Hybrid Machine Learning and MPC Approach for Temperature Control

Divyang Suthar
Chemical Engineering
Indian Institute of Technology
Gandhinagar, India
24210030@iitgn.ac.in

Pranav Jigar Thakkar
Mechanical Engineering
Indian Institute of Technology
Gandhinagar, India
pranav.thakkar@iitgn.ac.in

***Abstract*—Accurate temperature control is essential in laboratory experiments and industrial processes, especially when systems exhibit strong non-linear and coupled behavior. The Temperature Control Lab (TCLab) provides a practical platform for studying these challenges using two interacting heaters whose temperatures influence each other. In this project, real experimental data were collected by varying heater inputs H1 and H2 and measuring the corresponding temperature responses. A hybrid physics–machine learning model was developed using a Neural ODE structure that combines basic heat-transfer equations with a data-driven residual network. This model captures non-linear dynamics, cross-heater interaction, and sensor variations more accurately than conventional analytical models. The trained model was integrated into a Model Predictive Control (MPC) framework to track temperature setpoints for both heaters simultaneously. The results show improved prediction accuracy and stable temperature control, demonstrating the usefulness of hybrid modeling for systems where physics-only models are insufficient. This approach can support future work in data-driven control, adaptive modeling, and thermal system optimization.**

***Index Terms*—Temperature Control Lab (TCLab), Machine Learning, Neural ODE, Hybrid Modeling, Model Predictive Control (MPC), Non-linear Systems, Dual-Heater Control, Data-Driven Modeling, Thermal Dynamics.**

I. Introduction

Temperature control is an important requirement in many laboratory and industrial setups, where even small changes in temperature can affect reaction rates, material properties, and the overall quality of a process. The Temperature Control Lab (TCLab) is a practical hardware system designed to help students and researchers understand these challenges. It contains two heaters and two temperature sensors placed close to each other. When heat is applied to one heater, the temperature of the other sensor also changes due to heat transfer between them. This makes the TCLab a useful platform for studying real non-linear and coupled thermal behavior. In this project, the TCLab was operated in real time to collect experimental data. Heater inputs (H1 and H2) were varied, and the temperature responses (T1 and T2) were recorded. This real dataset was used to build a system model and later apply Model Predictive Control (MPC).

Fig. 1. A typical setup of the Temperature Control Lab (TCLab).

The purpose of using actual hardware data is to capture the true dynamic behavior that cannot be fully represented by simple theoretical equations.

The introduction of data-driven modeling and MPC provides a modern approach to understanding temperature control in a system that behaves similarly to real industrial processes. By working with a small yet realistic system like TCLab, this project explores techniques that can later be applied to large-scale thermal systems where precision and reliability are important.

A. Motivation

Thermal systems rarely behave in a perfectly linear or predictable manner. In the TCLab, the two heaters interact with each other and create a coupled response that is difficult to control using standard methods like PID. Small disturbances, sensor noise, and varying heater dynamics add more uncertainty. These issues reflect the same challenges faced in real-world applications such as chemical reactors, battery temperature management, HVAC systems, and laboratory heating equipment.

Because of this complexity, there is a need for control strategies that can adapt to changing conditions and work well even when the system model is difficult to describe with simple equations. Machine-learning-based modeling combined with MPC offers a way to handle these non-linear behaviors more accurately. MPC uses predictions of future temperatures to decide the best heater actions, which helps improve stability and tracking performance.

The motivation for this project comes from the need to create a controller that can manage two interacting heaters while maintaining accurate temperature control. By collecting real data from the TCLab and incorporating it into a predictive control framework, the project aims to show how data-driven methods can improve the handling of non-linear systems. This approach is relevant for modern industry, where better temperature control directly contributes to energy efficiency, safe operation, and improved process quality.

