

Development of a high-temperature PID controller with 0.25K precision

T. Foldy-Porto¹

¹*Department of Physics, Yale University*

(Dated: 2 May 2019)

We describe the design and development of a PID temperature controller with 0.25°C precision for an oven operating between 300°C and 600°C. We present the considerations that went into the mechanical, electrical, and software design for the controller, as well as the results we obtained and points of future improvement.

I. INTRODUCTION

Generating an atomic beam of constant flux is a problem of principal interest to anyone trying to produce a Bose-Einstein condensate (BEC). This is relevant to the Ultra-cold Quantum Matter Lab at Yale University which studies the quantum many-body problem through ultra-cold quantum matter that is prototypical of larger, more complex systems. In order to create the ultra-cold BEC, it is necessary to cool atoms (⁶Li, in our case) down to temperatures on the scale of 10⁻⁹K. To achieve this, the lithium atoms are first heated in an oven until they enter a gaseous state. They then exit the oven through a small aperture, creating an atomic beam (shown in figure 1), and enter a magneto-optical trap that cools the atoms down to the desired temperature. This trap consists of a laser—which is tuned on resonance with the lithium atoms such that it excites the emission of photons, thus slowing the atoms down by conservation of momentum—and a Zeeman slower, which exploits the Zeeman effect to counteract the changing Doppler shift (and changing resonant frequency) experienced by the atoms as they slow down throughout the trap.

The efficacy of the magneto-optical trap depends on the stability of the atomic beam, which in turn depends on the oven maintaining a constant flux of atoms through its aperture. The atomic flux Φ , defined as the number of atoms per unit time passing through the aperture, obeys the relationship

$$\Phi = \frac{1}{4}nA\bar{v} \quad (1)$$

where n is the density of the gas in the oven, A is the area of the aperture, and \bar{v} is the average velocity of the atoms exiting the oven. In our case, the vapor density n is given by the vapor pressure of the lithium gas at the temperature of the oven. This quantity is given by the *vapor pressure equation*:

$$n = P(T) = n_0 e^{-\frac{L}{RT}} \quad (2)$$

where n_0 is a constant, L is the latent heat of the lithium gas, and R is the universal gas constant. Combining equations 1 and 2, we see the dependency of atomic flux Φ on temperature T :

$$\Phi = \frac{1}{4}n_0 A \bar{v} e^{-L/RT} \quad (3)$$

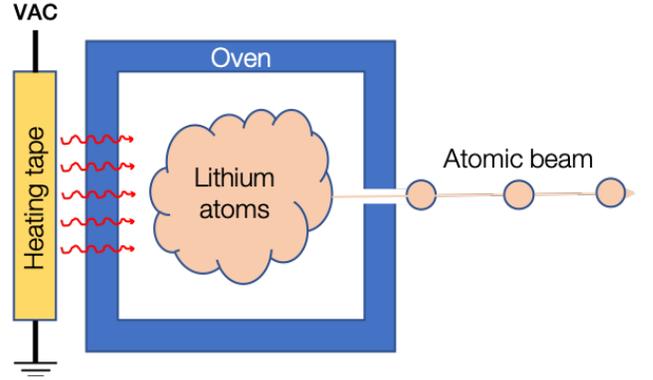


FIG. 1. The oven implemented in the Lithium Lab. The lithium atoms are placed in an insulated metal chamber that is adjacent to a heating tape connected to an AC voltage source. The heat flux into the oven is proportional to the electrical power through the heating tape.

Therefore, variations in T have an exponential effect on the flux of the atomic beam (shown in figure 2), meaning that the performance of the magneto-optical trap is very sensitive to variations in the temperature of the oven. This necessitates the creation of a device to precisely stabilize the temperature of the oven.

II. MECHANICAL DESIGN

A. The existing oven

The mechanical design of the temperature controller was constrained by the existing lithium oven. Luckily, this oven was quite simple and did not impose many constraints. Shown in figure 1, the oven consists of a metal enclosure, insulated with aluminum foil, and a heating tape (STH101-020) that is connected to an AC voltage source. The heat Q applied to the lithium atoms is proportional to the electrical power through the heating tape:

$$Q \propto \frac{V_{rms}^2}{R} \quad (4)$$

where R is the resistance of the heating tape ($24 \pm 0.5\Omega$ in our experiment).

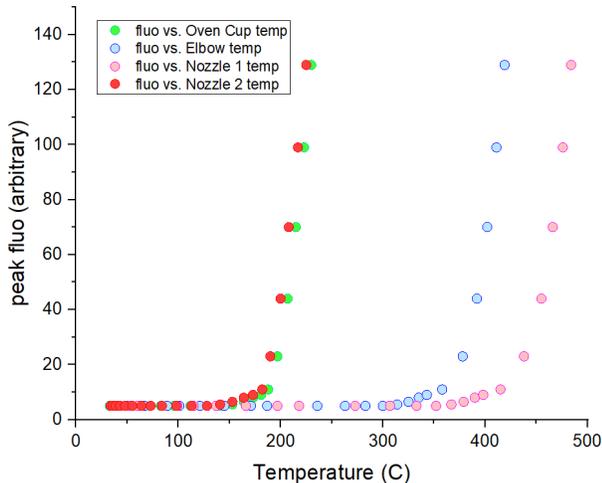


FIG. 2. The flux of our atomic beam as a function of temperature. The units of the flux are arbitrary, but this plot serves as a qualitative indicator of the exponential relationship between Φ and T . In the region where we’re operating the oven (around 500°), we see that a small change in temperature results in a large change in flux.

B. Control methods

Given equation 4, to control the temperature of the oven it is necessary to control the voltage across the heating tape. Initially, this was achieved by a variable transformer (or variac), but that proved difficult to automate because the dial that adjusted the voltage was large and unwieldy. Motorized variacs do exist, but they are undesirable for two reasons: they are expensive, and they are often driven by DC motors, meaning controlling the dial position accurately requires the implementation of a sophisticated motor controller and encoder feedback system.

