A METHODOLOGY FOR AUTOMATING THE IMPLEMENTATION OF ADVANCED CONTROL ALGORITHMS SUCH AS MODEL PREDICTIVE CONTROL ON LARGE SCALE BUILDING HVAC SYSTEMS

A Dissertation

by

ROHIT HARI CHINTALA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Bryan Rasmussen
Committee Members,	Suman Chakravorty
	David Claridge
	Sivakumar Rathinam
Head of Department,	Andreas Polycarpou

May 2018

Major Subject: Mechanical Engineering

ABSTRACT

Building operations consume about 40% of the total energy consumption in the US, with Heating Ventilation and Air-Conditioning (HVAC) comprising a significant portion of it. HVAC system of a typical commercial building consists of several components – cooling towers, chillers, Air Handling Units (AHUs), fans dampers, etc. Improving the performance of these components has the potential for large energy and cost savings. Implementing better control methodologies for regulating these components can be the first step in this direction for building managers, as it requires minimal retrofitting. Advanced control methodologies such as Model Predictive Control (MPC) can help realize this potential.

Three reasons for the inefficient operation of traditional control methodologies are identified in this dissertation, the improper tuning of PI controllers for nonlinear systems, a decentralized control architecture that doesn't perform any global optimization, and lack of planning for future operating conditions. The dissertation makes a contribution towards addressing the aforementioned areas of inefficiencies by offering a solution in the form of alternative control architectures such as cascaded control, and optimal control algorithms such as MPC.

The dissertation first provides the results of a survey that demonstrates the widespread nature of the phenomenon of hunting (undesired oscillations) in building HVAC systems. An algorithm to detect the presence of hunting in real time is proposed and implemented on data obtained from real buildings on the campus of Texas A&M University. A description of the cascaded control architecture is provided along with a simulation example to show how it can mitigate the problem of hunting.

The dissertation addresses the other two reasons for inefficient operation, namely

the decentralized control architecture and lack of planning for the future by proposing a method that would allow the process of implementing advanced control algorithms such as MPC to be automated and easy to implement. Currently implementing MPC on large scale building HVAC systems remains a major bottleneck as it requires the development of models that are accurate and easy to compute. This dissertation makes a contribution towards this front by proposing a modeling algorithm that can be automated and scaled to systems comprising hundreds of components. The modeling algorithm is verified using data from a real office building. Static models are developed for the Air Handling Unit (AHU) pressure subsystem which includes the AHU fan, and Variable Air Volume (VAV) boxes serving conditioned air to the rooms. In addition to the static models, dynamic models are developed for the AHU temperature subsystem comprising a heat exchanger that uses chilled water (CHW) to cool the air passing through the AHU. Dynamic models are also developed for the temperatures of 9 of the 11 rooms of the office building, thereby demonstrating that the proposed algorithm can be implemented for multi-zone systems.

The dissertation also makes a contribution towards the implementation of advanced controls by providing a method by which the black-box models can be used to implement MPC on building HVAC systems. MPC using models developed from the proposed modeling algorithm is applied to a high-fidelity simulation model of the office building. Results of the simulation show that MPC can provide significant energy savings over the traditional control algorithms.

DEDICATION

To my parents Shobha Chintala and Hari Chintala, and my brother Rahul Chintala.

ACKNOWLEDGMENTS

I'm extremely grateful to my advisor Dr. Bryan Rasmussen for mentoring me and providing invaluable support and guidance throught the course of my degree.

I'm also extremely thankful to all my labmates, Chris Bay, Chris Price, Rawand, Kaimi, Austin, Trevor, Jorge who provided support not just in reasearch but making the lab environment a more fun place to work.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Dr. Bryan Rasmussen (advisor), Dr. David Claridge, and Dr. Sivakumar Rathinam of the Department of Mechanical Engineering, and Dr. Suman Chakravorty of the Department of Aerospace Engineering.

Funding Sources

This work was made possible in part by the National Science Foundation under Grant No. CMMI-1563361. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

Commercial and residential buildings accounted for 41% of the total energy consumption in the US in 2014 [1]. In developed countries, the consumption of electricity by building operations is growing at the rate of 0.5-5% every year [2]. The building operations however, are far from optimal, and there is a large potential for energy and cost savings that can be achieved in this sector. There has been a growing effort to realize the energy and cost savings by seeking to reduce building energy demands and consumption. For instance, the Department of Energy (DOE) has been responsible for several states within the U.S to enforce energy codes which allow for 25 % lesser energy consumption in commercial buildings compared to the previous commercial energy codes. Furthermore, there have been voluntary programs such as Energy Star which have resulted in commercial buildings consuming 35 % less energy on average compared to similar buildings not part of the program [3].

One of the ways in which the energy consumption and demands can be reduced is by using more state-of-the-art equipment which have higher efficiencies and also minimize the energy loss to the surroundings. For example, some of the measures to reduce energy consumption in buildings include replacing the existing lighting with high efficiency LEDs, replacing the chiller with a higher Seasonal Energy Efficiency Ratio (SEER) value, increasing the insulation of the walls, etc. Such solutions however involve expensive retrofittings and modernizations [4] . Another approach by which significant energy savings can be achieved and also has minimal retrofitting requirements, is to implement advanced controls to regulate building operations such as Heating, Ventilation, and Air-Conditioning (HVAC), lighting, blind control etc.

Among building operations, HVAC systems are the principal consumers of energy.

The primary objective of HVAC systems is to condition the building space for occupants and equipment. Depending on the environmental conditions, space conditioning may account for over 50% of the energy consumption of a building [2]. As stated above, adopting advanced control methodologies can be the first step towards reducing this energy consumption. But HVAC controls still remains a major untapped avenue with regards to optimizing the performance of building operations. A significant gap exists between the control theory that is available in literature, and the controls applied on building HVAC systems.

The most widely implemented control algorithms in buildings are Proportional Integral (PI), and Rule-Based Controllers (RBCs). For example, a survey of the HVAC systems at the Texas A& M University showed that all HVAC components were controlled using simple PI-type control [5]. The widespread use of these algorithms is mainly due to their ease of implementation. The control outputs are easy to compute and have data requirements that are easy to obtain. The ease of implementation, however, comes at the price of HVAC operations being inefficient.

Just by switching to more advanced control algorithms, significant improvement in energy efficiency over the aforementioned traditional control methodologies can be obtained. One such advanced control methodology is Model Predictive Control (MPC). There are, however, several bottlenecks to making the switch from traditional to advanced control algorithms. For example, the implementation of MPC on real building systems relies on mathematical models that predict the behavior of the system being regulated. Development of reliable models that can be scaled to HVAC systems with hundreds or thousands of components is one of the principal bottleneck to its wide spread implementation on real buildings. In addition to problems with model development, there are not many papers in literature that have tested the MPC algorithm on large scale building HVAC systems.

