Project Report: Thermostat Model

By: Adit Tiwari

1. Introduction

Thermostats are an essential part of modern heating, ventilation, and air conditioning (HVAC) systems. They maintain a comfortable environment by regulating the temperature of a house based on a desired setpoint. This project focuses on modeling a first-order cooling equation to represent the temperature dynamics of a house and designing a digital control system to regulate the temperature. The proposed system emulates the functioning of a thermostat, leveraging discrete-time modeling to simulate its behavior in Simulink.

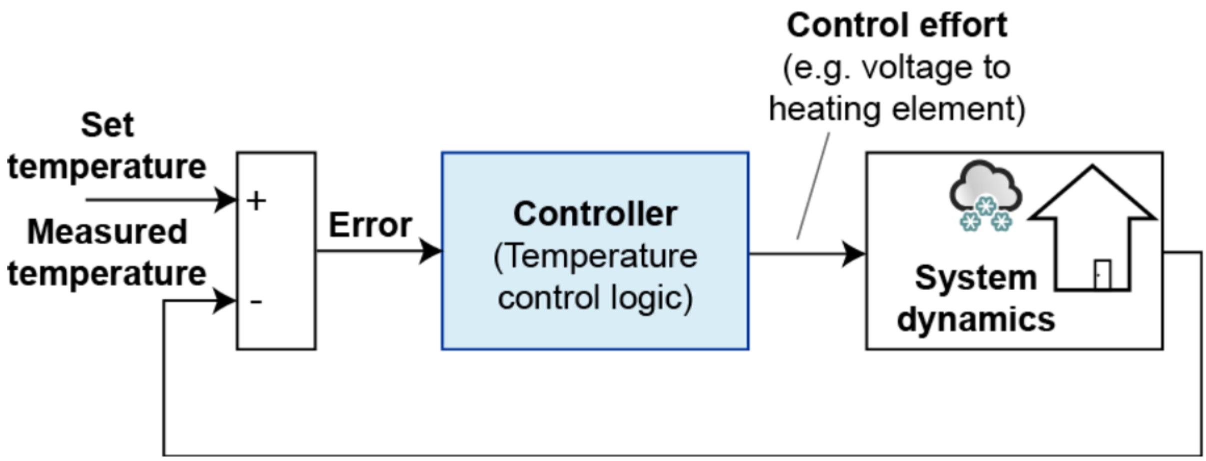
2. Objectives

- To model the temperature dynamics of a house using a first-order cooling equation.
- To design a digital control system that adjusts the temperature based on the current and desired temperatures.
- To simulate the thermostat's performance and analyze its effectiveness in maintaining a comfortable temperature using Simulink.

3. Background

System Overview

The control system model is illustrated in the following block diagram:



This diagram shows the flow of control:

1. The measured temperature is subtracted from the set temperature to calculate the error.
2. The controller processes the error using temperature control logic to generate a control effort (e.g., voltage to the heating element).
3. The control effort influences the system dynamics (temperature of the house).
4. The loop continues until the desired setpoint is achieved.

Modelling the Cooling Dynamics

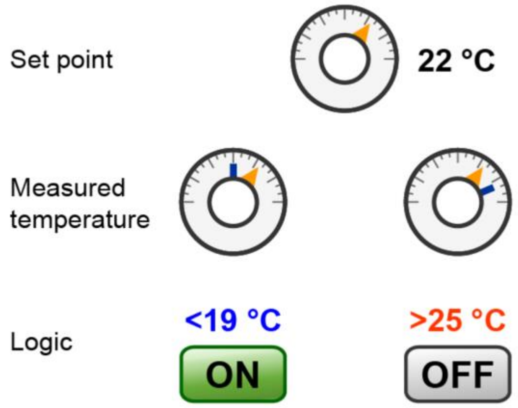
The continuous cooling equation was discretized using the Euler method for numerical simulation. The discrete model is:

$$T_{n+1} = T_n - k(T_n - T_{\text{ambient}})\Delta t$$

Where:

- T_n is the temperature at the n-th time step,
- T_{ambient} is the ambient (outside) temperature,
- k is the cooling constant, and
- Δt is the time step (interval between two consecutive calculations).