To simplify this process, we switched out the variac for a TRIAC (triode for alternating current) dimmer switch; these devices are commonly used in houses to dim light bulbs, and they provide an easy way to produce a variable AC voltage source. We selected the Lutron Rotary Dimmer Switch for Incandescent Bulbs¹ Essentially, the dimmer switch takes a sine-wave and chops it up depending on some threshold voltage set by the user. Internally, there is a TRIAC that conducts only when the voltage on its gate is higher than the threshold. Every time the input wave crosses zero, the TRIAC shuts off. A capacitor connected to the input acts as an integrator for the sine wave, and when the voltage across this capacitor crosses the threshold, the TRIAC turns on and conducts again. Thus, by varying the threshold voltage (via a potentiometer, for example) the user can set the phase at which the output starts to “see” the sine wave on the input.

C. Our design

The rotary switch configuration allowed the user to set the threshold voltage by turning a lightweight potentiometer. By coupling a small servo motor to the rotary switch, we had means to precisely automate the threshold voltage, and thus the V_{rms} that was output to the heating tape. The single-wire, position-controlled nature of the servo motor made it preferable to a DC or stepper motor.

The built-in mounting holes on the the rotary dimmer switch housing made the mounting of the servo motor straightforward. We constructed a wooden mount to connect the servo motor to the switch: one end of the mount was screwed into the mounting holes on the switch with #8-3/4” wood screws; the other end was screwed into the servo motor with #6-1/2” wood screws. The servo motor came with several mounting hubs, making it easy to couple the output shaft of the servo to the plastic hub on the rotary switch using small wood screws. To make the control code simpler, the 0° position of the servo was aligned with the “off” position of the switch. This design proved to be fairly strong, as it was able to withstand a few accidental drops.

III. ELECTRICAL DESIGN

With the question of how to physically adjust the temperature of the oven solved, we were able to move on to the design of the closed-loop control system that would stabilize that temperature. A block diagram schematic of our controller is shown in figure 3. Essentially, the temperature of the oven is measured by a J-type thermocouple, the output of which is passed through an amplifier/filter circuit before being fed into an Arduino microcontroller. Depending on what the user desires, the Arduino runs one of a variety of temperature-control programs. The primary program it runs is a PID loop that compares the temperature of the oven to a user-defined temperature set-point. The Arduino then adjusts the position of the rotary switch (via the servo motor, as described in section II.C.), which in turn heats or cools the oven.

The design of the electrical system was further influenced by the temperature constraints of the oven: it would be catastrophic for the experiment if the temperature dropped below 300°C or rose above 600°C . Below that range, two things happen: the dynamics of the lithium atoms becomes complex due to capillary action caused by the temperature gradient; and the lithium solidifies and clogs the oven, forcing us spend time and resources taking apart and cleaning the experimental apparatus. Above that range, the vacuum seals on the experiment break, again requiring dismantling and labor-intensive repair. This prompted the design of a fail-safe circuit that provided a hardware-level protection against the failure of the PID controller. The design of this cir-

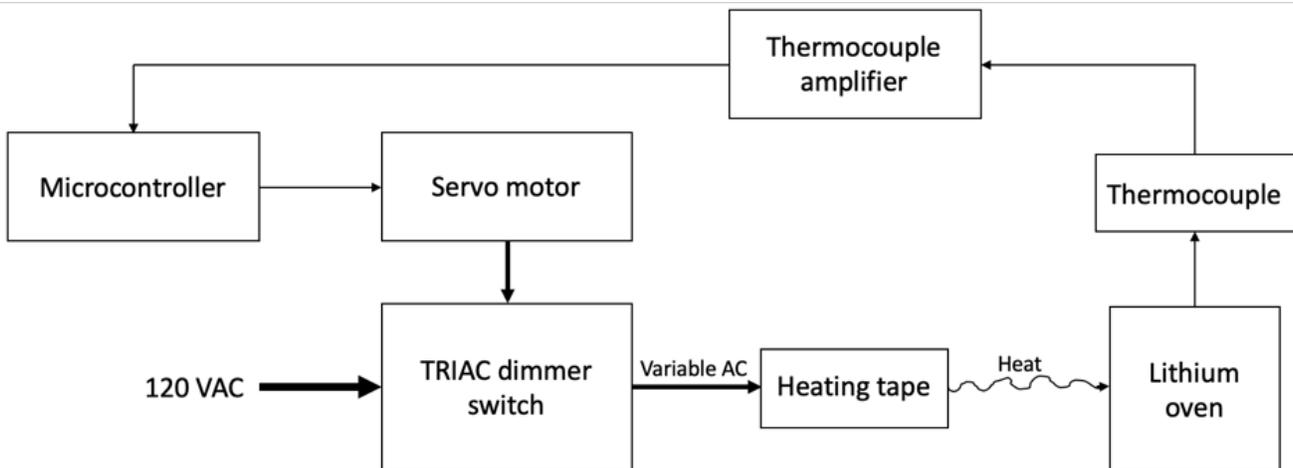


FIG. 3. High-level block diagram of our temperature controller. A microcontroller sets the position of a servo motor that is mechanically attached to a TRIAC rotary dimmer switch. This switch takes 120 VAC and outputs an AC waveform with variable power (dependent on the position of the switch), which then determines the temperature of the heating tape. The temperature of the lithium oven is sensed by a thermocouple, and this signal is fed back to the microcontroller to close the loop.

cuit and the thermocouple amplifier was non-trivial and warrants more detailed explanation.

A. Thermocouple amplifier circuit

This circuit was motivated by the small amplitude of the thermocouple output voltage due to the relative weakness of the thermoelectric effect. We chose a J-type thermocouple because it provided the best accuracy in the operating range of our oven. However, the Seebeck coefficient of J-type thermocouples is only $50 \mu\text{V}/^\circ\text{C}$. To put this in perspective, consider that the Arduino microcontroller we used to interpret this voltage had a 10-bit ADC with a reference voltage of 5V. This gives a resolution of 4.9 mV per bit, or 98°C per bit. We remedied this poor resolution using two methods: an amplifier/filter circuit to both boost and clean up the output of the thermocouple, and a software technique called *oversampling and decimation*.