This dissertation makes contributions towards two fronts. First, by proposing a black-

box modeling method that can be automated and scaled. The modeling method is also verified using data from a real working office building. Second by developing models of a high fidelity simulation system and applying MPC under varying operating conditions. The simulation results are used to quantify the energy savings that can be obtained over traditional control methodologies.

The rest of the chapter is organized as follows. Section 1 presents the reasons for the inefficiencies in traditional control. Section 2 provides a background and literature review of the modeling methods available in literature for implementing MPC, and the problems associated with each method. Section 3 discusses how MPC can be used to improve the efficiency of building HVAC operation. Section 4 describes the different control architectures by which MPC can be applied. Finally in Section 5, an outline of the dissertation is presented.

1.1 Reasons for Inefficiencies in Traditional Control Practices

There are three main reasons for inefficient operation of HVAC systems that employ traditional control that are identified in this thesis, namely, poor controller tuning, decentralized control architecture, and lack of planning for future disturbances and operating conditions. The first reason that is explored in this dissertation is the improper tuning of the proportional and integral gains of the PI-type controllers.

The gains of the proportional and integral components of the controller are generally tuned for a particular operating condition of the HVAC system. But when the operating conditions change as a result of changing weather conditions, changing internal loads etc., the previously tuned gains of the controller may no longer be suitable. The gains become unsuitable primarily due to the nonlinear and time-varying nature of HVAC systems. At the new operating condition, the controller with improper gains is no longer able to accurately track the setpoint assigned to it. This results in undesired oscillations also called

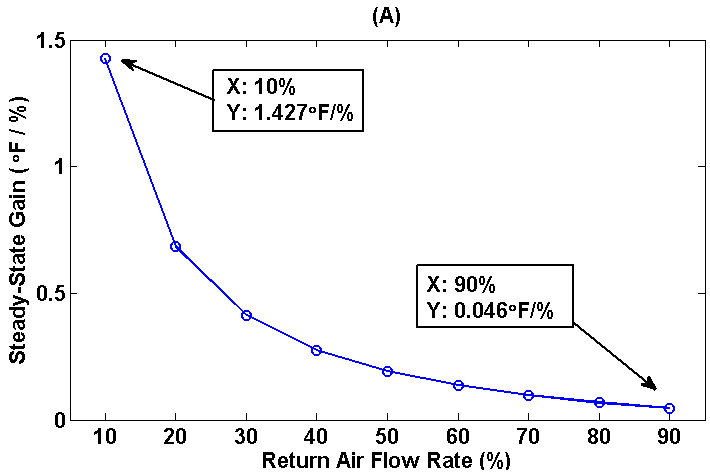


Figure 1.1: Steady State Gain as a Function of Valve Position

hunting of HVAC components.

For example, consider a chilled water (CHW) valve that regulates the amount of CHW flowing through the heat exchanger of an Air-Handling Unit (AHU). A PI controller is used to regulate the CHW valve opening to track a certain AHU discharge air temperature. The valve characteristics are nonlinear which means that for a same change in valve position the change in flow rate of the CHW is different when the valve is partially open compared to when the valve is fully open. A PI controller which is tuned for an operating condition corresponding to a fully open valve position, causes oscillations in CHW flow when the valve is partially open. The oscillation in the CHW flow in turn causes an oscillation in the discharge air temperature of the AHU resulting in hunting behavior. Figure 1.1 shows another nonlinear relationship where the steady-state gain of the discharge air temperature per unit increase in the return flow rate, changes with the flow rate.

The first step in trying to reduce the undesired oscillatory behavior in the HVAC components is to detect its presence and prevalence. In Chapter 2 of this dissertation an algorithm is proposed to detect the presence of hunting in HVAC components in real time. Next, a survey is performed across the various buildings of the Texas A&M University

campus to find the prevalence of hunting in CHW valves and fan speeds. An alternative to the traditional feedback control architecture called cascaded control is described. The cascaded control architecture has the potential to to reduce the problem of hunting in building HVAC systems. A simulation example is provided that demonstrates the same.

The second reason for inefficiency of traditional control in buildings arises due to the fact that the control architecture employed is decentralized. HVAC operations involve highly interconnected subsystems with often times competing requirements. For example in a chiller system, a lower supply condenser water temperature results in lower cost of operation for cooling CHW. At the same time a lower supply condenser water temperature requires higher speed of the cooling tower fan thereby consuming greater energy. In decentralized control architecture, each HVAC component has its own controller that regulates its actuator without taking into account the interaction with the neighboring systems. In the example above, the PI controller for the chiller seeks to track a CHW supply temperature setpoint without accounting for the impact on the fan speed of the cooling tower. Computing the optimal CHW setpoint temperature would require a knowledge of how the setpoint temperature affects the fan speed of the cooling tower. Since in decentralized control such interactions are not considered, the control setpoint generated results in suboptimal performance.

The third reason identified in this dissertation occurs due to the fact that the existing control methodologies do not take into account predictions of future disturbances which may be possible to forecast such as utility prices, weather conditions, occupancy etc. There are several components in the HVAC system that can be used to plan ahead by making use of the predictions of the aforementioned disturbances thereby saving energy and improving performance. One such example is making use of the building thermal mass to store cooling energy during the night time when the temperature and electricity prices are low. The stored energy can be used at a later time when the outside temperature and/or electric-

ity prices are higher to meet the cooling requirements. The PI-type control methodologies have no such planning capabilities.

1.2 Modeling of Building HVAC Systems

Some of the reasons for the inefficiencies associated with traditional control of building HVAC systems are listed above. Advanced algorithms such as MPC have the capability of vastly improving the efficiency of HVAC operation. The second and third reasons for inefficient operation associated with traditional control, namely, the decentralized architecture and lack of planning for future disturbances can be addressed by implementing MPC. MPC addresses the inefficiency caused due to the decentralized nature of traditional control by taking into considerations interactions between the various HVAC systems in its computation of component setpoints. For instance, in the chiller example MPC takes into consideration the energy consumption of both the chiller and cooling tower fan in computing the discharge temperature setpoint of CHW. In addition to performing component level optimization, MPC addresses the third reason for inefficient operation of traditional control by making use of predictions of future operating conditions to plan the best set of actions.

In order to perform component level optimization, and estimate the optimal set of inputs over a future time period, MPC requires a model of the system being regulated. This section provides a background and literature review of the methods that can be used to model building HVAC system behavior.

The opportunity for large savings by implementing MPC, and the difficulty in developing a reliable model has led to intense research in the area of building modeling [6]. A survey of the modeling approaches for building HVAC systems is provided in [7]. The modeling methods available in literature can be broadly classified into three categories – white box, gray-box, and black-box models [8]. The modeling methods differ in their

complexity, accuracy, and the amount of information that is required to develop them. Figure 1.2 shows a graphic representation of the three types of approaches. A description of the state-of-the-art corresponding to each modeling method, and the pros and cons of adopting the methods for implementing MPC on large scale building HVAC systems is provided as follows.

