Multi-room Heating Controller

Jane Wu
Department of Computer Science
Vanderbilt University
Nashville, TN
jiayi.wu@vanderbilt.edu

Anda Liang
Department of Computer Science
Vanderbilt University
Nashville, TN
anda.liang@jpl.nasa.gov

***Abstract*—This report presents the design and simulation of a multi-room heating system using Simulink, aiming to enhance energy efficiency and thermal comfort in residential settings. The project initiates a comprehensive analysis of thermal dynamics in multi-room environments, identifying critical factors influencing heat distribution and retention among different rooms. Employing Simulink's robust system simulation capabilities, a model incorporating these dynamics was developed. The system features individual room control for optimized heating, leveraging feedback loops and differential equations to adapt to varying environmental conditions.**

Simulation results demonstrate the system's effectiveness in maintaining desired temperature ranges while satisfying proposed safety requirements. The model also highlights areas for potential integration with other controller systems to achieve more sustainable heating solutions.

***Index Terms*—Simulink, home automation, control systems**

I. Introduction

In the evolving landscape of building automation and energy management, the optimization of heating systems in residential environments has emerged as a pivotal challenge. The quest for energy efficiency and improved thermal comfort has driven the adoption of advanced control systems in home heating. This report introduces a multi-room heating system, designed mainly using MATLAB's Simulink, a platform renowned for its robust modeling capabilities.

The motivation for this project stems from the increasing need for energy-efficient solutions and comfort in residential heating. Traditional heating systems often operate under a 'one-size-fits-all' approach, lacking the flexibility to address the diverse thermal requirements of different rooms within a home. This not only leads to energy waste but also compromises occupant comfort. The proposed multi-room heating system is a first step to address these issues by employing individual room control, thus ensuring comfortable living conditions.

Utilizing Simulink's dynamic simulation environment, this project models the thermal dynamics of a multi-room residence. The model integrates feedback loops to adaptively regulate the temperature in each room based on various factors, including external weather conditions, room conditions (thermal dynamics), and individual vent preferences. This approach not only enhances energy efficiency but also ensures the occupants' comfort of residential heating.

This report unfolds the development process of the multi-room heating system, detailing the methodologies employed, the simulation setup, and the ensuing results. The implications of these findings extend beyond mere energy savings, hinting at potential future developments in smart home technologies and sustainable living. Through this project, we aim to construct a meaningful solution to the ongoing discourse in energy-efficient residential heating, highlighting the robustness and efficiency of our model for sustainability and comfort.

II. Problem Description

To explore the problem of multi-room heating, in this project we will consider the following system based on the scenario described in the project description provided by Dr. Abhishek Dubey[1]:

Suppose there is a house with four rooms equipped with a central heating system, and each room has a single vent for heat to pass through and heat up the room. The outside temperature is lower than room temperature, so the vents need to be constantly opened to heat the rooms up. There is limited heating power so only 2 vents are allowed to open at the same time in the house. The room dynamics is defined in Equation 1:

$$\dot{x}_i = c_i h_i + b_i(u - x_i) + \sum_j a_{i,j}(x_j - x_i), i \in \{1, 2, 3, 4\} \quad (1)$$

with constants $a_{i,j}$, b_i, c_i. $A = [a_{i,j}] \in \mathbb{R}^{4\times4}$, A is a matrix defining how heat transfer factors between rooms. If $a_{i,j} = 0$, it means i, j refers to the same room or i, j are not adjacent; otherwise, i, j are adjacent. The constants b_i in a vector $\mathbf{b} = [b_i] \in \mathbb{R}^4$ represent some heat transfer factor that defines how heat from outside transfers to room i. The constants c_i in a vector $\mathbf{c} = [c_i] \in \mathbb{R}^4$ that defines how some heat transfer factor from the vent if it is open. The initial temperature of each room will be represented as a vector $\mathbf{x}_0 \in \mathbb{R}^4$.