The circuit we designed to amplify the thermocouple voltage, shown in figure 8 in the appendix, was based on the instrumental amplifier, a type of differential amplifier that has a high common-mode rejection ratio (CMRR). Because the voltage produced by the thermocouple was so small, common-mode DC noise generated by inductive effects tended to dominate the signal; the high CMRR characteristic of the instrumental amplifier allowed our circuit to remove this DC bias, whereas a normal differential amplifier would have boosted it. The theoretical differential gain of the circuit is given by:

$$A_d = \left(\frac{75.1\text{k}}{75.1\text{k} + 75.1\text{k}} \right) \left(1 + 2 \frac{1\text{M}}{47\text{k}} \right) \left(\frac{10\text{k}}{4.7\text{k}} \right) = 185.27 \quad (5)$$

The values in the equation above come from the resistor values in the amplifier circuit itself, shown in figure 8. With the amplifier circuit in place, the resolution of the thermocouple was increased to 0.529°C per bit. To attain the sub- 0.25°C precision we were aiming for, however, it was necessary to employ oversampling and decimation to increase the resolution of the ADC on the Arduino. Essentially, this technique trades frequency resolution for voltage resolution; by sampling at a frequency higher than the Nyquist frequency and then averaging the result, we are able to obtain greater resolution on the analog signal. For every bit of additional resolution n , it is necessary to sample 4 times the Nyquist frequency:²³

$$\text{sample frequency} = (2^n)^2 \quad (6)$$

The temperature of the oven does not vary rapidly with time and we do not care about resolution in the frequency domain. Our implementation of this technique is shown in figure 4. For a more detailed explanation of the method, see the appendix of reference 3. Together, the amplifier circuit and oversampling software allowed us to measure the voltage of the thermocouple with millivolt precision.

B. Fail-safe circuit

Due to the nature of the lithium atoms in our oven, it would be catastrophic if the temperature of the oven

```

uint16_t analogReadN(uint8_t pin, uint8_t bits) {
  bits = constrain(bits, 10, 16) - 10;

  int samples = 1 << (bits << 1);

  uint32_t sum = 0;
  for (int i = 0; i < samples; i++)
    sum += analogRead(pin);

  return sum >> bits;
}

```

FIG. 4. Our implementation of the *oversampling and decimation* technique. For reference: words in blue establish data types; words in orange are built-in functions; and words in green are control structures (if, for, while loops, etc.).

dropped below 600°C or rose above 600°C. We implemented software stops to prevent this from occurring, but to be safe we developed an isolated fail-safe circuit that switches the input of the heating tape to a safe “baseline” AC source in the event of the PID controller driving the voltage too high or low. This baseline source is calibrated to heat the heating tape to a temperature inside the safe range.

In essence, the circuit requires three forms of authentication before it enables the PID controller. The primary element of the fail-safe circuit (shown in figure 9) is a double-pole double-throw (DPDT) relay that switches the input of the heating tape between a constant AC source, set by a variac, and the PID-controlled AC source. The coil voltage of the relay is 12V DC, so it was necessary to use an N-MOSFET to drive the relay from 5V logic. The default (no power) position of the MOSFET connects the heating tape to the baseline source. The gate of the MOSFET is connected to the output of an AND IC (SN74AC08).

This line is driven high if and only if: 1) the thermocouple signal is above the *low* threshold; 2) the thermocouple signal is below the *high* threshold; and 3) the Arduino asserts its presence by driving an input high (the input is connected to a pull-down resistor). In the event that the Arduino fails, its connection to the fail-safe goes to ground, the output of the AND gate goes low, and the relay returns to the default mode that connects the baseline source to the heating tape. The same chain of events happens if the output of the thermocouple rises above or below certain thresholds, which are set by the user and correspond to low-temperature and high-temperature limits.

IV. PROGRAMMING THE CONTROLLER

Having completed the mechanical and electrical assembly of the controller, the final step was to implement control software. Among the many available types of

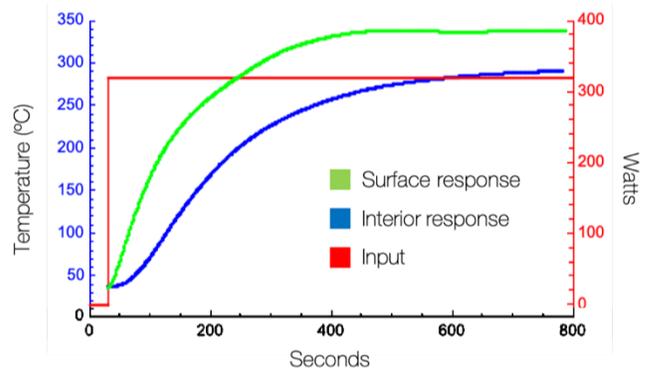


FIG. 5. The temperature response of the plant (1” aluminum tube, in our case) to a step input. The response was measured at two points: on the surface of the tube, shown in green, and in the interior of the tube, shown in blue. Note that although the surface response is colored green, it is in units of temperature and measured according to the blue axes. As one would expect, the interior was slower to respond than the surface.

controller we could have designed (bang-bang, internal model, etc.), a straightforward PID controller seemed a sufficient starting point. In fact, in the final algorithm we set the derivative gain to zero, so we really produced a PI controller.

Initially, there were two factors that presented serious difficulties in stabilizing the control loop: 1) there was a significant time delay in the temperature response of the system, and 2) the response of the system looked different depending on whether we were driving it hotter or colder. The time delay can be seen in both figure 5 and figure 11 (in the appendix).

To tune the PID loop, we first recorded the response of the system to a variety of inputs. From the results of these tests (which can be seen in figures 12 in the appendix), we recovered the impulse response of the system and were able to deduce the transfer function. This allowed us to simulate the plant response to a PID controller with variable gains, thus tuning the controller to the optimal settings. However, as soon as the system was disturbed slightly—someone bumped into it, the heating tape shifted slightly, the room temperature changed, etc.—the impulse response changed. Because of this, we decided that it would not be practical to use this tuning method moving forward.