Designing the Thermostat Controller

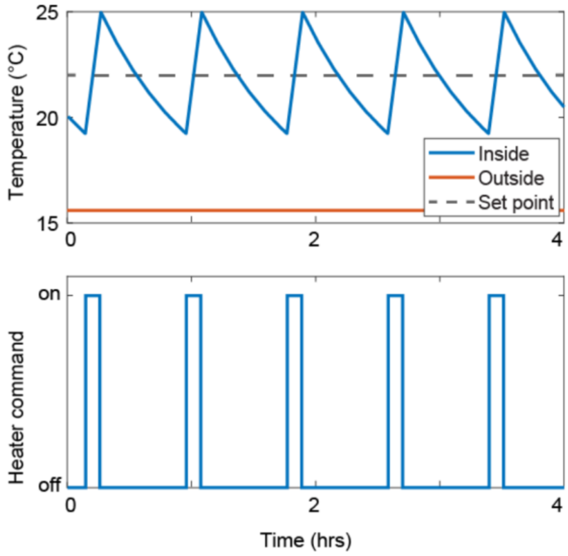


A simple on/off control strategy was implemented:

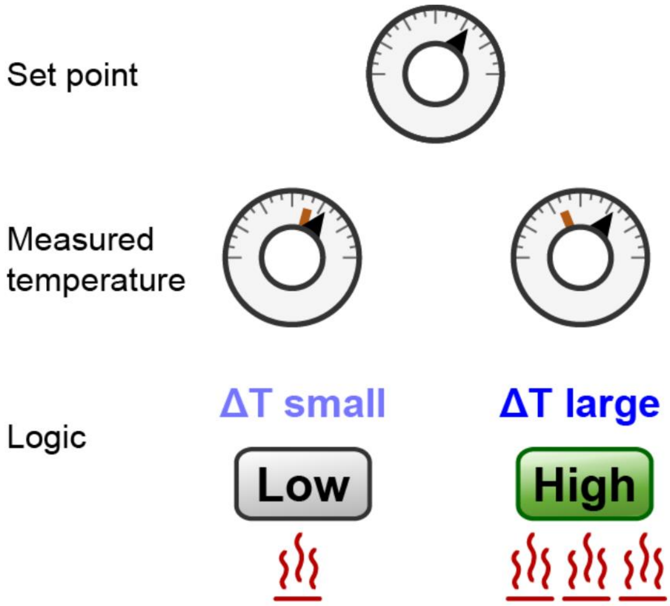
- **Heating On**: If the temperature T falls below the setpoint minus a small tolerance ϵ, i.e., $T < T_{\text{set}} - \epsilon$, the heating system is turned on.
- **Heating Off**: If the temperature T exceeds the setpoint plus the tolerance ϵ, i.e., $T > T_{\text{set}} + \epsilon$, the heating system is turned off.

  Here, ϵ\epsilon is a small tolerance to prevent frequent switching (hysteresis).

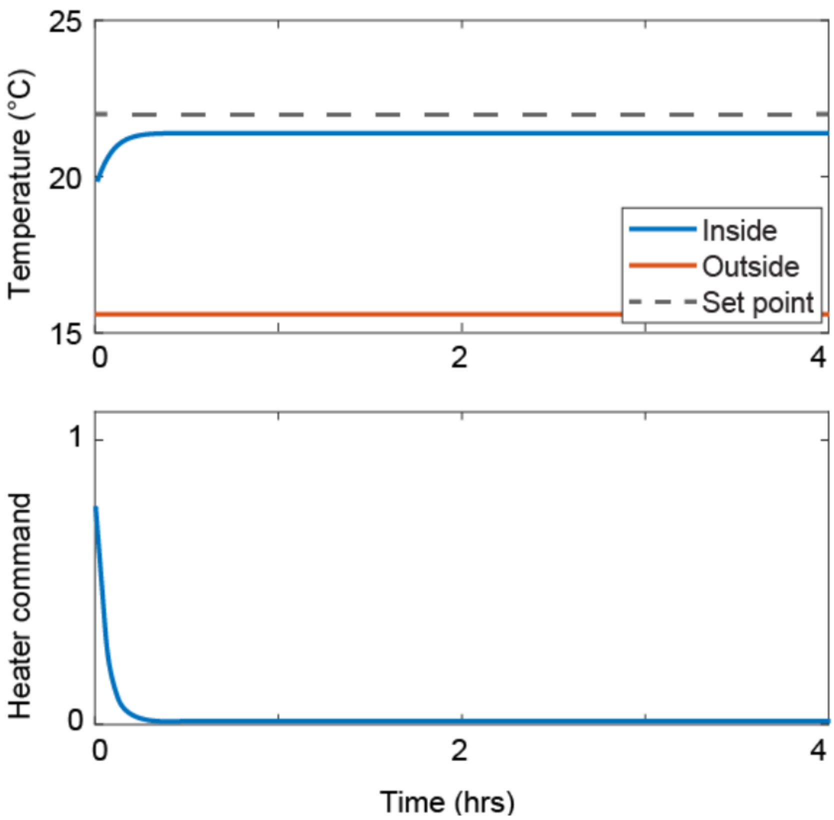
The system was implemented in Simulink to visualize the temperature variations and heater commands over time.