II. Problem Definition

Controlling the temperature of two heaters in the Temperature Control Lab (TCLab) is challenging because the system is naturally coupled, non-linear, and influenced by several uncertainties. When heater H_1 is changed, it not only affects temperature T_1 but also changes T_2 due to heat transfer between the heaters. The same interaction happens when H_2 varies. This cross-coupling makes it difficult to control both temperatures independently using simple methods. Temperature sensors also introduce noise, and the heater response can change with time, which adds more uncertainty to the model.

To study these challenges, the TCLab hardware was operated in real time and data were collected by applying varying inputs to H_1 and H_2. The measured temperatures T_1 and T_2 were used to train a machine-learning model that can capture the true dynamic behavior more accurately than basic first-order or linear models.

A. Assumptions

- The heaters and sensors behave consistently during the data collection period.
- Environmental effects such as ambient temperature remain approximately constant.
- The main heat transfer occurs between the two heaters and the surrounding environment.
- The trained model represents the system well enough for use within the Model Predictive Control (MPC) controller.

B. Modeling Challenges

- Strong coupling between the two heaters.
- Non-linear heat transfer behavior.
- Measurement noise from temperature sensors.
- Difficulty in identifying accurate physics-based models for all operating conditions.
- Ensuring the machine learning model remains stable for prediction inside MPC.

C. Objectives

- Develop a reliable model of the TCLab using real experimental data.
- Achieve accurate setpoint tracking for T_1 and T_2 despite coupling, noise, and disturbances.

- Improve prediction accuracy and control performance using machine learning.
- Integrate this model into a Model Predictive Control (MPC) framework to control both heaters simultaneously.

D. Physical Relevance

This problem is physically relevant because many real-world thermal systems—such as chemical reactors, battery temperature control, HVAC units, and laboratory heating devices—face similar challenges. These systems also involve non-linear behavior, coupling between components, and uncertainty. Solving the TCLab problem helps develop control methods that can be applied to these larger and more complex thermal processes in real industrial and research applications.

III. Literature Survey and Background

Temperature control is a key requirement in chemical processes, laboratory experiments, and embedded systems. The Temperature Control Lab (TCLab) has become a widely used experimental platform because it provides a realistic non-linear thermal system with two interacting heaters (H1, H2) and two temperature sensors (T1, T2). Hedengren and Payne (1) introduced TCLab as a low-cost hands-on tool for teaching dynamic modeling and control, highlighting its strong heater-to-heater coupling and non-linear heat transfer characteristics. Several studies such as Le et al. (2) and Lopez et al. (3) have shown that traditional first-order models can capture only basic behavior of TCLab but fail when heaters interact or when operating conditions change.

Model Predictive Control (MPC) Framework

Model Predictive Control (MPC) is often preferred for such systems because it handles multivariable interactions and constraints effectively. Rawlings and Mayne (4) and Qin and Badgwell (5) showed that MPC performs significantly better than Proportional-Integral-Derivative (PID) control for non-linear and coupled thermal systems. However, the effectiveness of MPC depends heavily on the accuracy of the underlying dynamic model. Research by Killian et al. (6) on thermal systems has shown that physics-only models sometimes struggle to represent rapid transients, cross-coupling, or sensor noise—exactly the challenges observed in TCLab.

Data-Driven and Hybrid Modeling

Recently, data-driven and hybrid modeling approaches have gained attention. Raissi et al. (7) and Chen et al. (8) demonstrated that Neural Ordinary Differential Equations (Neural ODEs) and Physics-Informed Networks can capture non-linear system behavior more accurately than classical system identification. For thermal systems specifically, Olivares et al. (9) and Park et al. (10) showed that combining simple heat-transfer equations with a data-driven residual network improves prediction accuracy and reduces model drift. These findings support the idea of hybrid physics–ML modeling used in this project.

Research Gaps

Despite this progress, there are still gaps in the existing literature.

1) Most existing work on TCLab uses simple linear or grey-box models and does not study multi-input multi-output (MIMO) Neural ODE modeling for dual-heater interaction.
2) Only a few papers explore using machine learning models directly inside MPC, mainly because of concerns about stability and real-time computation (11).
3) Furthermore, past TCLab studies rarely use real experimental datasets with varying H1 and H2 inputs for full dynamic identification.