In our search for a way to tune the parameters of the PID controller without measuring the system response, we settled on the Ziegler-Nichols method, a heuristic tuning method developed by John Ziegler and Nathaniel Nichols in 1942⁴. The technique consists of first setting the integral gain K_i and the derivative gain K_d to zero, and then adjusting the proportional gain K_p (increasing it from zero) until the system enters steady oscillations. This value of K_p is called the “ultimate gain” K_u and the period of these oscillations is labeled T_u . For a PI controller, these two quantities are used to determine

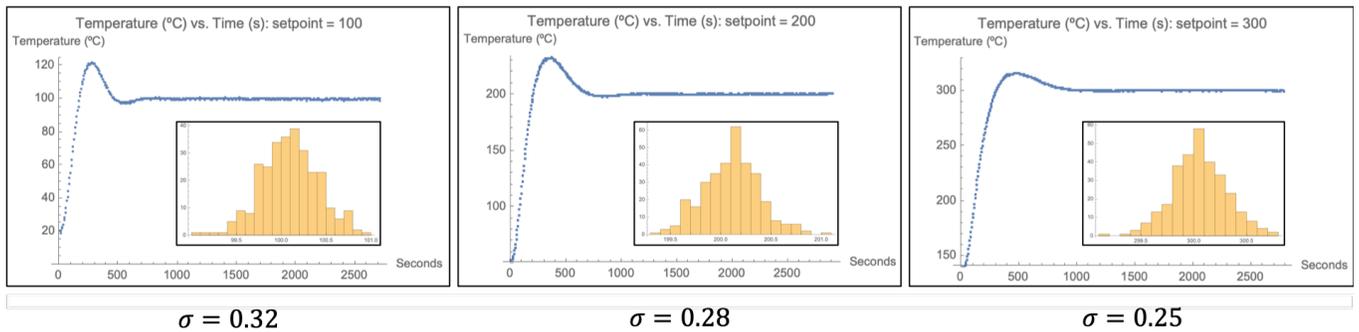


FIG. 6. The temperature of the system ($^{\circ}\text{C}$) plotted against time (seconds) for three setpoints: 100°C , 200°C , and 300°C . The size of the temperature bins was 0.1°C , and data points were recorded every 5 seconds. A closeup of the inset histograms are shown in figure 10.

the optimal parameter values according to the following rules:

$$K_p = .45K_u \quad (7)$$

$$K_i = .54\frac{K_u}{T_u} \quad (8)$$

Using this method, we were able to create a control loop that stabilized the temperature of the oven to 0.25°C at a setpoint of 300°C . The results of several trials, each with a different setpoint, are shown in figures 6 and 10.

V. POINTS OF IMPROVEMENT

The thermocouple amplifier circuit could be made better by moving it from a breadboard to a printed circuit board (PCB). Breadboards are known to have high amounts of parasitic capacitance, and although there were no such effects that were noticeable when measuring the thermocouple voltage to millivolt precision, these effects could show up if the gain of the amplifier is increased. If precision beyond 0.001V is required, it will be necessary to increase the gain. A PCB would be ideal in two ways: it would not exhibit nearly as much parasitic capacitance as breadboards; and it would make the circuit more mechanically secure (wires less likely to be pulled loose, etc.).

While our fail-safe circuit is fairly robust, there are several ways in which it could be improved. We did not account for the case in which the thermocouple itself gives an erroneous measurement of the temperature. This problem could be rectified by the implementation of redundant thermocouples that are electrically isolated. It is unlikely that all thermocouples fail at once, and a well-designed control algorithm would be able to differentiate good thermocouple readings from erroneous ones. Additionally, our fail-safe mechanism does not protect the system against total power failure. In the event that

the power to the lab fails (e.g., due to a storm), both the baseline source and the PID source will stop working, and the temperature of the oven will cool to room temperature. Unfortunately, there is no easy solution for this problem aside from supplying the lab with a backup generator. For the most part, however, insuring against total power failure is a problem beyond the scope of the device presented in this paper.

VI. CONCLUSION

The device described by this document succeeded in several ways: it stabilized the temperature of the lithium oven to a high degree of precision (0.2°C), it was cheap to build, and it offered the user a high level of functionality and customizability. To answer the question of whether this was a worthwhile project, it is helpful to compare our device to those that are commercially available. There are many PID temperature controllers available to buy; a typical example of a high-quality one is the Omega CN32PT, which retails for \$225 to \$495 (depending on features included) and boasts a precision of 0.4°C . Not only is the precision achieved by our device better than that of the Omega device, but the open-source availability of our controller makes it significantly more versatile and robust.

Since the microcontroller in the temperature controller we built is easily programmable, our device offers users the ability to customize temperature profiles, remotely record data, interface with other autonomous controllers on the experiment, and program any other functionality that users wish for. Additionally, the user interface we designed is significantly more intuitive and easy-to-use than that of the Omega controller. This cosmetic feature may seem insignificant, but it will reduce potential user-errors in the future that could impact the experiment.

VII. REFERENCES

¹Available on Amazon for \$6.54.

²“Enhancing adc resolution by oversampling,” Tech. Rep. (Atmel Corp., 2005).

³“Improving adc resolution by oversampling and averaging,” Tech. Rep. (Silicon Labs, 2013).

⁴J. G. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” trans. ASME **64** (1942).