Besides the constants, $h_i \in \{0, 1\}$ represents the open/close status of the vent in the room: h_i is 0 if there is a vent closed, $h_i = 1$ if the vent is open. And $u \in \mathbb{R}$ represents the outside temperature.

The goal is to design a controller, in the context of this problem it is likely a thermostat, that will control the status of the vents in each room to make sure all the room temperatures stay within a reasonable range.

[1]https://github.com/JaneWu423/CS6376-FinalProject/main/description.pdf

III. Formal Specification

A. Assumptions

Before starting the design process of this multi-room heating system, some assumptions need to be made. Below are some higher-level assumptions about the overall system:

- The house has four rooms, each with a single identical vent that can be closed and opened by the designed controller with no delays.
- The heating system always functions properly, and vents delivering heat to the rooms have no consumption of heat.
- The expected range of temperature in the rooms are between 15 and 20 degree Celsius. The initial temperature is also within the given range.
- Outside temperature remained constant over time at 6 degrees Celsius to simplify the problem.
- The heater transfer factors used in calculating the dynamics of the rooms are also set constant and do not change over time.

Specific assumptions about each component of the system will be covered later when describing each of them in detail.

B. Requirement Specification

For the proposed system, we have also proposed both safety requirements and liveness requirements. For safety:

- The temperature in all rooms is always between a given threshold, for this specific problem, it should be between 15-20 degrees.
- There are at most two vents open at the same time.

We will a Safety Monitor to ensure our system always satisfies the proposed safety requirements. For liveness, each room should have its vent open at once. This keeps the system from stalling in some states. We have not specially designed any components to check on the liveness requirements, but it can be visually inspected by looking at the simulation results of the vent status updates. If one entry of the vent status remains 0 throughout, it likely means the liveness requirement is not satisfied.

C. Interfaces

There are three main components in the system, which are the multi-room system that models the house with four rooms, the controller that will be used to control the vents opening for the room systems, and eventually the safety monitor that shows the current state of the other two components. This system structure in shown in Figure 1. Since they are designed for different purposes, they have different inputs and outputs, and we will discuss them separately.

1) Multi-room System: For the multi-room system, it will have two inputs: outside temperature $u \in \mathbb{R}$ and vent status $h \in \mathbb{R}^4$. As we specified in the assumption, for this problem, we will make the outside temperature a constant; however, in most real-life cases, the temperature outside should be a separate continuous time system, so for future extensibility of our current design, we will make the outside temperature an input to also allow input from other continuous-time systems. The updated vent status will be generated by the controller, which will update the vent status in each room taking into account the room temperature in the previous step. The default initial status of the vents will be closed in this problem.

There will be only one output from this system, which is the current temperature of the rooms $x \in \mathbb{R}^4$. This will become input to the controller and also the safety monitor.

2) Controller: The controller will take one input $x \in \mathbb{R}^4$ from the multi-room system. This input will be used to calculate the updated vent status. It will have one output, the updated vent status $h \in \mathbb{R}^4$, which will feedback to the multi-room system as well as go to the safety monitor.

3) Safety Monitor: The safety monitor is a crucial component of the multi-room heating system, designed to ensure operational safety at all times. It operates by continuously analyzing two primary inputs: the current room temperature and the vent status. These inputs are pivotal in assessing the safety state of the system.

- **Current Room Temperature:** This input is critical as it provides real-time data on the thermal conditions of each room. The safety monitor checks this input against predefined safe temperature thresholds. Temperatures significantly above or below these thresholds could indicate potential hazards, such as overheating or insufficient heating, both of which can lead to unsafe conditions.
- **Vent status:** The status of the vents, whether they are open or closed, plays a significant role in maintaining a safe environment. A closed vent when heating is needed or an open vent under high-temperature conditions can lead to safety issues, as well as the opening of more than 2 vents, which is an unexpected state according to specification. Thus, the safety monitor is designed to evaluate the vent status in relation to the room temperature to detect any discrepancies or anomalies that could signify safety risks.