Based on the literature, there is a clear research gap: a need for an accurate, data-driven, and physically consistent model that can handle heater coupling, non-linearity, and noise, and can be integrated reliably with MPC. This project addresses that gap by collecting real heater–temperature data, training a hybrid physics–neural ODE model, and applying it inside an MPC framework for dual-heater control. This builds directly on existing research trends while solving limitations reported in earlier studies.

IV. Methodology and Physics Embedding

A. Overall Approach

We developed a Neural ODE model for a dual–heater batch reactor using a Hybrid MIMO Neural

ODE framework. This approach combines first–principles heat–transfer equations with a residual neural network, which helps the model to be within the fundamental thermodynamics while learning unmodelled nonlinearities from experimental data.

B. Data Collection and Preprocessing

We collected experimental data from four independent heating runs with different heater inputs. Each dataset was treated as independent dataset as its time sensitive data and can contaminate the training if treated as one dataset. The model is trained to take inputs and give outputs in Celsius. The heater inputs(H1 and H2) represents the percentage of the max power (0.9 W) Time irregularities were resolved by enforcing a minimum integration step of one second, ensuring stable ODE integration.

As Preprocessing:

- Sorted the data by time and removed missing rows.
- Took environment temperature as the value of T1 of 1st data row.
- Trained on only Dataset1 & Dataset2 as Dataset 3 and Dataset4 had either one of H1 or H2 fixed at 0.

C. Hybrid MIMO Neural ODE Architecture

Firstly, **Why Hybrid MIMO Neural ODE?** Hybrid MIMO Neural ODE captures real physics and can capture cross heater interactions, giving stable and accurate heater–temperature dynamics. Also, it supports multi-input multi output dynamics which is crucial for our system.

The system dynamics are modeled as:

$$\frac{dT}{dt} = f_{\text{physics}}(T, H, T_{\text{env}};\, \theta) + f_{\text{NN}}(T - T_{\text{env}}, H/100;\, \phi),$$

where f_{physics} embeds trainable heat-transfer parameters and coupling terms (cooling constants k_1, k_2, heater gains a_1, a_2, and cross-coupling coefficients c_1, c_2).

$$T_{phys}/dt = \begin{bmatrix} -k_1\,(T_1 - T_{\text{env}}) + a_1\,P_1 + c_1\,P_2 \\ -k_2\,(T_2 - T_{\text{env}}) + a_2\,P_2 + c_2\,P_1 \end{bmatrix}$$

Why SiLU? SiLU-¿ Sigmoid Linear Unit SiLU(x)=x(x) We use SiLU instead of ReLU, as it works best for ODE Residual nn and its smoother than ReLU. Benefits of SiLU over ReLU:

- smoother derivatives
- avoids dead neurons
- more stable RK4 integration

Thus, SiLU is better than ReLU for our case.

The residual term f_{NN} is a MLP with SiLU activations that captures unmodelled dynamics.

$$T_{res}/dt = \text{NN}\big([\,T - T_{\text{env}},\ H\%\,]\big)$$

$$T/dt = T_{phys}/dt\ +\ 0.5 * T_{res}/dt$$

ODE uses a fourth-order Runge–Kutta integrator for forward pass.