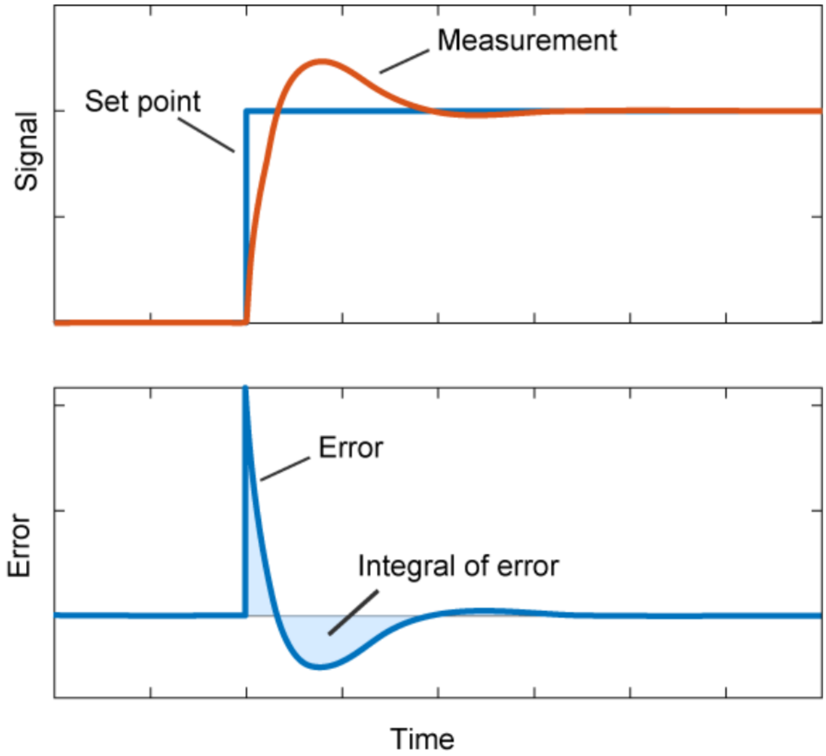
The result, as shown here, is the proper average temperature, but with a lot of fluctuations.



Instead, I specified the controller output as proportional to the error. For example, if the temperature is close to the set point, little heat is required. If the temperature difference is large, though, the heater should be at maximum power.



The result is a much more consistent temperature, but the temperature never quite reaches the set point. To address this, you can also consider the accumulated error over time.



5.

This is done by integrating the error and increasing the control effort if the accumulated error grows. This type of control is aptly named proportional-integral, or PI, control.

6. Methodology

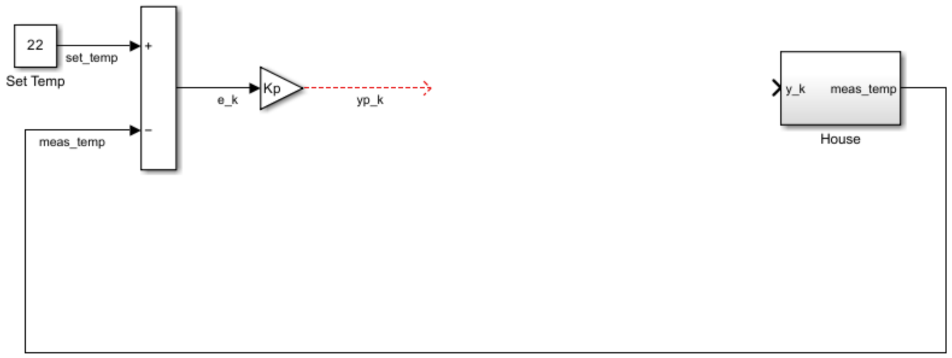
The model was created based on the equation of a discrete PI controller with design variables K_p , K_i, and sample time T_s as shown in the equation below.