The output of the safety monitor is a binary signal, $y \in \{0, 1\}$, where 0 indicates a safe state and 1 indicates an unsafe state. This simplistic yet effective output mechanism allows users to easily comprehend the safety status of the system. It's important to note that this output is not fed back into the control system of the heating mechanism. Instead, it serves as an informational tool for the user, which allows users to configure warning alarm systems. This design choice was made with the consideration that room heating is unlikely to pose serious safety-critical conditions.

Incorporating the safety monitor in the multi-room heating system architecture underscores our commitment to safety and reliability. By constantly evaluating the safety parameters and providing a clear, immediate indication of any unsafe conditions, the Safety Monitor acts as a vigilant watchpoint, further enhancing the system's reliability and allowing preemptive identification of potential hazards. Overall, the safety monitor is a key feature of the multi-room heating system, providing an essential layer of safety assurance.

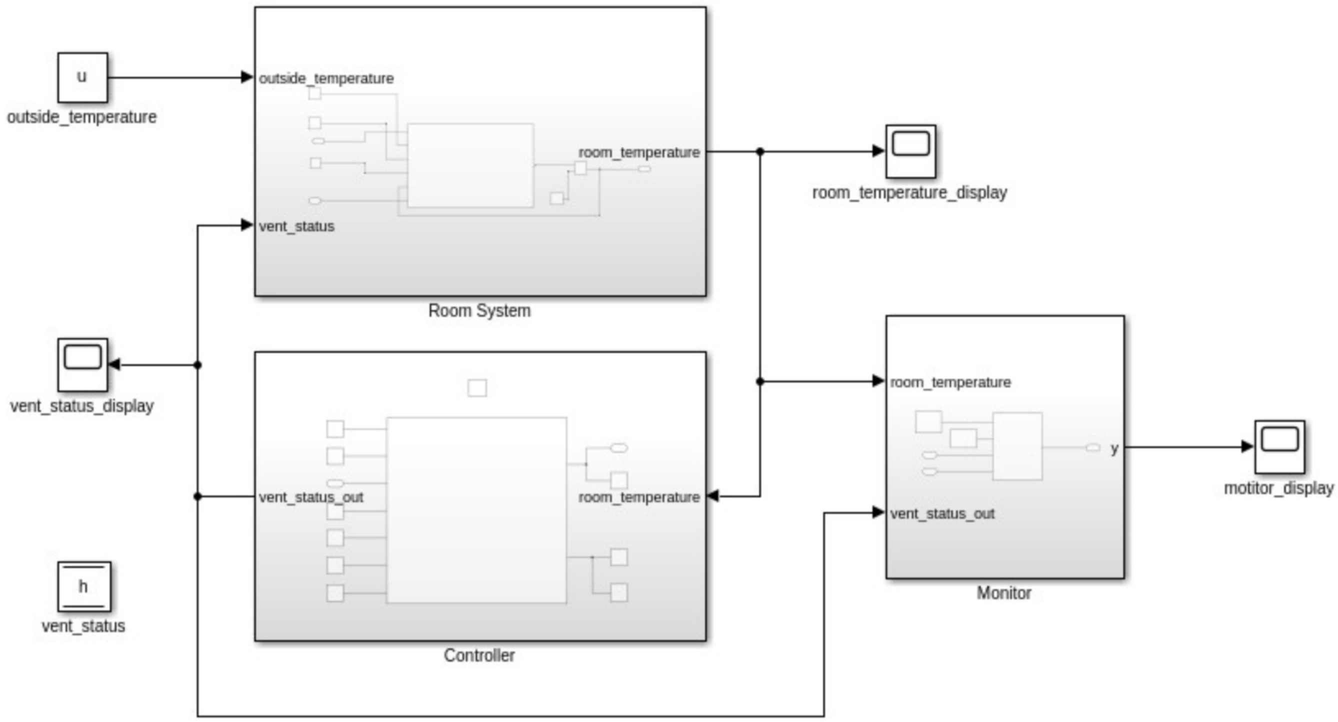

Fig. 1. Overall System Design in Simulink

D. Simulink Components

To visually inspect the outputs of those components, the current temperature *x*, the vent status *h*, and the safety monitor output *y* are all connected to a Simulink Scope Viewer block for viewing the value change over time. The output will be a line plot with different colors for lines representing different rooms for the display of temperature and vent status. The safety monitor plot will be a single link with a binary value of either 0 or 1.