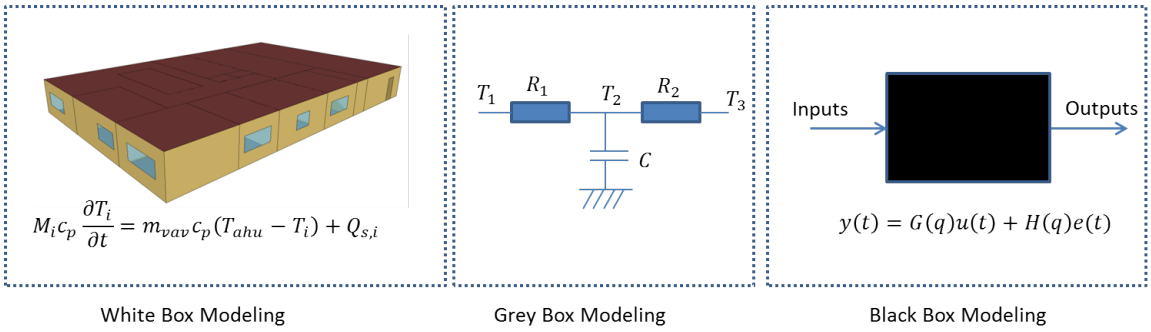


Figure 1.2: Three Types of Building Modeling Approaches Available in Literature

1.2.1 White-Box Modeling Method

White-box models such as EnergyPlus [9], and TRNSYS [10] make use of extensive mathematical equations developed from first principles of physics. They provide the most accurate description of the thermal dynamics of the building compared to other modeling methods. In order to perform simulations and predict outputs using white-box models, however, requires a detailed description of the components of the system. For example, developing an EnergyPlus model of a building requires information about the building plans, construction materials, placement of windows and doors, geographical location etc. A flow diagram of the steps associated with running simulations using a white-box model such as EnergyPlus is shown in Figure 1.3.

Inputs are entered into the EnergyPlus model which constitute static and dynamic val-

ues classified based on whether the values remain constant or are varying throughout the simulation. The static inputs include parameters such as the aforementioned building constants, system parameters, and plant parameters. The term system is used to describe the components associated with the distribution of cooling energy such as the AHU fan and heat exchanger. The term plant is used to describe the components such as chiller and cooling towers which deal with the primary cooling fluid like CHW. The dynamic inputs on the other hand include factors such as outside air conditions, occupancy, control setpoints such as discharge air temperature setpoint, room temperature setpoint, etc. The values of the dynamic inputs can be varied at each simulation time step.

The simulation engine then uses the aforementioned static and dynamic inputs provided by the user to perform the thermodynamic calculations to compute the values of the simulation outputs. Simulation outputs include room temperature and humidity, energy consumption of the components, etc.

Although the white-box models are the most accurate method available for modeling building HVAC systems, the information required to model is difficult to obtain, and even if available, is extremely time consuming and difficult to build. In addition, the complexity of the equations to be solved make the white-box models unfeasible for implementation of on-line model based control.

Notwithstanding the complexity, there have been attempts made at using EnergyPlus models for online control. For example in [11] an EnergyPlus model of a building in Colorado was created to optimize window operations that facilitate natural ventilation in conjunction with mechanical cooling. In order to use the white-box models for control, authors in [11] coupled the simulation engine of EnergyPlus with a computational software such as MATLAB. The simulation outputs generated by EnergyPlus were read by the computational software which then produced a series of candidate values for the window opening using an optimization algorithm at each time step. EnergyPlus simulations were

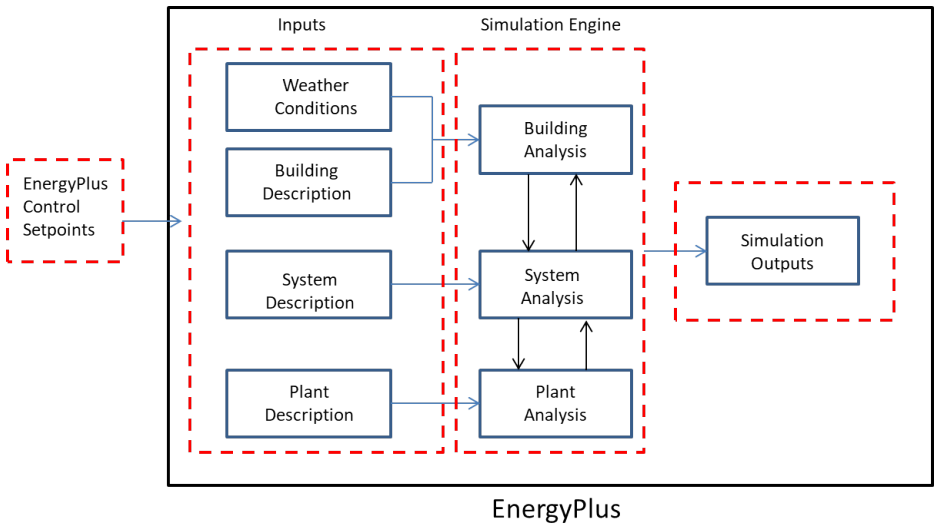


Figure 1.3: Layout Showing Flow of Data of a White-Box Model Coupled with MATLAB

run for each candidate value. From among the simulations, the window operation that provided the best objective function value was selected to advance the simulation.

In order to find optimal cooling schedules for a building, a particle swarm optimization algorithm was used in [12]. In the paper, the cooling schedules were regulated by a single optimization variable,namely, the opening and closing times of windows. In the particle swarm algorithm several EnergyPlus simulations are run corresponding to each particle in the solution space. A particle corresponds to a feasible solution to the optimization problem. The particle that resulted in the objective function being minimized is selected as the solution. Particles are generated based on a combination of randomized and rule based decisions. Simulations were performed over a year and the authors were able to demonstrate that by using night-time cooling strategies cooling energy could be saved by up to 40%.

The examples above showed instances of white-box models being used for online control. The authors optimized a single control variable namely the window operation of

buildings. In order to advance the simulation or real building operation by one time step, several paths that the building system can take in the time-step under question were tested by running EnergyPlus simulations corresponding to each path. Each path represents a different window operation. The window operation schedule that provided the best result is then applied to advance by one time step.

Since the only variable being optimized in the aforementioned examples is the window operating schedule, the solution space was one dimensional. But an efficient operation of building HVAC system requires the optimization of several components. As the number of components increases the dimensions of the solution space also increases and the number of particles in the swarm to be tested grows exponentially. Hence the process of running several simulations at each time step to determine the optimal operation becomes infeasible when trying to optimize the entire building HVAC operation and not just window opening schedules.

Although not suited for implementing control, the white-box models can serve as a simulation environment. The high level of accuracy makes them a perfect platform to test various modeling and control strategies under a wide range of operating conditions which would generally not be possible to perform on a real building. In this dissertation the EnergyPlus simulation environment is used to compare the traditional PI-type control with MPC.

