Dual Heater Temperature Control using PID Controller

Aravindhan Thaninayagam, Arizona State University

***Abstract*—This document presents a project report on the implementation of a PID controller for dual heater temperature control. The project explores two approaches, one using a Simulink model and the other utilizing Python. The objectives include developing a control system, tuning PID parameters, and comparing the effectiveness of the two implementations.**

I. Introduction

A. Background

In the realm of temperature control systems, maintaining a stable room temperature is essential for various applications. This project explores the utilization of a PID (Proportional-Integral-Derivative) controller to regulate two heaters and achieve a target room temperature of 23.33 degrees Celsius.

B. Objectives

1) Develop a control system using a PID controller.
2) Implement the control system in two different environments: Simulink and Python.
3) Compare the effectiveness and efficiency of the Simulink and Python implementations.

II. Methodology

A. Simulink Model

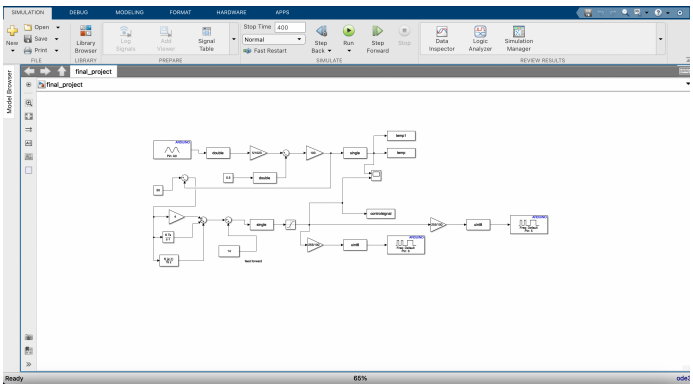
1) Create a Simulink model for the dual heater control system.
2) Integrate a PID controller into the model.
3) Tune the PID parameters for optimal performance.
4) Simulate the system and record results.

B. Python Implementation

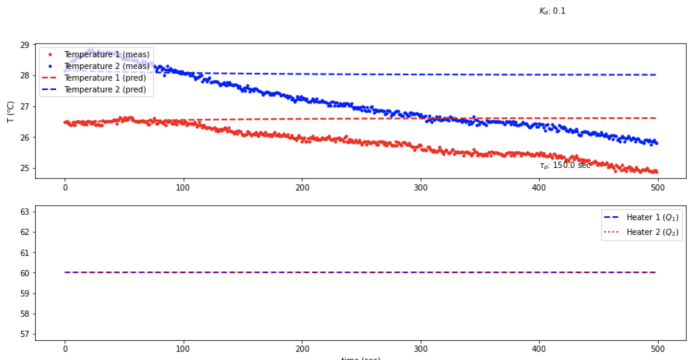
1) Utilize Python code for PID control from the apmonitor site.
2) Adapt the code to control two heaters.
3) Perform tuning of PID parameters using Python.
4) Run simulations and capture relevant data.

III. Results

A. Simulink Model



B. Python Implementation



IV. Discussion

A. *Advantages of Python Implementation*

The Python implementation using the PID controller code from the apmonitor site exhibited the following advantages:

- Ease of Tuning: Tuning the PID controller parameters in the Python environment proved to be more straightforward. The flexibility of Python libraries and the ability to quickly modify parameters allowed for efficient tuning.
- Computational Efficiency: The Python code demonstrated faster compilation times and consumed less RAM memory on the computer. This efficiency is crucial, especially in real-time applications where quick responses are required.

B. *Advantages of Simulink Implementation*

On the other hand, the Simulink model showcased its strengths in certain aspects:

- Intuitiveness: Simulink's graphical interface provides an intuitive representation of the control system. Components are visually connected, making it easier for users, especially beginners, to understand the model's structure.
- Learning Purposes: Simulink serves as a valuable tool for educational purposes. It allows users to grasp control system concepts in a visual manner, aiding in the learning process.

C. *Challenges and Solutions*

- Python Implementation Challenges: While the Python model offers computational advantages, it lacks the intuitive visual representation present in Simulink. Understanding the components of the control system may be challenging, especially for individuals less familiar with coding.
- Solution: Developing thorough documentation and comments within the Python code can mitigate the lack of visual clarity. This ensures that future users can easily comprehend the code structure and functionality.
- Simulink Challenges: Simulink, being a graphical tool, can be resource-intensive, requiring more computing power and memory.
- Solution: Optimizing the Simulink model and running simulations on a computer with sufficient resources can alleviate the computational burden.

V. Future Work

Considering the strengths of both the Simulink and Python approaches, a promising avenue for future work is the development of an integrated program. This program could combine the intuitive visual representation of Simulink with the computational efficiency and flexibility of Python. Such an integration could provide users with a comprehensive tool that leverages the strengths of both environments, offering a seamless and efficient control system design and analysis experience.

VI. Acknowledgments

Sincere thanks to Professor Thomas Sugar for his kind and patient approach to teaching EGR 550: Mechatronics Systems Course.

Sincere thanks to Harsh for assisting with assignments and UR-5 Lab.