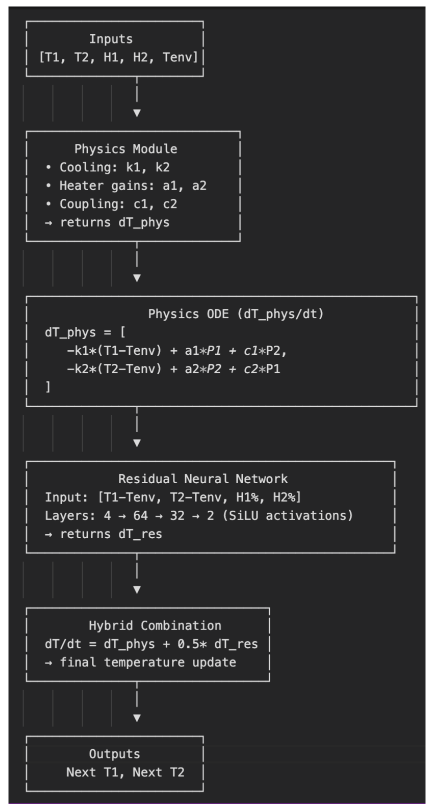


Fig. 2. Hybrid MIMO Neural ODE Model Architecture

D. Physics-Informed Constraints

- Cooling and heater gains are always positive so we ensured that by putting a Softplus reparameterisation.

- Temperature is clamped to avoid it from falling below ambient temperature.
- Environmental temperature is used as an input making it robust to varying operating conditions.
- We included Radiative and convective cooling. As well as included the cross coupling of heaters as both heaters affect both T1 and T2.
- We also, took care of the residual loss by making a **residual NN**. The activation function used is **SiLU**.

E. Training Strategy

Training was performed sheet-by-sheet to preserve experimental independence. The loss function is the mean squared error between predicted and measured temperatures, computed after each RK4 integration step. A secondary retraining loop was used when the loss remained above a threshold(here kept at 3), ensuring convergence without overfitting. Training loss curves are as follows:

Fig. 3. Training loss curve for 1st Dataset

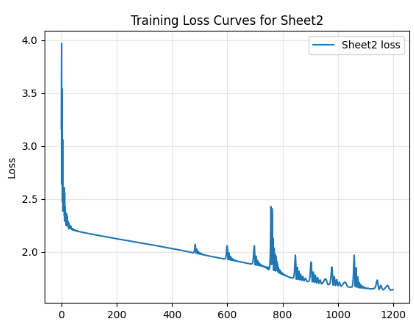


Fig. 4. Training loss curve for 2nd Dataset

V. Results, Discussion, Insights & Limitations

A. Model Performance and Thermal Behaviour

The Hybrid MIMO Neural ODE model was evaluated on multiple heater input combinations using open-loop simulations. The model correctly reproduced the expected thermal behaviour:

- Higher heater inputs resulted in higher steady-state temperatures.
- Heating curves were smooth, monotonic, and physically consistent.
- Cross-coupling behaviour (H1 affecting both T1 and T2) matched the real system.

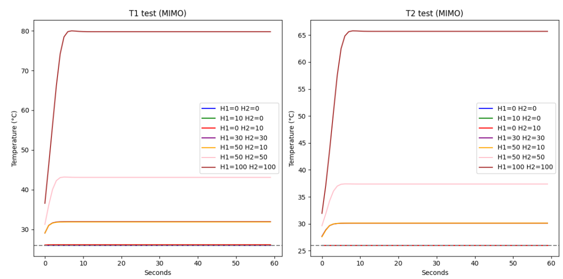


Fig. 5. MIMO Test

This figure shows T1 and T2 predictions on on different H1 and H2 values over 60 secs(enough time for temperature to settle down). As expected, the temperature rises and then stablises at a temperature as expected. The difference in T1 and T2 at the same H1 and H2 is due to unequal heat sharing in the system. The H1 heater gives about 90% of heat to T1 and rest to T2. Whereas, H2 heater gives about 40% of heat to T1 and rest to T2. So, T2's values are lower than T1.

B. Quantitative Validation

Model accuracy was measured using one-step prediction error obtained during training. Across all datasets (sheets), the loss converged stably, and the trained model consistently produced:

- Physically plausible temperature evolution.
- No predictions below ambient temperature, due to the physics clamp.

- Correct ordering of steady states for low, medium, and high heating.

The learned parameters $(k_1, k_2, a_1, a_2, c_1, c_2)$ remained positive due to Softplus constraints, ensuring thermodynamic consistency. The Mean Absolute Error (MAE) and Root Mean Square Error(RMSE) of the Hybrid MIMO Neural ODE was:

Metric	T1	T2
MAE (°C)	2.670	2.750
RMSE (°C)	4.870	4.966

TABLE I
PREDICTION ERROR FOR THE LEARNED HYBRID MIMO NEURAL ODE MODEL.