VIII. APPENDIX

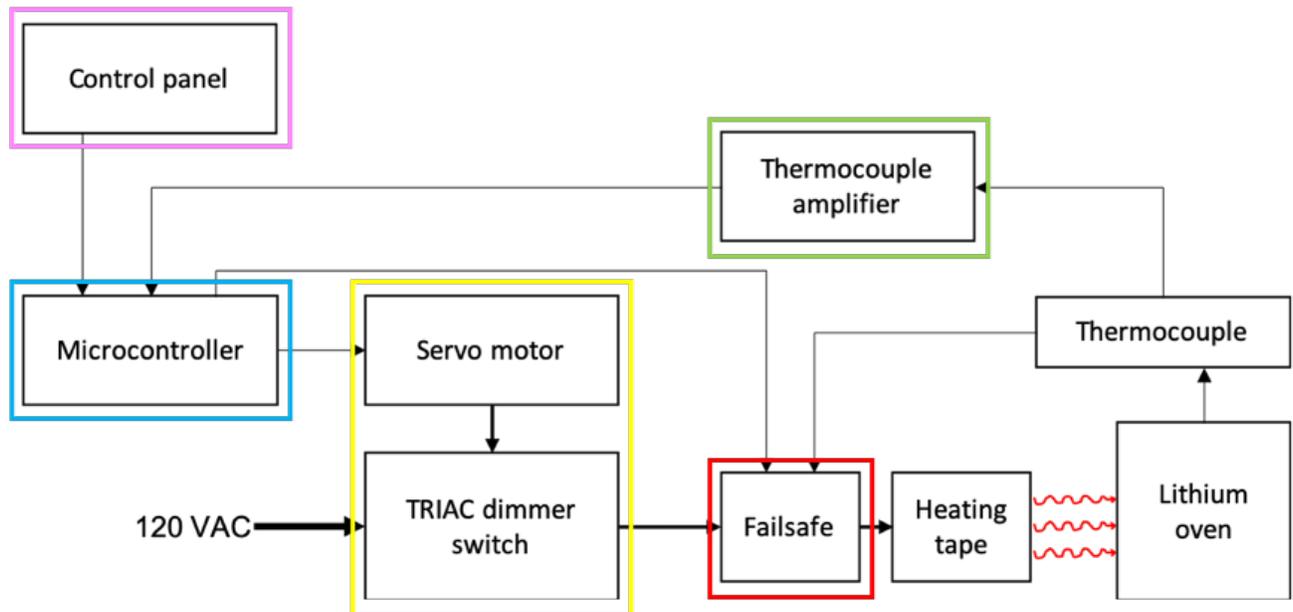
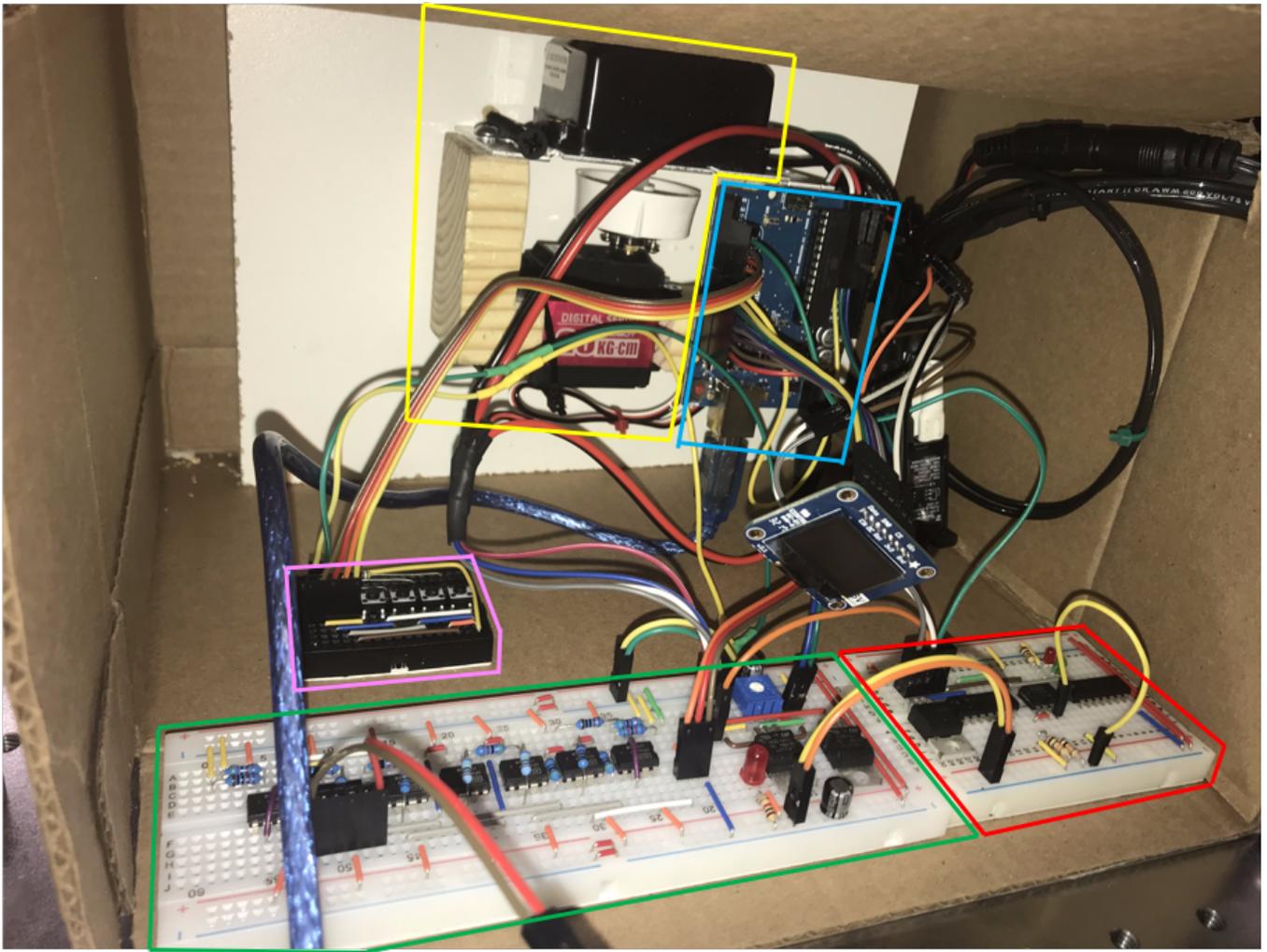


FIG. 7. **Above:** Photograph of the device in place on the lab bench with key components outlined. **Below:** The logical diagram that corresponds to the above image; boxes are color coordinated with the components outlined above.