To avoid await dependency in our system, we explicitly set an initial value for room temperature and vent status, which is by either creating constant or data store memory blocks. There are also many alternative solutions such as setting a delay block between the multi-room system and the controller, but we want to minimize the number of blocks used in our design for simplicity.

IV. Component Design

In this section, we will go over the design of each component of the system, including the multi-room system, the controller, and the safety monitor.

A. Multi-room System

The design of the multi-room system is presented in Figure 2. For modeling the multi-room system, the most important task is to implement the room dynamics introduced in the previous section. Since in this problem, all the heat transfer factors are constants, we used constant blocks in Simulink to represent them. The actual dynamics formula is implemented as a MATLAB function, which takes input A, b, c, h, u and output $\dot{x}$. The output rate of change of temperature $\dot{x}$ will go into the Integrator block for calculating the updated room temperature. There is also a x_0 constant that also goes into the integrator block, which is the initial temperature of the room and is used to avoid await dependency when running the simulation of the whole system.

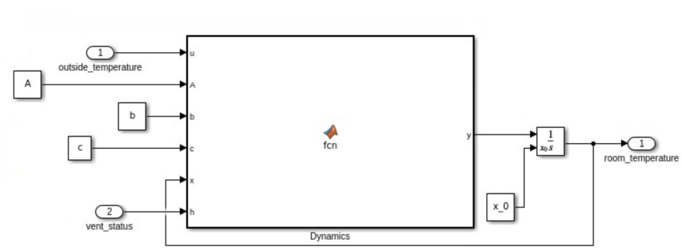

Fig. 2. Design of Multi-room System in Simulink

B. Controller

The design of the multi-room system is presented in Figure 3. It similarly takes in a lot of constants. In order to properly control the vent status to maintain the desirable temperature and not violate the safety requirements that require a) room temperature is within a reasonable range and b) only allowing up to two vents to be open at the same time, we need to consider two cases: when a vent should be open and closed in a room, and when to close up a vent in a room to allow the opening of the vent in another room.

1) Open/Close Vent in each room: To determine whether a vent should be open or closed, we will set thresholds for determining when is a good time to open or close vents. For this problem, we will define two constants $\mathbf{on} = [on_i] \in \mathbb{R}^4$, which contains the temperature threshold for opening vents in each room, and $\mathbf{off} = [off_i] \in \mathbb{R}^4$ which contains temperature thresholds for closing vents in each room. Note that the on and off thresholds can be different for each room since the heater transfer factor in each room can be very different. The logic to open vents thus becomes checking first if it is possible to open a vent in this room, i.e. opening here will not cause the overall vent open number to go beyond 2, and then if the current room temperature is below given *on* thresholds. The logic to close vents is simpler; if there is an open vent present in the current room and if the temperature is above the *off* threshold.

2) "Moving" Vents: When a room is in need of heating up but already two vents are open at the same time, it is important to close up the vent in the room that does not urgently need more heat to maintain the desired temperature. Although we are not physically moving the vents, the logic is very similar to "moving": one vent in room j will be closed up for room i to have its vent open to maintain the number of open vents below three. To determine when is reasonable to close vent in one room and open in another, we will also need to define additional constant: $\mathbf{get} = [get_i] \in \mathbb{R}^4$will be used to represent the temperature thresholds for allowing each room taking over a vent from another room, and $\mathbf{dif} = [dif_i] \in \mathbb{R}^4$ will be used to indicate the allowed temperature difference for each room to get vent from another room.