1.2.2 Gray-Box Modeling Method

The white-box modeling approach described above is a forward modeling approach where the model development starts with a physical description of the building. Inverse modeling approaches on the other hand are developed using an empirical approach. The properties of the system are expressed in terms of inputs and parameters that are determined using statistical tools [13]. Gray-box modeling method is an inverse modeling

approach that is a hybrid of white-box and the inverse modeling method.

The gray-box methods in literature for modeling of building HVAC systems are used to estimate the cooling or heating loads of the building space being conditioned. The building heating and cooling loads are estimated by predicting the cooling or heating energy required to maintain the temperature and humidity of the thermal zones [13]. The cooling or heating energy required for maintaining the temperature and humidity has two principal components, namely, sensible loads and latent loads.

Latent loads are the loads associated with the moisture content in the air. They are added to a thermal zone mainly by respiration from occupants and infiltration of humid outside air. Sensible loads on the other hand are loads that result in the change in the temperature of the air. They arise due to transfer of heat from the surfaces of the zone such as walls, floors and ceilings to the room air. The source of sensible zone loads can be both internal and external. The internal sensible loads arise due to the heat emitted by the occupants, room lighting and equipment etc. The external zone loads arise due to factors such as solar radiation, conduction between walls and outside air etc [13]. Sensible zone loads have a significantly greater impact on the total zone load and is the primary focus of most building modeling methods available in literature.

The most widely used gray-box method for modeling sensible loads of thermal zones is the Resistance-Capacitance (RC) network approach. The thermal dynamics of a building zone is converted into an equivalent RC circuit. The RC network approach is based on the transfer function 1.1 proposed by Braun in [14] which predicts the sensible cooling load required to maintain the temperature of a thermal zone.

$$\dot{Q}_{zs}(t) = \sum_{i=0}^{n} a_i T_{oa}(t-i) + b_i T_z(t-i) + c_i \dot{Q}_{int}(t-i) + d_i \dot{Q}_{sol}(t-i) + \sum_{i=1}^{m} \dot{Q}_{zs}(t-i) \tag{1.1}$$

The thermal energy gained by a particular zone at time instant t ($\dot{Q}_{zs}(t)$) is a function of the past outside air temperatures (T_{oa}), the past temperatures of the zone (T_z), the internal heat loads ($\dot{Q}_{int}$), and the external solar radiation ($\dot{Q}_{sol}$). The parameters a_i, b_i, c_i, d_i, and e_i are experimentally determined using statistical analysis.

In [13], Braun proposed that the transfer function shown in 1.1 can be represented in a state space form as shown in Equation 1.2

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{1.2}$$

by assigning the driving forces such as outside air temperature, internal and external solar loads etc as the inputs u, the sensible load as the output y and selecting a state vector x that comprises the temperature of various nodes in the building. Braun further showed that the state space description shown in Equation 1.2 can be expressed as an equivalent RC circuit. An example of how the building thermodynamics is represented by an equivalent RC circuit is shown in Figure 1.4 .

In the RC circuit the thermal masses of the rooms are treated as capacitance, and the resistance to the flow of heat between two zones or between the zones and the outside air are represented as thermal resistors. Each room comprises two distinct thermal masses. The temperature of the air in the room corresponds to the lower thermal mass C_{1a} and C_{2a}, and the temperature of the walls, ceilings and floors corresponds to the higher thermal mass C_{11} and C_{22}. The temperature of the thermal masses form the nodes of the RC circuit. The selection of nodes is a design parameter that depends on the number of sensor available for model identification. For example the nodes can correspond to the thermal zone air temperatures. In addition, if there are sensors that measure wall temperatures, they can be treated as separate nodes to develop a more accurate higher order model.

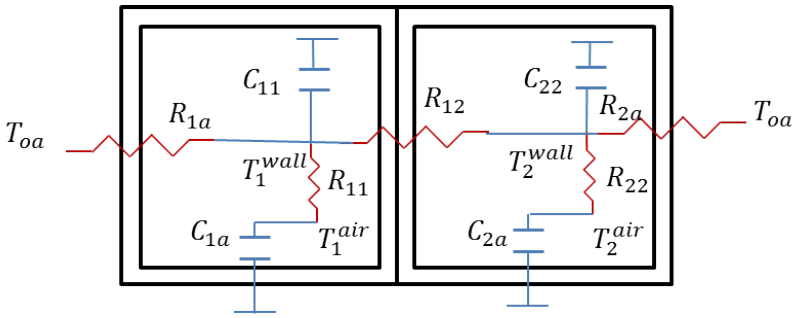


Figure 1.4: An Equivalent RC Circuit for a System Containing Two Thermal Zones

In Figure 1.4 R_{1a} and R_{2a} correspond to the thermal resistance between the outside air temperature and the wall temperatures of the two rooms, and R_{11} and R_{22} correspond to the thermal resistance between the wall and room air temperatures. The objective of the identification procedure is then to find the values of the thermal resistors and capacitance values using nonlinear regression analysis. The RC modeling procedure described above was used in [13] to estimate the cooling load of a building with two thermal zones and was shown to have an error less than 2% when compared to simulations run for a period of one year on a white-box model.

The RC network approach has been utilized in several papers [4],[15], [16], [17], [18] for the implementation of MPC in building HVAC systems. For example, in [4] an entire 4 story building on which MPC was implemented was divided into two thermal zones. The heating load required for the zones was modeled using the RC network approach. MPC was used to compute the optimal supply heating water temperatures to heat the thermal masses corresponding to the two zones. Energy savings between 15% to 28% were obtained depending on the operating conditions.

One of the common themes in the papers discussed above is that they gray-box model identification is performed by approximating a multi-zone building into a two zone system as was done in [4], or develop the model under experimental conditions by intentionally perturbing the operating conditions to obtain data suitable for identifying the parameters of

the RC circuit as was done in [19]. The main reason for making simplifying assumptions such as reducing number of zones, or setting up experimental conditions for identifying the RC parameters is that gray-box modeling is not trivial. Popular identification algorithms such as subspace and parametric identification methods are not suitable for identifying the RC parameter values [20]. Methods for estimating the RC parameter generally involve nonlinear regression such as that used in [21]

Although gray-box models are simpler to develop than white-box models, developing them for large scale systems is still difficult and time consuming. The difficulty arises due to the fact that the number of parameters to be estimated by nonlinear regression methods increases exponentially as the number of zones to be modeled increases. In addition to computational requirements, the process of automating the model development becomes more challenging due to the information requirement of graybox models. Gray-box models require a complete knowledge of the internal structure of the building. In order to increase the accuracy of the models, the papers in literature have also used information about the building description such as thickness of walls, etc to estimate the resistance and capacitance values of the equivalent RC network such as in [13]. The papers in literature have also used incident solar light on the outer building walls as inputs to the model. Information such as building wall thickness, orientation of the walls, internal structure of the rooms, and incident solar radiation on walls is not easily available.

Thus, the information requirement, computational burden required for the nonlinear regression analysis to compute the gray-box model parameters, and the fact that the RC modeling method is suitable only to model thermal zones make the gray-box modeling method less suitable for automating the development of models and scaling the process for HVAC systems with a large number of components.