$y[k] = y_p[k] + y_i[k]$ .... (1)

$y_p[k] = k_p e[k]$ .... (2)

$y_i[k] = k_i T_s e[k] + y_i[k-1]$ ......(3)

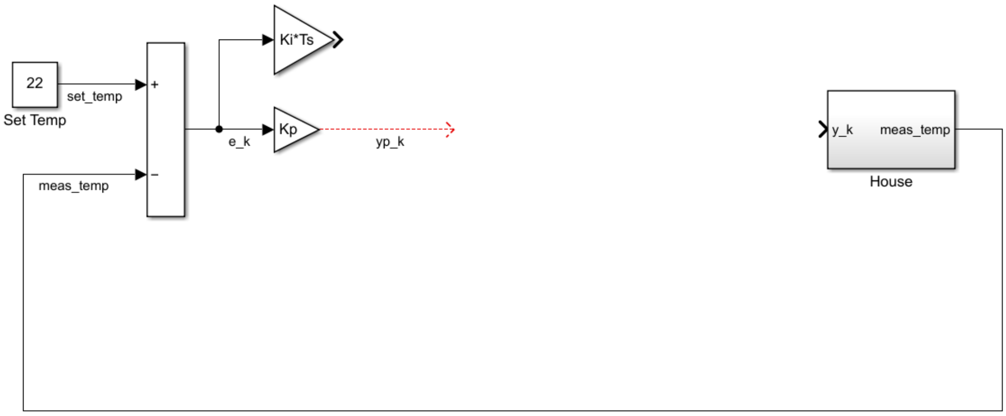
The values of K_p , K_i, and sample time T_s are already defined in MATLAB workspace as 0.20, 0.05 and 0.15 respectively. The initial state of the model is shown below in figure 2.



(Time given in units of hours)

Figure 2: Initial Parameters

As shown in equation (3) above, the first step is to add a gain block to the model and set its value to K_i*T_s and connect e[k] to the gain block. This methodology achieves the first part of equation (3) as shown in figure 3 below.



(Time given in units of hours)

Figure 3: Gain added with predefined design variables

From equation (3) we can deduce that a single unit delay block is needed in the model to represent the integrator term y_i[k-1]. The reason for adding the unit delay block is to delay input by one iteration. Meaning, for every input, the system delivers an output. At this point, it is important to label the input and output parameters of the unit delay block. The unit delay block receives input y_i[k] and outputs y_i[k-1] as shown in figure 4 below.

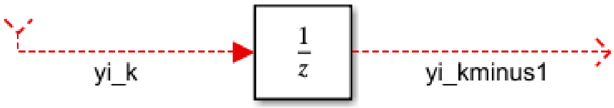


Figure 4: Unit delay block with labeled stubs

The model so far has defined equation 3. The figure 5 below gives a clearer representation.

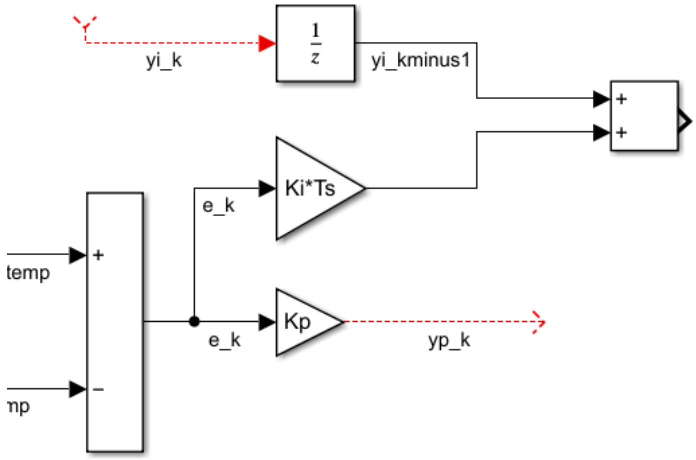


Figure 5: Adding the gain block and unit delay

Connecting $y_i[k]$ to the signal assessment block gives us the output of equation 3.