To "move" a vent from room j to room i, the rooms need to satisfy the following requirements

- Room i does not have an open vent.
- Room j has an open vent current or has just closed it.
- Room i has temperature $x_i < get_i$
- Room i and Room j has temperature difference $x_j - x_i >= dif_i$

There is a possibility that multiple rooms satisfy those requirements for a single room i that needs its vent open. At this time, we used the most simple strategy that pick the first room possible. This does not guarantee optimality and is just one of many strategies possible.

To implement both vent status update logic, we used a MATLAB function block in Simulink in the controller component. The inputs to this function will be the four constant on, off, get, dif, the current temperature x, the previous vent status h, and an additional input pos which records the most recent open vent positions to keep track of open vent status and number. It's important to note that $pos = [pos_i] \in \mathbb{R}^4$ with $pos_i \in \{0, 1\}$ is not the same as vent status h. $pos_i = 1$ means that room i has most recently possessed an open vent, but the status of the vent may not necessarily be open. It serves as an indicator of whether a room can just turn on the vent immediately or may require to "move" a vent from another room that used to "possess" an open vent. Those inputs will be used in calculating the updated vent status in each room according to the two above-mentioned update logic, which eventually outputs a single updated vent status vector that will then be fed into both the multi-room system for updating room dynamics and the safety monitor for checking safety requirements.

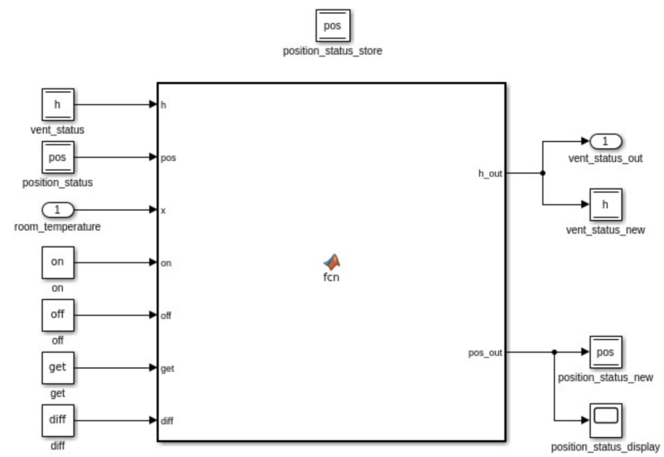

Fig. 3. Design of Controller in Simulink

C. Safety Monitor

Similarly, the design of the safety monitor is shown in Figure 4. Compared to the above two components, its logic is less complicated. It will take in two inputs, the current temperature from the multi-room system, and the updated vent status from the controller, and make sure those outputs satisfy the safety requirements defined in the previous section. A MATLAB function block is used to implement the checking logic, which takes in four inputs: min_temp, max_temp, vent status, and room temperature; the function will then check if all room has temperature within the temperature thresholds specified by the two constant min_temp, max_temp. It will also check if there are more than two open (i.e. two 1s) in the vent status vector. If any of the requirements fail, it will output a 1, otherwise, it outputs 0.