1.2.3 Black-Box Modeling Method

Although, the two approaches described above provide a more accurate description of the underlying physics, the complexity, and the amount of information that is required to develop the models make them unfeasible to be implemented on large scale systems. Black box modeling approach on the other hand is developed purely from data. The models may be of lower fidelity but offer the best scope in automating the process of model development for large scale subsystems.

The output being modeled by the black-box approach is generally continuous in nature like room temperature, humidity etc. The sensors employed by the Building Energy Management System (BEMS), however, take measurements at fixed intervals of time, giving the output a discrete nature. The discrete output is modeled as a function of the previously measured output and input values. For example, sensors at an office building at the Texas A&M University campus in [5] measure room temperatures, volume flow rates, discharge air temperature, and weather conditions. By using the black-box modeling approach the room temperature at time instant t is expressed as a function of the room temperature, and other input values measured before t.

Not all the factors that influence the output being modeled are generally captured by the BMS sensor data. For instance in the above example where the room temperature is being modeled, factors such as solar incidence, cloud cover, wind-driven infiltration, etc. affect the output but are not measured by the BMS sensors. In the black-box modeling approach the unmeasured inputs are assumed to be stochastic in nature, generally Gaussian with zero mean and variance 1. The modeling approach uses statistical tools to extract the influence of these stochastic inputs from historical data. Thus the output modeled by the linear parametric black-box approach is written as a sum of deterministic and stochastic components.

Based on the structure of the linear function used to describe the relationship between the inputs and the outputs, the linear black-box modeling methods can be classified into two methods, parametric and subspace identification. In the linear parametric method, the relationship is expressed as a linear combination of contributions from past measured input and output values. For example, consider the parametric relationship shown in Equation 1.3

$$\begin{aligned} y(t) =& a_1 y(t-1) + a_2 y(t-2) + ... + a_{n_a} y(t-n_a) + b_1 u(t-1) + b_2 u(t-2) + \\ & ... + b_{n_b} u(t-n_b) + c_0 e(t) + c_1 e(t-1) + ... + c_{n_c} e(t-n_c) \end{aligned} \tag{1.3}$$

where $y(t)$ is the output at time t, u is the deterministic input, e is the stochastic input, and $a_1, a_2, ..., b_1, b_2, ...b_{n_b}, c_1, ..., c_{n_c}$ are the parameters that determine the contribution of the respective past input or output value.

Equation 1.3 shows just one of the ways in which the linear parametric function can be expressed. For example consider another linear parametric equation shown in Equation 1.4.

$$\begin{aligned} y(t) =& a_1 y(t-1) + a_2 y(t-2) + ... + a_{n_a} y(t-n_a) + b_1 u(t-1) + b_2 u(t-2) + \\ & ... + b_{n_b} u(t-n_b) + e(t) \end{aligned} \tag{1.4}$$

Equations 1.3 and 1.4 differ in how the deterministic and stochastic components are assumed to influence the output. In 1.4, the output is assumed to be independent of the past values of the stochastic input. The different ways of expressing the linear relationship are called as model structures. The most commonly used model structures in literature are ARX, ARMAX, Box-Jenkins (BJ) and Output Error (OE). Once a model structures is selected, the parameters of the structure are identified using standard statistical tools. Depending on the statistical tool used to identify the parameters, the linear parametric method

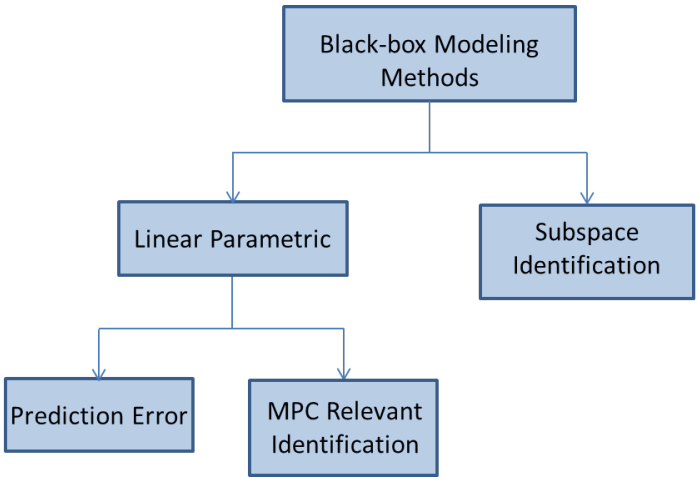


Figure 1.5: Black-Box Modeling Methods

can be further divided into Prediction Error Method (PEM) and MPC Relevant Identification (MRI) method. A discussion of how to select the model structure, and a comparison of the PEM and MRI methods is provided in Chapter 3.

The second approach for linear black-box modeling explored in this dissertation is the subspace identification method. In the subspace identification method the input and output relationship is expressed as a state space equation as shown in Equation 1.5,

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + Ke(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1.5}$$

where A, B, C and K are the state space matrices to be determined by statistical analysis. The states used to describe the model are purely mathematical and bear no physical significance. A detailed description of the two black-box modeling approaches is provided in Chapter 3. Figure 1.5 shows the different types of linear black-box modeling approaches popular in literature.

There are several papers available in literature that use the aforementioned black-box

approaches for modeling building thermal dynamics [22],[23],[24],[25]. In [22] linear parametric models with ARX and ARMAX structures were developed for a naturally ventilated greenhouse. The output selected was the air temperature within the greenhouse, and the inputs selected were the outside air temperature and humidity, solar radiation, and cloudiness of the sky. The authors were able to develop black-box models that provide accurate predictions of the air temperature. In order to maintain the accuracy of the models, however, the model parameters were regularly tuned to account for the changing operating conditions. In [26], the authors demonstrated that linear parametric black-box models can be developed to model room temperature and humidity of a two-zone office building with full occupancy using ARX, ARMAX, BJ, and OE model structures. The models were verified using data collected for over a period of nine months.

Although the papers provide good predictive models, there are certain inherent problems associated with the black-box modeling approach that are not addressed in literature. Black-box models are least computationally intensive compared to the other two approaches, and require the least amount of data for development. The drawbacks of the black-box modeling approach however is that they are developed with minimal underlying physics principles. As a result the predictions made by the model correspond to a specific set of operating conditions. There is no guarantee that the predictions made by the model would be accurate under a different set of operating conditions. This generates a question that remains largely unanswered in literature. Does a black-box model with good predictive properties translate into a good model for control? Furthermore, black-box models require a certain quality of input excitation to obtain models with good input-output relationships [27]. The question that arises as a result is whether the data obtained during the normal course of building operation is of sufficient quality, or whether experimental conditions where inputs are intentionally excited are required to develop a good model.

In addition to the two problems associated with black-box modeling mentioned above,

there are two areas in which the available literature on black-box modeling is lacking. Firstly, none of the papers include both buildings and AHU with all of the details [28]. Most of the papers model only the room temperatures. Optimizing the building HVAC operation requires models of other components such as the AHU fan, heat exchanger etc. Secondly, black-box models are developed only for systems with one or two rooms. Most buildings have several rooms with interconnecting dynamics. If the process of implementing MPC needs to be automated, a method to model multi-zone buildings is required.