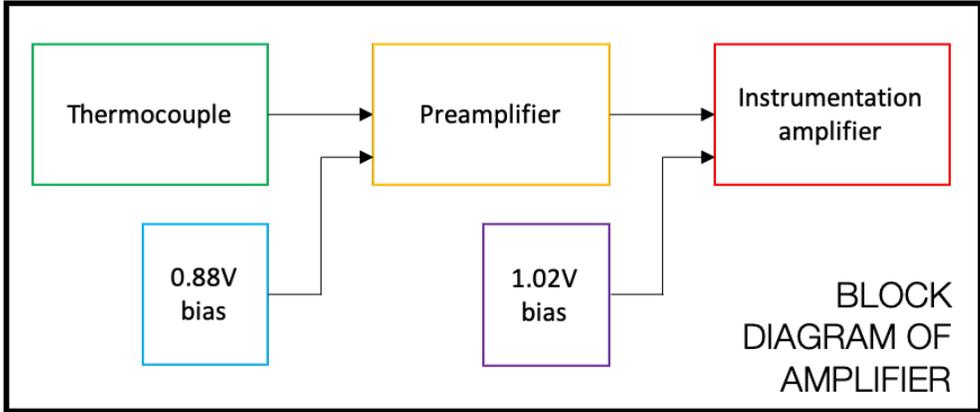
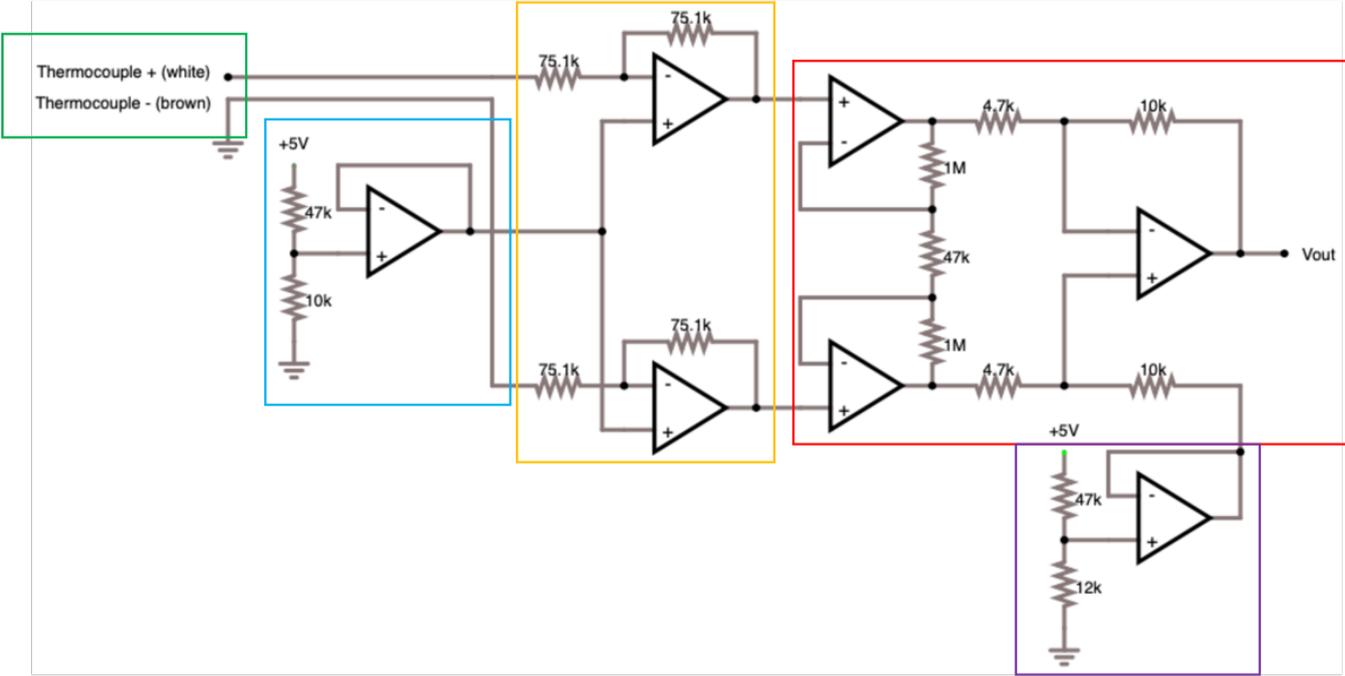


FIG. 8. **Above:** Schematic for the thermocouple amplifier circuit we built for our temperature controller. **Below:** The logical diagram that corresponds to the above schematic. Boxes are color coordinated with the components in the schematic. The circuit is a modified version of an instrumental amplifier, with a gain of 185.27 (see equation 5). Note that the negative lead of the thermocouple must be tied to ground for the amplifier to work properly.