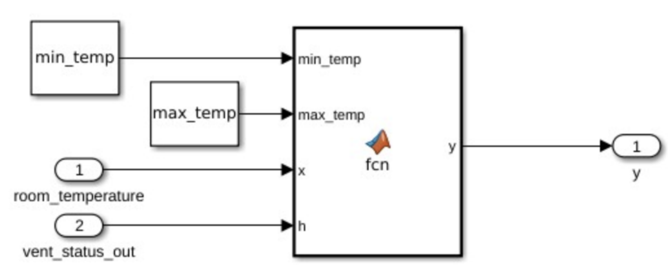

Fig. 4. Design of Safety Monitor in Simulink

V. Code Implementation

The system outline was initially developed in Google Colab using Python for quicker testing and result visualization. However, due to the challenge of modeling continuous-time systems in Python and the complexity introduced by using Classes to represent system components, we transitioned to Simulink after completing the fundamental logic in Python. Both the Python Colab and Simulink implementations are available in the submitted Github Repository, accompanied by comprehensive code comments. Additionally, the project demo video on the repository showcases both versions. It's worth noting a slight disparity in the simulation approach between Simulink and Python. In Simulink, the solver employed is ode45, the default for continuous systems, with a stop time set to 10 wall clock second, the step size is automatically adjusted by the solver. Conversely, in Python, the solve_ivp method from the scipy package is used, employing RK45 (Explicit Runge-Kutta method of order 5(4))[2] as the default method. The Python version utilizes a step size of 0.1 within a loop to simulate continuous time, with the stop loop index set to 1000, equivalent to 100 seconds.

VI. Simulation Result

In this section, we will present the simulation results from the Simulink implementation, given the initial conditions and constants defined in the provided project description file:

$$\mathbf{A} = \begin{bmatrix} 0.30 & 0.00 & 0.50 & 0.00 \\ 0.40 & 0.50 & 0.00 & 0.30 \\ 0.30 & 0.00 & 0.30 & 0.00 \\ 0.50 & 0.00 & 0.50 & 0.00 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0.30 \\ 0.20 \\ 0.50 \\ 0.40 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 7.00 \\ 11.00 \end{bmatrix}$$
$$\mathbf{on} = \begin{bmatrix} 19 \\ 19 \\ 19 \\ 19 \end{bmatrix}, \mathbf{off} = \begin{bmatrix} 20 \\ 20 \\ 20 \\ 20 \end{bmatrix}, \mathbf{get} = \begin{bmatrix} 17 \\ 16 \\ 16 \\ 17 \end{bmatrix}, \mathbf{dif} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad (2)$$

Initially, the vents in room 2 and room 3 are enabled to open.

Under those conditions, a Simulink simulation was run with a stop time equal to 10 wall clock seconds, and all components were updated at the same rate. The temperature over time for each room is shown in Figure 5, the vent status for each room is in Figure 6, and the Safety Monitor output is in Figure 7.

From the temperature plot, we are able to observe that temperature always stays within the specified range between 15 and 20 degrees, with various degrees of fluctuation for different rooms at different times. This visual inspection already shows that the first temperature safety requirement is satisfied.

However, from the vent status plot, it is unclear whether at most two vents are open at the same time, which requires us to check the output of the safety monitor for clearer visualization.

[2]https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html

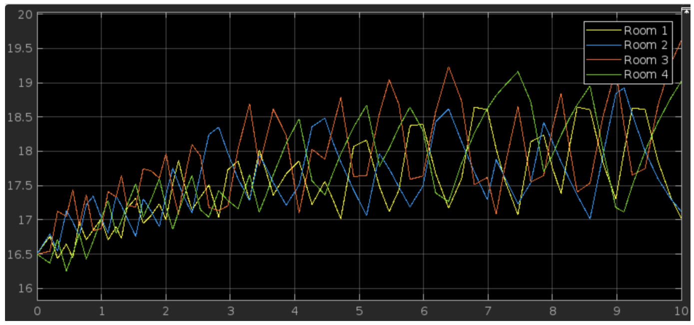

Fig. 5. Room Temperature over 10s

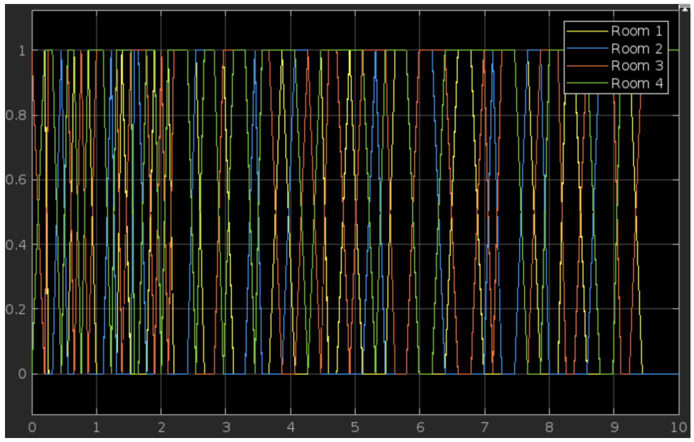

Fig. 6. Heater Status over 10s

Fig. 7. Safety Monitor output over 10s

The output from the safety monitor remains 0 throughout the simulation, indicating that all the safety conditions are met.

