band width and its associated steady-state error.

4 Discussion

Examining the open loop configuration, Fig 2 (a), we find the glacially slow rise, lack of stabilization, and lack of a way to determine the output temperature without experimentation clearly demonstrate the limitations of an open loop system for thermal control.

As shown by Fig 5, although on/off hysteresis did achieve its goal in slowing the rate of oscillations, this proved to be useless in the control of this particular system. Even without any hysteresis, there was significant the lag-time between changing input power and detecting a change in temperature - as thermal energy propagated between heater and thermometer. This lag-time was already long enough as to prevent undesirably rapid oscillation in the *on/off* controller, without the precision penalties of additional hysteresis.

Figure 14 displays the relationship between the steady-state error and the proportional controller band width. Here, each data point represents the outcome of a single control trial; and as the proportional band width narrows, a generally lower steady-state error follows suit, thus indicating an improved accuracy of the control system.

Since this was only a quick confirmation of expectations before the PID trials, each trial was conducted only once without repeated measurements, thus each data point has no statistical error associated with it.

For *P*-only models, lower band width controllers are able to more precisely able to track the target value, at the cost of greater oscillations. In our findings, these effects seem to approximately balance out for any band-width smaller than $\sim 2.5^{\circ}C$, without any clear minimum point denoting a single best controller (see fig 14). For any band-width larger than $\sim 5^{\circ}C$, the increased oscillation-precision isn't enough to counteract the decreased tracking accuracy, and over-all controller performance diminishes roughly linearly with increasing bandwidth.

Much as there were distortionary effects of a noisy *D* term, when the total output was clamped to the range $0 - 1$, it's possible the quantisation of the output to intervals of 1/60th distorts the contribution of *I* term. However, any such effect would be mild, and so we chose to leave it unexamined.

In Fig. 13, the controller with the best total *quality* (a combination of rise time and E_{ss} described in **Experimental Methods 2**) is shown. The optimised controller was found using the ansatz detailed in Sec. 2, where the proportional gain parameters in Eq. (1) are derived by running successive tests and observing the system's response to the changes in each gain, tuning the controller onto the best set of parameters for the system.

Our optimised proportional gain parameters were $K_p = 0.364540$, $K_i = 0.014467$, and $K_d = 0.062500$, producing a rise time of 119.17 sec. (i.e. the time from first crossing of 70°C to first entering $\pm 1^\circ\text{C}$ of the target temperature), and a steady state error of 0.16°C (the average error over the period after the system has settled).

For a primitive approximation of the heat capacity of the system, we can use the data obtained from Fig. 8 and integrate the power of the heater over the rise time, and divide by the change of temperature observed within this region:

$$C = \frac{PT_{\text{rise}}}{\Delta Tm} = \frac{(20 \text{ W})(121.24 \text{ s})}{(114^\circ\text{C} - 60^\circ\text{C})(29.0 \text{ g})} = 1.54 \text{ J}/^\circ\text{C}$$

Which is an overestimate from the reported value of 0.90 J/g°C.

5 Conclusion

The broad purpose of this lab was to provide a comparative investigation of the several control strategies that exist to regulate and stabilize temperature. Namely, an open-loop control system, and on/off control with hysteresis, and a PID control. Each method, for consistency, was tested under the same desired setpoint temperature, under relatively the same ambient conditions.

The open-loop strategy, while straightforward, was inherently limited, as it provided no mechanism to correct for thermal disturbances, and relies solely on equilibration with ambient conditions - which often results in a steady state error or drift in the temperature. The on/off control with hysteresis provided a notable improvement due to the implementation of a feedback nature. The significant limiting factor in this strategy is that the feedback

lends itself to indefinite cycling around the target temperature without any mechanism for convergence.

By contrast, the PID control system demonstrated significant improvements to the limitations of the prior strategies. By careful optimization and leveraging the system's response to each gain factor, a feedback loop was engineered to minimize steady-state offsets and suppress oscillations; allowing for timely and accurate convergence to the given target temperature.

References

- [1] M. Mohseni and S. M. Islam, “A new vector-based hysteresis current control scheme for three-phase pwm voltage-source inverters,” *IEEE Transactions on Power Electronics*, vol. 25, no. 9, pp. 2299–2309, 2010. [1](#)
- [2] C. Kadu and C. Patil, “Design and implementation of stable pid controller for interacting level control system,” *Procedia Computer Science*, vol. 79, pp. 737–746, 2016, proceedings of International Conference on Communication, Computing and Virtualization (ICCCV) 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050916002283> [2](#)
- [3] T. K. Palaniyappan, V. Yadav, Ruchira, V. K. Tayal, and P. Choudekar, “Pid control design for a temperature control system,” in *2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, 2018, pp. 632–637. [2](#)