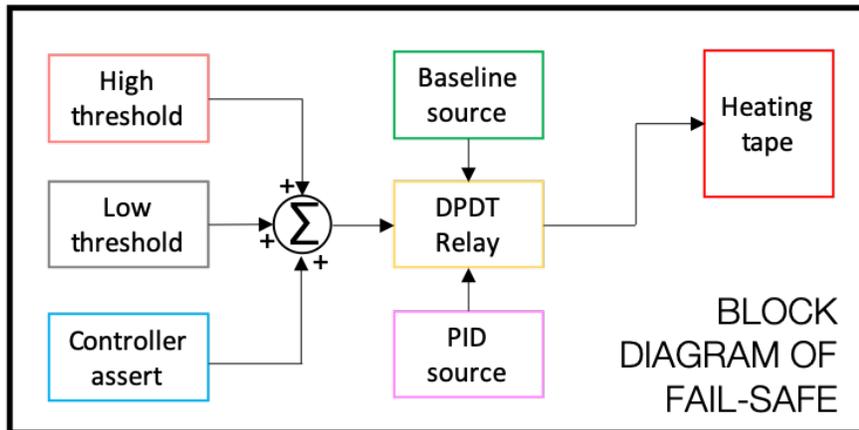
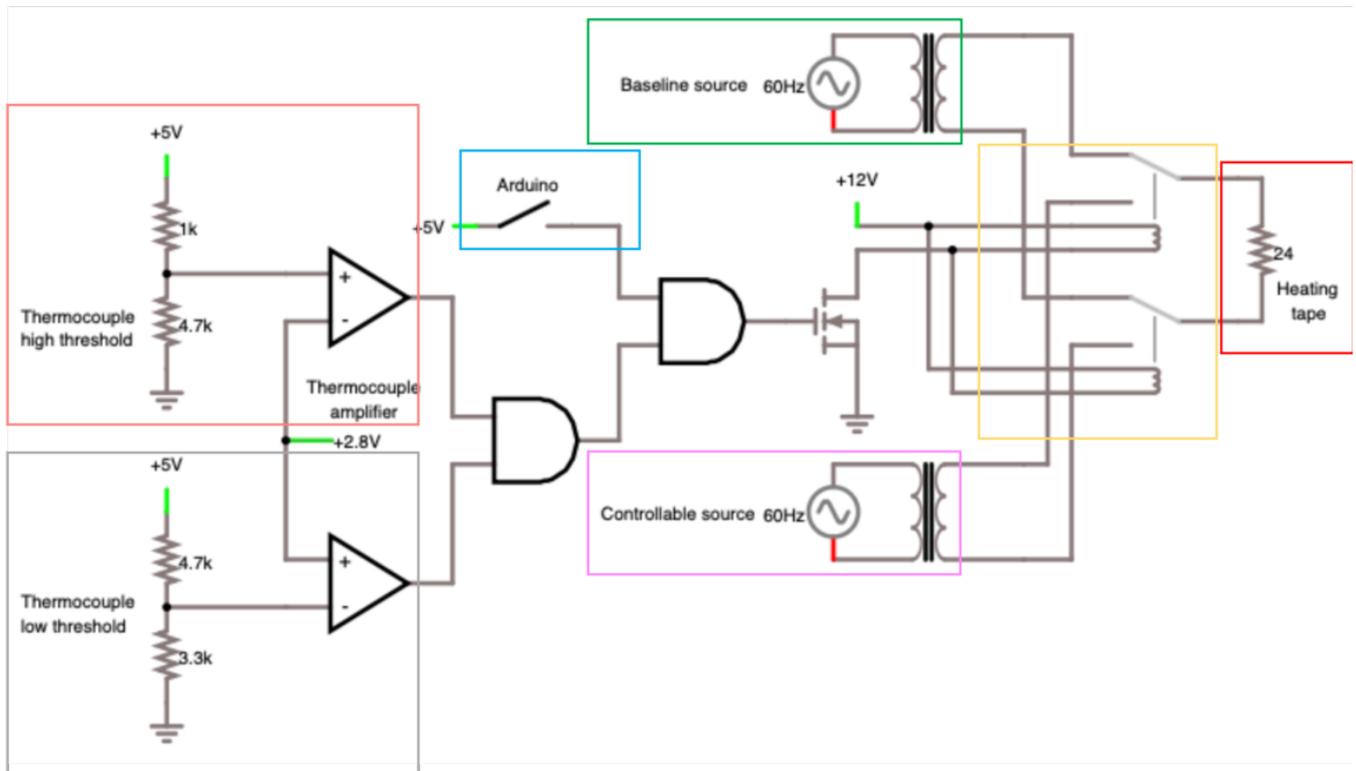


FIG. 9. **Above:** Schematic for the low-temperature fail-safe circuit. **Below:** The logical diagram that corresponds to the above schematic. Boxes are color coordinated with the components in the schematic. For simulation purposes, a switch is shown on the “Arduino +5V” line to represent the ability of the Arduino to drive that line high or low. Additionally, the thermocouple on the non-inverting input of the op-amp has been set to 2.8V, whereas in reality that input is drawn straight from the output of the thermocouple amplifier.

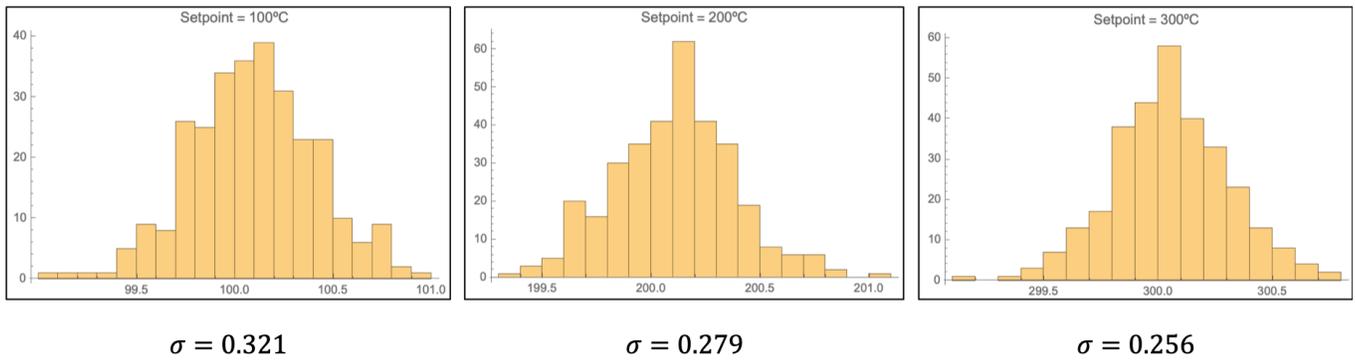


FIG. 10. A histogram of the temperature of the system ($^{\circ}\text{C}$) taken over $\tilde{45}$ minutes for three setpoints: 100°C , 200°C , and 300°C .