For the liveness requirement, there is no specific component designed to check for that, but we will be able to see that from the vent status output plot. We can observe that for all rooms, there is at least one that the value is 1, which means a vent is open in that room. Thus, the liveness requirement is also satisfied in this simulation.

We have also tested running a simulation with a stop time being 100s. The resulting plots are not shown in this paper, but the results are very similar, with the safety monitor being all 0 throughout the simulation, and temperature fluctuating but not crossing the upper threshold 20 and lower threshold 15.

VII. Discussion and Future Work

In this project, we adhered to the provided project description and implemented a multi-room heating system. We proposed assumptions, designed the system and its components, defined interfaces, and eventually implemented various components with corresponding logic in both MATLAB Simulink and Python. Despite achieving a fully implemented system and conducting simulations under specified conditions to verify system correctness, several improvements are still needed to render the system applicable to real-world multi-room heating problems.

A notable drawback in this project, which deviates significantly from real-life systems, is the use of constant values for heating factors, temperature thresholds, and outside temperature, a choice made for the sake of computational simplicity. The external temperature functions as a continuous-time system, much like the heat transfer factor, potentially affected by factors such as room occupancy, ongoing activities, and various other considerations. Moreover, the existing methodology treats the on and off thresholds for vent operations as constant inputs, overlooking dynamic adjustments influenced by external factors like ambient temperature and room-specific heat transfer characteristics.

Finally, the existing approach for "moving" vents relies on the straightforward selection of the first possible movement, a strategy that may not be optimal for vent status control. The project implemented the most basic form of controller, but various advanced controllers could be explored to address this issue. Incorporating machine learning models into controllers is anticipated to enhance performance, which possibly can not only maintain the desired temperature in all rooms but also be extended to consider additional factors, such as minimizing the number of vent status changes to conserve energy.

VIII. Related Works

The multi-room heating problem is a classic example of a hybrid system controller, which combines both continuous and discrete elements in its operation. In such a system, the continuous aspect is the temperature in each room, which changes over time due to factors like external weather conditions, insulation efficiency, and heat generated within the room. The discrete element comes from the heating controls, such as thermostats, which switch the heating on or off based on predefined temperature thresholds. This creates a dynamic environment where the controller must constantly adjust to maintain desired temperature levels in different rooms. The challenge lies in optimizing energy consumption while ensuring comfort, requiring sophisticated algorithms that can predict temperature changes and make efficient decisions. Over the past years, many models and control methodologies have been proposed to address this control problem.

A. Model Predictive Control

Model Predictive Control (MPC) is an advanced control strategy that uses a mathematical model of the multi-room heating system to predict future temperature outcomes [1]. This model takes into account various factors such as the current temperatures in each room, heating system dynamics, external weather conditions, and even occupant preferences. By simulating future scenarios, MPC can anticipate how different control actions will impact the system [2]. One of the strengths of MPC is its ability to handle constraints effectively and is inherently adaptive [3]. Recent advances have incorporated fuzzing with MPC. Specifically, the implementation of a soft constraint scheme through fuzzy optimization allows for occasional violations of constraints in scenarios where it doesn't compromise the system's proper operation, leading to reduced power consumption of the HVAC systems [4].