Chapter 3 of this dissertation seeks to address the aforementioned shortcomings in literature by contributing on two fronts. Firstly by proposing a linear parametric modeling algorithm that automates the process of model development. The proposed modeling algorithm can use either the PEM or the MRI approaches. A comparison of the proposed algorithm using PEM and MRI approaches is made with the subspace identification method. The comparison is performed by making use of data from real working multi-zone office building. In addition to the room temperatures, the AHU discharge air temperature and the volume flow rates through each room are also modeled. Secondly, the dissertation contributes by using models developted from the proposed modeling method to apply MPC on a high fidelity simulation model of the office building. Thereby showing that black-box models not only provide accurate descriptions of the building HVAC systems, but can also be used to implement model based control.

1.3 Model Predictive Control

The previous section provides a background of the various approaches available in literature to model building HVAC systems. The objective of the modeling methods is not just to predict system behavior, but also to be able to use the predictions for model based control. The model based control investigated in this dissertation is MPC.

MPC does not comprise of a single strategy, but a collection of various control methods

which seek to minimize a certain objective function [29]. For building HVAC systems, the objective function is designed so as to meet the thermal comfort requirements of the occupants and machinery and at the same time minimize the energy consumption of the HVAC components. Thus the objective function is generally expressed as a sum of a measure of thermal comfort, and the cost of operations of the various subsystems such as the AHU fan, heat exchanger, chiller etc.

A relationship is then sought between the objective function that is required to be minimized and the control setpoints that regulate the system. Examples of some of the control setpoints that need to be optimized include the AHU discharge air temperature setpoint, the CHW supply temperature setpoint, etc. The models that are developed using any one of the methods described in the preceding section help in identifying a relationship between the objective function and the control setpoints. The relationship is then used by a mathematical optimization program to determine the set of control inputs that minimize the objective function. The control setpoints are optimized not just for the current time instant but over a future prediction horizon. The optimal inputs corresponding to the current time-step are then applied to the real system. Figure 1.6 shows a schematic of how MPC computes the optimal control setpoints.

There are two mechanisms by which the optimal inputs computed by MPC are able to realize an improvement over traditional control algorithms – load shifting, and component optimization [19]. Load shifting is the process of taking actions before-hand anticipating changes in demand conditions. For example, precooling the building so as to reduce the amount of cooling required during peak demand, storing energy for later use etc. Component optimization is the process of computing the optimal control action for each component in a large system with several inter-connected subsystems, with quite often competing requirements. For example, there are several combinations of discharge air temperature, and flow rates that can provide the same amount of cooling to a thermal zone.

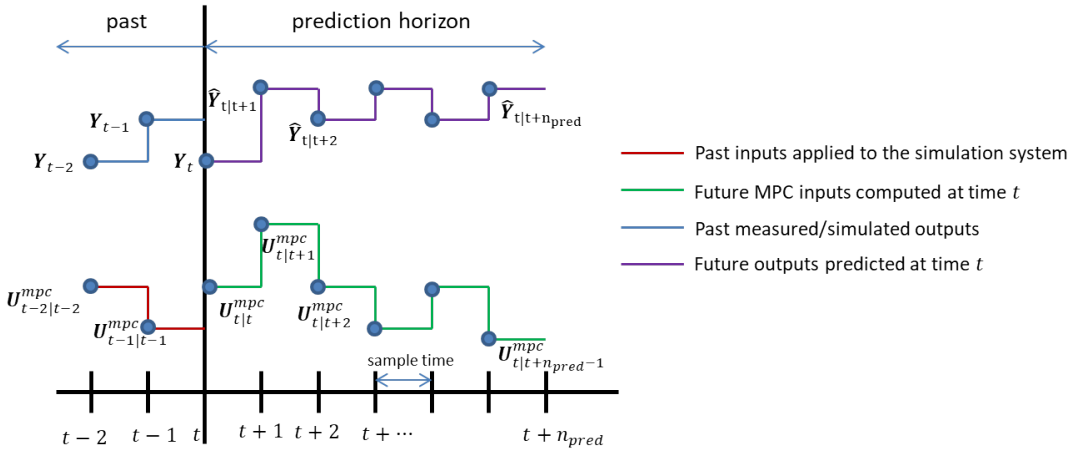


Figure 1.6: Flow-Diagram Showing the Measured Past and the Future Predicted Inputs and Outputs for Applying MPC

MPC seeks to find the optimal combination of discharge air temperature and flow rate that would provide the desired cooling with minimum energy consumption.

There are several papers in literature that propose methods for improving HVAC controls by focusing on the load shifting strategy of MPC. The load shifting strategy of MPC can be adopted by making use of thermal energy storage devices such as building masses, cooling water tanks etc. The central idea behind the load shifting strategy is to use these storage devices to store thermal energy during off-peak hours when the electricity prices are generally lower and/or the ambient conditions favor a more efficient operation. One of the earliest examples of using predictions for control of building HVAC systems was done in [30]. Different control strategies were tested on a simulation model of simple domestic hot water system with a solar collector. Simulations showed that the control strategies that used weather prediction data performed better than the non-predictive control strategies. In [31] a TRNSYS model of a 5 zone building was created in order to test a predictive control strategy that would optimize a global temperature reference setpoint that used the building thermal storage (passive storage), and the charging and discharging rates of a

chilled water storage tank (active storage) based on predictions of weather data and utility prices. The authors then tested the impact of the accuracy of weather predictions on the effectiveness of the predictive controller in [32]. The authors demonstrated through simulations that with good weather prediction models, a short-term prediction model can realize significant savings when the off-peak and on-peak electricity rates vary. In [4] MPC was implemented on a real building in Prague to find the optimal supply water temperature of a heating system for two zones. The heating system of the building consists of beams placed in the ceilings of the rooms to utilize the building thermal capacity. The authors found that depending on the weather conditions and insulation levels savings of about 15% to 28 % were obtained compared to a non-predictive control algorithm.

In [33], an improvement in performance of building climate control was achieved on a simulation model with the use of MPC coupled with weather predictions. In [15], three main subsystems were modeled – chillers and cooling towers, a thermal energy storage tank, and a campus load model. The control variables that were optimized were the reference temperature exiting the cooling towers, the mass flow rate of the chilled water supply, the reference temperature of the water supplied by the chillers, and the start-up time of the chillers and cooling towers. The performance of MPC was compared to a baseline performance where there was no optimal control involved. An improvement of 19.1% was enabled by implementing MPC. In [34], MPC was implemented on a Swiss office building to compute the optimal setpoints for the supply temperature and the flow rates of the AHU, supply water temperature and operating mode of the Thermally Activated Building System (TABS), and the blind commands. MPC provided better performance than a standard rule-based controller over the course of an entire year of simulation.