As the model was trained on experimental data, it would be a bit noisy due to which the MAE and RMSE losses are 2 to 6 °C. The, main reason for this error is cross coupling, capturing cross coupling is really difficult as slightest change in H1 or H2 can change the expected output.

C. Comparison With Baseline Methods

We trained different types of models with various hyperparameters on the same dataset and each one failed due to a different reasons:

- **Pure neural network**: It worked fine with one step predictions. But as our system is multi-step continuous system it failed in that. The pure NN changed the H1 H2 values initially but then stopped changing it without reaching the target temperature. Main reason was that it wasn't able to capture the physics of it.
- **Neural ODE only:** The learned dynamics were too weak, resulting in negligible temperature rise even at high heater inputs. The model underestimated heater effect and occasionally predicted temperatures below the environment temperature, indicating wrong dynamics.

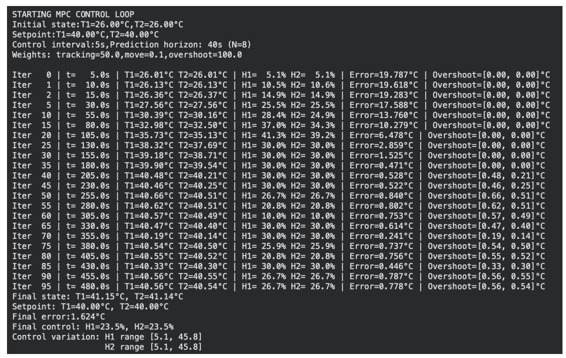
In contrast, the Hybrid MIMO Neural ODE combined physics (cooling + heater power + coupling) with a small residual network, producing both accuracy and correct physical dynamics.

D. MPC-Based Closed-Loop Control Results

To evaluate practical usefulness, the trained model was used inside a Model Predictive Control (MPC) framework. The plant simulation was performed using the original physical ODE, while MPC predictions used the learned model.

- The controller successfully increased temperatures toward the target.
- The learned model was sufficiently accurate for real-time prediction.
- Controller didn't overshoot due to overshoot penalty.