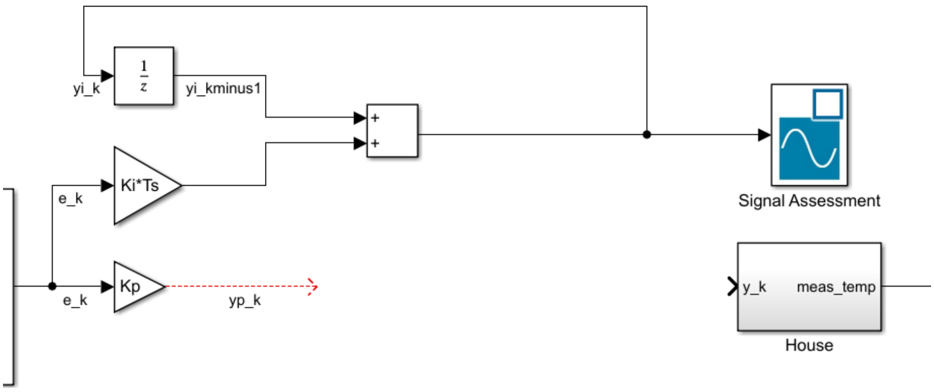


Figure 6: Proportional Term, $y_i[k]$

The graph below shows the result obtained from the model.

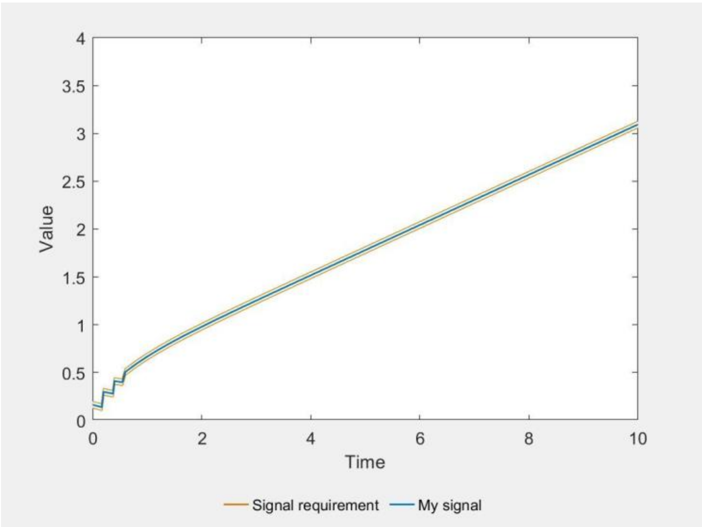
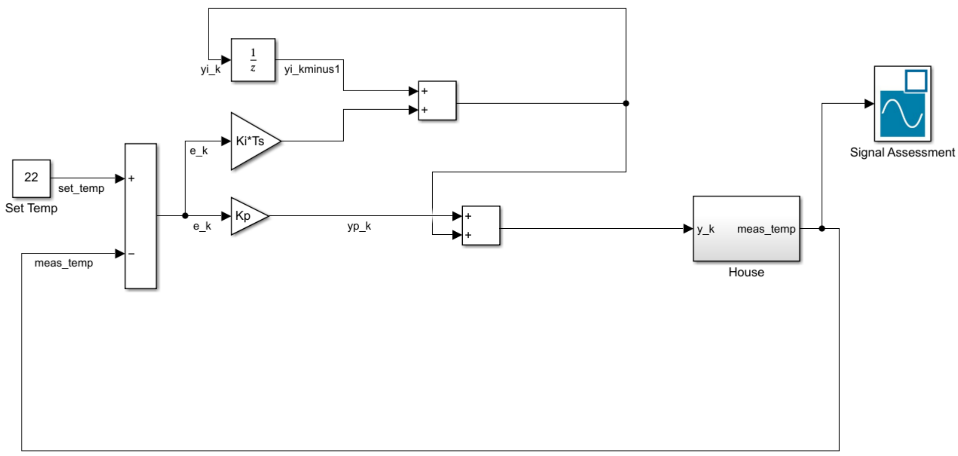


Figure 7: Proportional Term Result

To obtain the output of the controller y[k], we need to add the proportional and integral terms as shown in equation (1) above.