The papers discussed above demonstrated the effectiveness of using MPC on certain specific aspects of the HVAC system such as the supply water temperature of the heating system, or the charging and discharging rate of chilled water etc. HVAC systems, however,

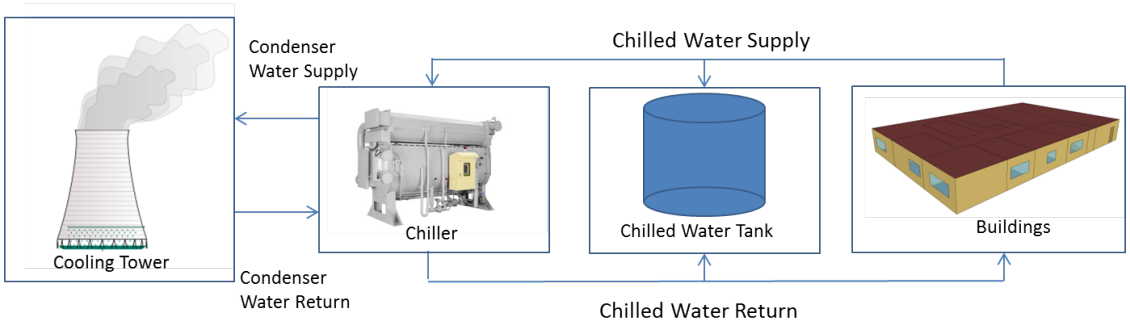
consists of several different components and the type of components used varies from building to building. In order to extract the full potential of energy savings that can be achieved, a methodology for implementing MPC on a comprehensive list of the most common building HVAC components was done in [19]. A brief description of the process adopted in the paper, and the challenges of its implementation that still remain are provided below.

1.3.1 Higher and Lower Level Components

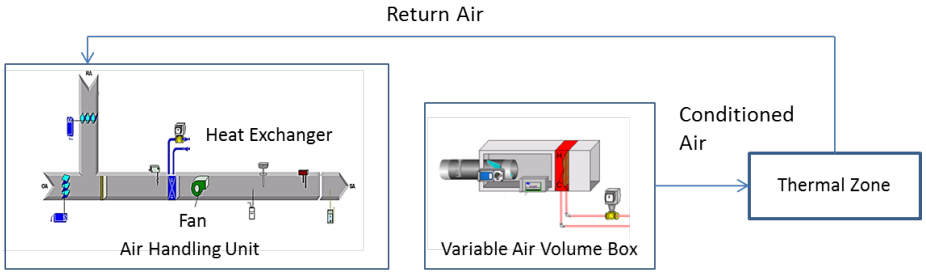
The test bed for the implementation of MPC in [19] is a campus building in Berkeley. MPC is applied by first dividing the HVAC components into an hierarchical structure comprising two levels – an Energy Conversion Level (ECL), and an Energy Distribution Level (EDL). ECL consists of components associated with the generation and distribution of chilled water which is the primary fluid through which cooling energy is supplied to the various buildings on the campus. The components include chillers, cooling towers, chilled water storage tanks, pumps etc. The energy distribution level consists of components associated with the supply and distribution of the conditioned air providing thermal comfort to the occupants of the buildings. The components include the heat exchanger, supply fans, and the thermal zones of the building. Figure 1.7a and 1.7b provide a graphical representation of the two levels of HVAC components. A description of the control strategy and components in each level, and the method that was used to model each of the components [19] is provided below.

1.3.2 High Level MPC

The aim of the higher level MPC is to minimize the energy consumption of the components in its level and at the same time meet the building cooling load requirements. The first step is to design an objective function that expresses the energy consumption and discrepancies in achieving the cooling load requirements in a single unit (generally a $



(a) Energy Conversion Level



(b) Energy Distribution Level

Figure 1.7: Flow Diagram Showing the HVAC Components of the Two Hierarchical Levels

amount) such as in Equation 1.6

$$\boldsymbol{J}_{hl}^{mpc}(t) = \sum_{i=1}^{n_p} \hat{\boldsymbol{J}}_{ch}(t|t+i) + \hat{\boldsymbol{J}}_{ct}(t|t+i) + \hat{\boldsymbol{J}}_{p}(t|t+i) + \hat{\boldsymbol{E}}_{bl}(t|t+i) \tag{1.6}$$

where $\hat{\boldsymbol{J}}_{ch}$, $\hat{\boldsymbol{J}}_{ct}$, and $\hat{\boldsymbol{J}}_{p}$ are the cost of operations of the chiller, cooling tower, and pumps respectively. $\hat{\boldsymbol{E}}_{bl}$ is the penalty on not meeting the cooling load requirements, and n_p is the number of time steps in the prediction horizon over which the objective function is to be minimized. The notation $t|t+i$ is used to indicate an i-step-ahead prediction made at time instant t. The symbol $\hat{}$ is used to indicate estimated values.

MPC seeks to compute optimal control inputs corresponding to ECL represented by

$\boldsymbol{U}_{hl}^{mpc}$, that minimize the objective function above as shown in Equation 1.7

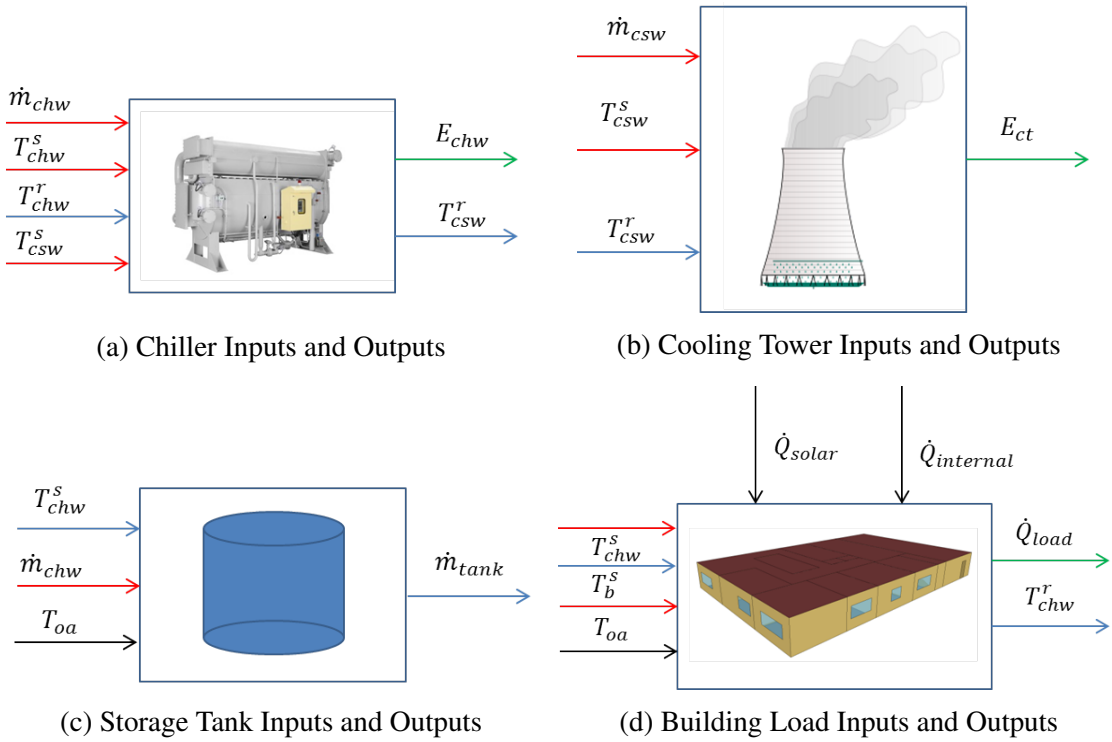
$$\boldsymbol{U}_{hl}^{mpc} = \underset{\boldsymbol{U}_{hl} \in \boldsymbol{D}_{hl}}{argmin}\ \boldsymbol{J}_{hl}^{mpc} \tag{1.7}$$