B. Neural Network-Based Control

With the rise of machine learning, neural network-based controllers have been explored for multi-room heating problems [5]. Neural networks are trained using historical data, which may include temperature variations, heating system responses, weather conditions, and user preferences. Over time, the network learns to identify patterns and correlations in this data, allowing it to predict and respond to future heating requirements more effectively [6], [7]. Neural network-based models especially excel in complex environments, with many interacting variables and nonlinear relationships. The neural networks can handle these nonlinearities and interactions more effectively than traditional control systems by finding meaningful patterns, which makes them particularly suited for optimizing heating across multiple rooms with different heating needs [8], [9].

C. Internet of Things (IoTs) Integration

The integration of IoT technology in multi-room heating systems allows for more granular data collection and control [10]. IoT-enabled systems can gather real-time data from various sensors in each room and adjust the heating accordingly, often in conjunction with advanced control algorithms, such as fuzzy logic [11] and neural networks, to achieve home automation [12]. Future developments in IoT for multi-room heating will likely involve more advanced machine learning algorithms and increased data interoperability between different devices [10].

D. Proposed Heating Controller

Though these methodologies were not implemented in our controller, we have utilized the concepts behind MPC and developed a model that is environment-aware and adaptable to user preferences and thermal properties of a house. In fact, the model described in our paper incorporates thermal dynamic properties to prioritize and ensure optimal comfort in each room while maintaining energy efficiency, making it well-suited for simple heating applications to maintain a desired temperature.

ACKNOWLEDGMENT

We want to express our gratitude to Dr. Abhishek Dubey from Vanderbilt University for providing the project specifications and guidance on how to approach this project.

REFERENCES

[1] E. Carrascal, I. Garrido, A. J. Garrido, and J. M. Sala, "Model predictive control for the heating system of a public building," in *2014 World Automation Congress (WAC)*. IEEE, 2014, pp. 433–438.

[2] P. Gruber, M. Gwerder, and J. Tödtli, "Predictive control for heating applications," in *Clima World Congress*, 2001.

[3] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: An engineering perspective," *The International Journal of Advanced Manufacturing Technology*, vol. 117, no. 5-6, pp. 1327–1349, 2021.

[4] A. Endo, O. Cartagena, D. Sáez, and D. Muñoz-Carpintero, "Predictive control based on fuzzy optimization for multi-room hvac systems," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–8.

[5] A. Agouzoul, B. Chegari, M. Tabaa, and E. Simeu, "Using neural network in a model-based predictive control loop to enhance energy performance of buildings," *Energy Reports*, vol. 8, pp. 1196–1207, 2022.

[6] Z. Li, Z. Sun, Q. Meng, Y. Wang, and Y. Li, "Reinforcement learning of room temperature set-point of thermal storage air-conditioning system with demand response," *Energy and Buildings*, vol. 259, p. 111903, 2022.

[7] L. Lei and W. Liu, "Predictive control of multi-zone variable air volume air-conditioning system based on radial basis function neural network," *Energy and Buildings*, vol. 261, p. 111944, 2022.

[8] A. Ruano, E. Crispim, and P. Frazão, "Moga design of neural network predictors of inside temperature in public buildings," in *Soft Computing Based Modeling in Intelligent Systems*. Springer, 2009, pp. 35–61.

[9] G. Gommers and M. Lazar, "Modular smart decentralized mpc for temperature control in multi-zone buildings by using identified physical models and neural networks," 2022.

[10] E. Png, S. Srinivasan, K. Bekiroglu, J. Chaoyang, R. Su, and K. Poolla, "An internet of things upgrade for smart and scalable heating, ventilation and air-conditioning control in commercial buildings," *Applied Energy*, vol. 239, pp. 408–424, 2019.

[11] B. Irawan, K. Tambunan, and H.-C. Chen, "Internet of things (iot) based multi-server room temperature and humidity monitoring and automatic controlling by using fuzzy logic controller," in *Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science)*, vol. 102. Springer Nature, 2023, p. 76.

[12] M. A. Emmanuel and M. J. Baptiste, "Home automation with remote monitoring: Design of a hybrid model for internet of things technology."