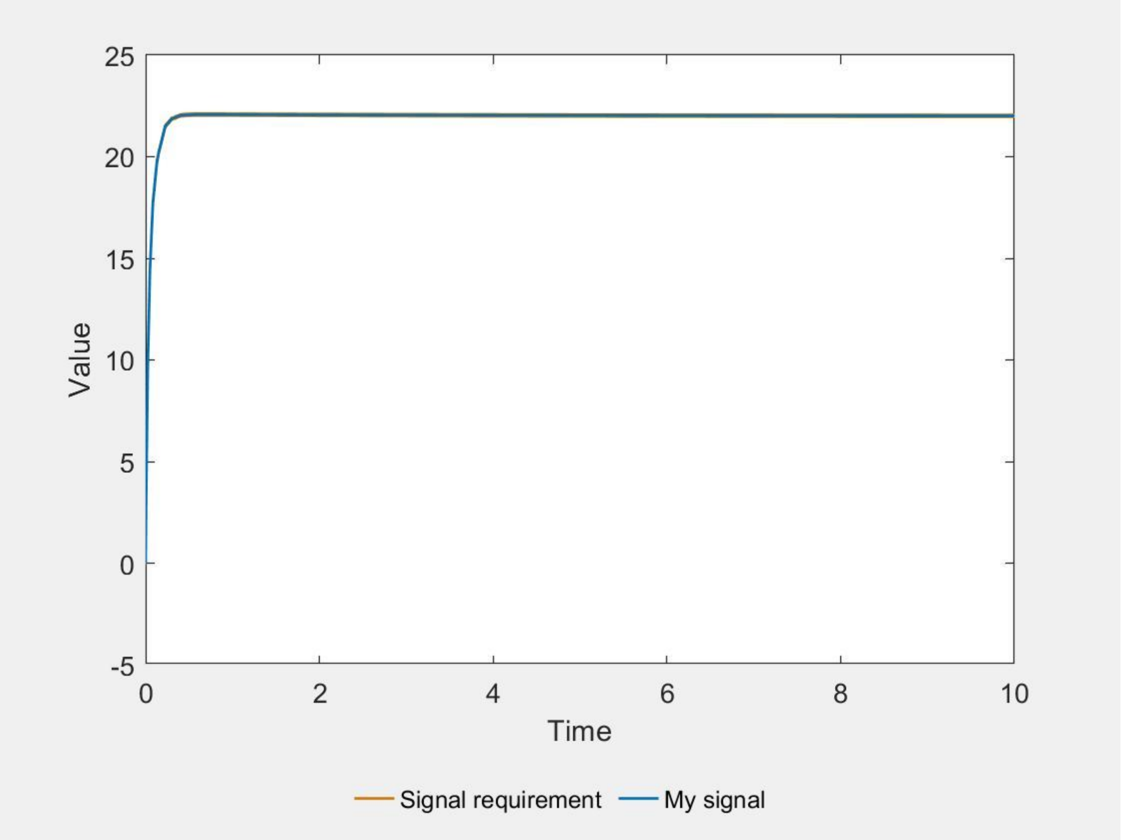
(Time given in units of hours)

Figure 8: Controller Output Model

4. Results

Simulation Outputs

The results of the simulation are shown below:



The image below displays the output of the controller (thermostat). On the y-axis is the temperature reading against time on the x-axis. We can deduct from the result that the temperature is consistent and never quite reaches the set temperature. The heater receives a signal from the controller to regulate the temperature whenever the temperature of the room is approaching the set temperature.

Energy Consumption

The on/off pattern of the heating system provided insights into energy usage. The heating system was active for approximately 30% of the simulation duration, balancing comfort and efficiency.

System Stability

The control system demonstrated stability under varying conditions, including changes in ambient temperature and setpoint adjustments.

Discussion

- The thermostat's simple on/off control strategy proved effective for maintaining comfort in a residential setting. However, more advanced strategies (e.g., PID controllers) could enhance performance and energy efficiency.

- The choice of ϵ\epsilon significantly influenced system performance. A larger ϵ\epsilon reduced switching frequency but increased temperature deviations, while a smaller ϵ\epsilon improved precision but caused more frequent switching.

5. Conclusion

This project successfully demonstrated the design and simulation of a digital thermostat model in Simulink. The first-order cooling equation provided a realistic representation of temperature dynamics, and the on/off control strategy effectively maintained the desired temperature. Future work could explore adaptive control strategies and incorporate real-world factors such as insulation and external heat sources.

6. References

a. Ogata, K. (2010). *Modern Control Engineering*. Prentice Hall.

b. MATLAB Documentation: Numerical Integration Methods. Available at: https://www.mathworks.com

c. Cengel, Y. A., & Boles, M. A. (2015). *Thermodynamics: An Engineering Approach*. McGraw-Hill Education.