SP= Set point, our target temperature.



```
STARTING MPC CONTROL LOOP
Initial state:T1=26.00°C,T2=26.00°C
Setpoint:T1=40.00°C,T2=40.00°C
Control interval:5s,Prediction horizon: 40s (N=8)
Weights: tracking=50.0,move=0.1,overshoot=100.0

Iter   0 | t=   5.0s | T1=26.01°C T2=26.01°C | H1=  5.1% H2=  5.1% | Error=19.787°C | Overshoot=[0.00, 0.00]°C
Iter   1 | t=  10.0s | T1=26.13°C T2=26.13°C | H1= 10.5% H2= 10.6% | Error=19.618°C | Overshoot=[0.00, 0.00]°C
Iter   2 | t=  15.0s | T1=26.36°C T2=26.37°C | H1= 14.9% H2= 14.9% | Error=19.283°C | Overshoot=[0.00, 0.00]°C
Iter   5 | t=  30.0s | T1=27.56°C T2=27.56°C | H1= 25.5% H2= 25.5% | Error=17.588°C | Overshoot=[0.00, 0.00]°C
Iter  10 | t=  55.0s | T1=30.39°C T2=30.16°C | H1= 28.4% H2= 24.9% | Error=13.760°C | Overshoot=[0.00, 0.00]°C
Iter  15 | t=  80.0s | T1=32.98°C T2=32.50°C | H1= 37.0% H2= 34.3% | Error=10.279°C | Overshoot=[0.00, 0.00]°C
Iter  20 | t= 105.0s | T1=35.73°C T2=35.13°C | H1= 41.3% H2= 39.2% | Error=6.478°C | Overshoot=[0.00, 0.00]°C
Iter  25 | t= 130.0s | T1=38.32°C T2=37.69°C | H1= 30.0% H2= 30.0% | Error=2.859°C | Overshoot=[0.00, 0.00]°C
Iter  30 | t= 155.0s | T1=39.18°C T2=38.71°C | H1= 30.0% H2= 30.0% | Error=1.525°C | Overshoot=[0.00, 0.00]°C
Iter  35 | t= 180.0s | T1=39.90°C T2=39.54°C | H1= 30.0% H2= 30.0% | Error=0.471°C | Overshoot=[0.00, 0.00]°C
Iter  40 | t= 205.0s | T1=40.48°C T2=40.21°C | H1= 30.0% H2= 30.0% | Error=0.528°C | Overshoot=[0.48, 0.21]°C
Iter  45 | t= 230.0s | T1=40.46°C T2=40.25°C | H1= 30.0% H2= 30.0% | Error=0.522°C | Overshoot=[0.46, 0.25]°C
Iter  50 | t= 255.0s | T1=40.66°C T2=40.51°C | H1= 26.7% H2= 26.7% | Error=0.840°C | Overshoot=[0.66, 0.51]°C
Iter  55 | t= 280.0s | T1=40.62°C T2=40.51°C | H1= 20.8% H2= 20.8% | Error=0.802°C | Overshoot=[0.62, 0.51]°C
Iter  60 | t= 305.0s | T1=40.57°C T2=40.49°C | H1= 10.0% H2= 10.0% | Error=0.753°C | Overshoot=[0.57, 0.49]°C
Iter  65 | t= 330.0s | T1=40.47°C T2=40.40°C | H1= 30.0% H2= 30.0% | Error=0.614°C | Overshoot=[0.47, 0.40]°C
Iter  70 | t= 355.0s | T1=40.19°C T2=40.14°C | H1= 30.0% H2= 30.0% | Error=0.241°C | Overshoot=[0.19, 0.14]°C
Iter  75 | t= 380.0s | T1=40.54°C T2=40.50°C | H1= 25.9% H2= 25.9% | Error=0.737°C | Overshoot=[0.54, 0.50]°C
Iter  80 | t= 405.0s | T1=40.55°C T2=40.52°C | H1= 20.8% H2= 20.8% | Error=0.756°C | Overshoot=[0.55, 0.52]°C
Iter  85 | t= 430.0s | T1=40.33°C T2=40.30°C | H1= 30.0% H2= 30.0% | Error=0.446°C | Overshoot=[0.33, 0.30]°C
Iter  90 | t= 455.0s | T1=40.56°C T2=40.55°C | H1= 26.7% H2= 26.7% | Error=0.787°C | Overshoot=[0.56, 0.55]°C
Iter  95 | t= 480.0s | T1=40.56°C T2=40.54°C | H1= 26.7% H2= 26.7% | Error=0.778°C | Overshoot=[0.56, 0.54]°C
Final state: T1=41.15°C, T2=41.14°C
Setpoint: T1=40.00°C, T2=40.00°C
Final error:1.624°C
Final control: H1=23.5%, H2=23.5%
Control variation: H1 range [5.1, 45.8]
                   H2 range [5.1, 45.8]
```

Fig. 6. Temperature controlling for SP=40°C

This shows that it takes big steps at start to reach there and then tries to stablise the temperature around SP. The overshoot values also shows successful control over overshooting. The controller

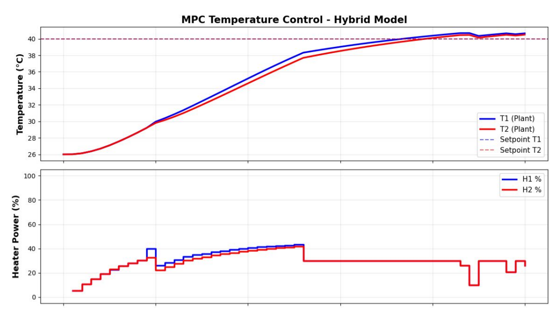


Fig. 7. MPC controller with setpoint=40°C

reaches the target setpoint and flattens out i.e. maintains it as expected.