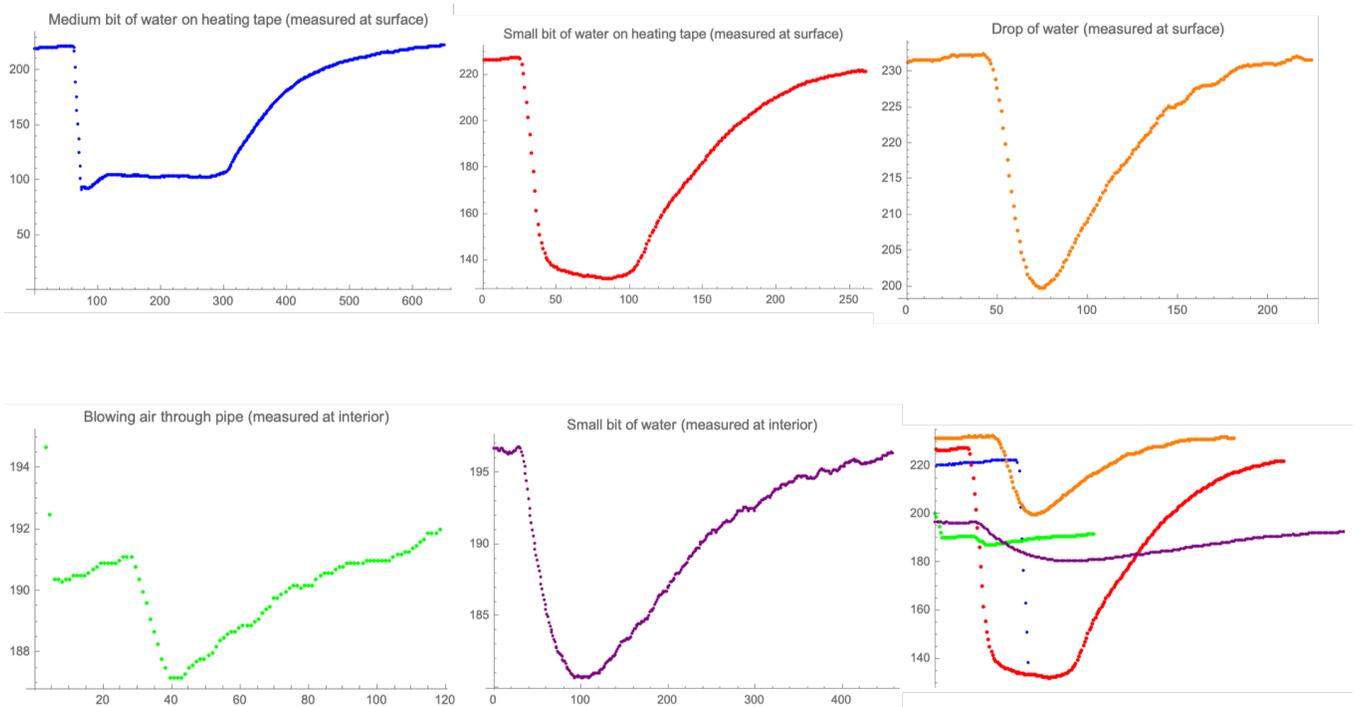


FIG. 11. The system response to various, water-based disturbances. For all but one test, we heated the oven to a constant (but non-PID-stabilized) temperature, poured an amount of water on the heating tape (the size of the amount indicated in the title of the plots), and recorded the response of the system. For the plot shown in green, we blew air through the oven instead of pouring water on it. In the lower-right plot, we show all the trials on the same axes, to give some perspective as to the magnitude and time-scale of the response.

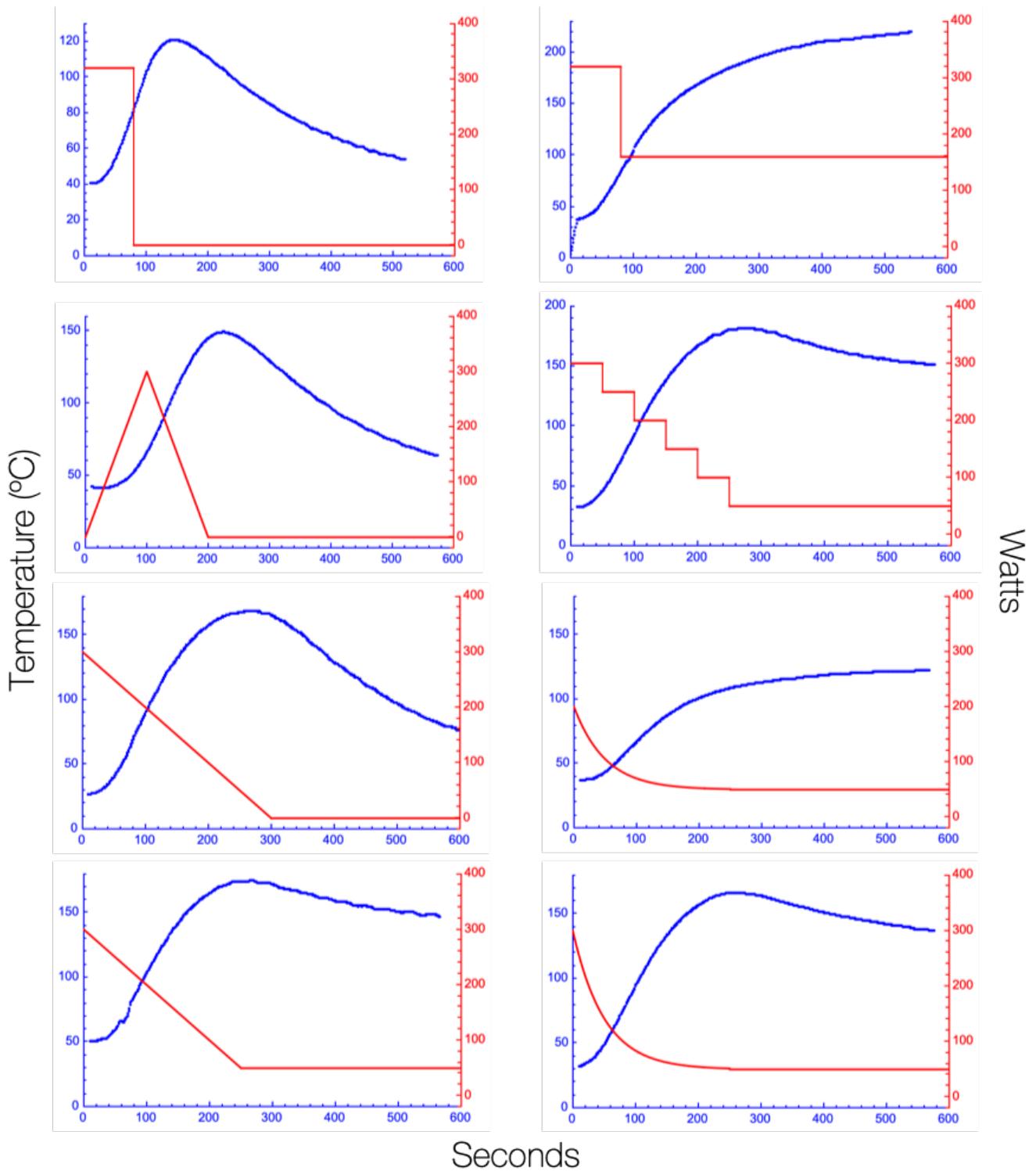


FIG. 12. The system response (blue) to various inputs (red). These tests were mostly exploratory in the sense that we were simply curious about what would happen if X were our input, hence the unusual shapes of some of the input waveforms. Note that before $t = 0$, the value of each input was zero.