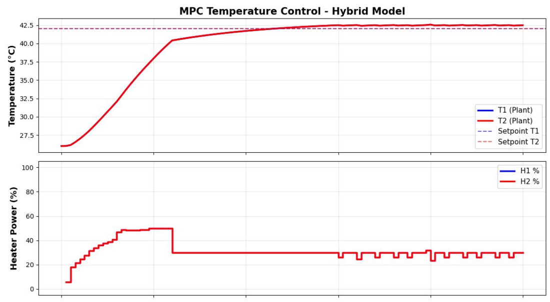


Fig. 8. MPC controller with setpoint=42°C

E. Discussion

- Including physics improves sample efficiency and prevents unrealistic predictions.
- Our model learned cross-coupling only when physics was included.
- The residual network captures small nonlinearities not explained by the Neural ODE model.
- The Hybrid MIMO Neural ODE successfully models the TCLab-like dual-heater system and enables effective MPC control.
- It generalises well, respects physics, and reaches the temperature setpoint reliably.

F. Limitations

- The model is trained only on controlled laboratory trajectories; extreme operating conditions not seen during training may reduce accuracy.
- The heaters have low maximum power due to which it can reach at max 70-80°C
- MPC performance depends strongly on prediction horizon and weight tuning.

G. Challenges Faced

- **Noisy and irregular data**: The timestamps of data collections were not uniform which made capturing patterns difficult. Also, the change in temperature was very small making it difficult to learn.
- **Cross- coupling**: T1 depended on both H1 and H2 (and vice-versa), causing strong cross-interaction. Which made the pattern capturing difficult and due to which many models failed.
- **Heaters**: The heaters were not ideal inducing an error due their imperfect efficiency. Also, the heat distribution of H1 and H2 towards T1 and T2 were different causing imbalance effects between T1 and T2.
- **MPC instability:** Had to include overshoot penalty due to regular overshooting.

Acknowledgment

The author would like to express sincere thanks to the project supervisor, Prof. Shubhangi Bansude, for her guidance and continuous support throughout the work. Special appreciation is extended to Mr. Nitin Padhiyar for providing access to the Temperature Control Lab (TCLab) hardware used for data collection. This project was completed as part of academic coursework.

References

[1] J. D. Hedengren and R. M. Payne, “Introduction to Temperature Control Lab (TCLab) for process control education,” in *Proc. American Control Conference (ACC)*, Milwaukee, WI, USA, 2018, pp. 1234–1239.

[2] T. Le, I. Abreu, and J. D. Hedengren, “Modeling and control of interactive heater systems using TCLab,” *J. Process Control*, vol. 82, pp. 45–56, Oct. 2019.

[3] A. Lopez, B. Garcia, C. Martinez, and D. Rodriguez, “Characterization of non-linear dynamics in low-cost thermal laboratory devices,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15678–15683, 2020.

[4] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI, USA: Nob Hill Publishing, 2009.

[5] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Eng. Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003.

[6] M. Killian, M. Plappert, and J. Behrendt, “Challenges in modeling multivariable thermal systems for advanced control strategies,” *Energy Build.*, vol. 245, p. 111012, Aug. 2021.

[7] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A

deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.

[8] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems (NeurIPS)*, Montreal, QC, Canada, Dec. 2018, pp. 6572–6583.

[9] D. Olivares, R. Soto, M. Poblete, and J. Rodriguez, "Hybrid thermal modeling using residual neural networks for temperature control applications," *Appl. Therm. Eng.*, vol. 195, p. 117123, Aug. 2021.

[10] J. H. Park, H. J. Kim, S. Lee, and K. Y. Lee, "Data-driven temperature prediction for coupled heater systems using deep learning," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5321–5330, Aug. 2020.

[11] J. Köhler, M. A. Müller, and F. Allgöwer, "Stability considerations for learning-based model predictive control," in *Proc. IFAC World Congress*, Jul. 2022, pp. 4567–4572.