where $\boldsymbol{D}_{hl}$ is the domain containing all possible values of the control input $\boldsymbol{U}_{hl}$. As is evident from Equation 1.7, in order to compute the optimal control inputs, a relationship between the objective function $\boldsymbol{J}_{hl}^{mpc}$ and the control inputs $\boldsymbol{U}_{hl}$ is required. This relationship is depends on the dynamics of each of the components as shown in Equation 1.8

$$\boldsymbol{J}_{hl}^{mpc} = f_{mpc}(f_{ch}, f_{ct}, f_p, f_{st}) \tag{1.8}$$

where f_{mpc} is the relationship between the objective function and the outputs being modeled, f_* is the function representing the dynamics of the component $*$ (ch = chiller, ct = cooling tower, p = pumps, st = storage tank) that provides a relationship between the outputs and the control inputs $\boldsymbol{U}_{hl}^{mpc}$.

The next step in the application of MPC is to determine low order relationships f_* for each component. The relationship f_* is generally expressed in terms of control and disturbance inputs, and the output being modeled. The previous section provided modeling approaches that can be used to identify the relationships f_*. This section provides a description of what relationships need to be identified corresponding to each HVAC component. The control and disturbance inputs, and outputs corresponding to each component of ECL is shown in Figure 1.5. The red and black arrows correspond to control and disturbance inputs respectively, and the blue and green arrows correspond to outputs used in the construction of the MPC objective function, and outputs which help in identifying interactions between neighboring subsystems, respectively. A brief description of how the components of ECL were modeled in [15] is provided below.



(a) Chiller Inputs and Outputs

(b) Cooling Tower Inputs and Outputs

(c) Storage Tank Inputs and Outputs

(d) Building Load Inputs and Outputs

Figure 1.8: Inputs and Outputs of the Components in the Energy Conversion Level

1.3.2.1 Chiller

The outputs of interest of the chiller system, are the CHW supply temperature, the chiller energy, and the condenser water return temperature. A static model of the chiller is used in [15], where the CHW supply temperature is assumed to be equal to the CHW setpoint temperature. In order to model the chiller energy consumption several simulations of a high fidelity model of the chiller were run under different operating conditions. The simulation results were used to construct a 5-d lookup table where the output was the chiller energy and the inputs were the CHW setpoint temperature T^{set}_{chw}, the mass flow rate of CHW $\dot{m}_{chw}$, the supply condenser water temperature T_{cws}, the CHW return temperature T_{chwr}, and the outside air wet bulb temperature T_{wb}. The condenser water return temperature was computed using the conservation of energy principle where the heat lost by the

CHW is equal to heat gained by the condenser water. The control inputs that affect the energy consumed by the chiller are the mass flow rate and temperature of the chilled water, and the supply temperature of the condenser water. In [19], the relationship between the inputs and the outputs of the chiller were not found using any of the linear parametric method described in the previous section. Instead a nonlinear static function was used. The static function in the form of a 5-d lookup table was obtained from [15] by running several simulations of a high fidelity model of the chiller corresponding to different operating conditions.

1.3.2.2 Cooling Tower

The cooling tower consists of a fan whose speed is regulated to meet a certain condenser water supply temperature. The outputs of interest of the cooling tower are the condenser water supply temperature, and the cooling tower fan energy consumption. As was done with the chiller system, a static model was again considered by the authours in [15]. The supply condenser water temperature is assumed to be equal to its setpoint temperature. The local controller in the cooling tower regulates the cooling tower fan to supply condenser water at the desired temperature. A similar approach as was adopted for the chiller model is used to model the energy consumed by the cooling tower fan. Regression analysis was performed to identify coefficients of a fifth degree polynomial that provided a relationship between fan speed and T_{cws}, T_{cwr}, T_{wb}, and mass flow rate of the condenser water $\dot{m}_{cws}$ [35]. Regression entailed the determination of 27 parameters. The cooling tower fan energy was then approximated as a cubic function of the fan speed.

1.3.2.3 Cooling Water Storage Tank

The dynamics of the cooling water storage tank were modeled in [15] by first assuming it to contain two lumped masses, warmer water at the top with a height z_a and temperature of T_a, and colder water at the bottom with height z_b and temperature T_b. The outputs of

interest are the heights and temperature of the warm and cold lumped masses. The outputs are computed as a function of $\dot{m}_{chws}$, T_{chws}, the mass flow rate and the return temperature of CHW supplied to the building $\dot{m}_{cmp}$, and $T_{cmp,r}$.

1.3.2.4 Building Load

Building loads in [15] are modeled using the RC equivalent circuit method described in Section 1.2.2. In [15], the building being investigated was divided into two zones, and the capacitance and resistance values of the equivalent RC circuit were determined through nonlinear regression analysis. In order to increase the accuracy of the model, measurements of the incident solar radiation were also made.

1.3.3 Low Level MPC

The components considered in EDL were the cooling coils, AHU fan, and the thermal zones. The objective of the low level MPC is to minimize the energy consumption of the cooling coil and fan, at the same time maintain occupancy comfort. As was done with the high level MPC, an objective function is first designed that penalizes energy consumption, and deviations of predicted temperatures from set point values over a horizon as shown in Equation 1.9

$$\boldsymbol{J}_{lll}^{mpc}(t) = \sum_{i=1}^{n_p} \hat{\boldsymbol{J}}_{cc}(t|t+i) + \hat{\boldsymbol{J}}_{fan}(t|t+i) + \hat{\boldsymbol{J}}_{comf}(t|t+i) \tag{1.9}$$

where $\hat{\boldsymbol{J}}_{cc}$ and $\hat{\boldsymbol{J}}_{fan}$ are the energy consumed by the cooling coil and the AHU fan respectively, and $\hat{\boldsymbol{J}}_{comf}$ is a penalty for not meeting the thermal comfort requirements.

The control inputs that were optimized in [19] were the AHU discharge air temperature, the recirculation damper position, the volume flow rate across the VAV boxes and the CHW vavlve position of the AHU heat exchanger. The optimal control inputs were

computed by minimizing the objective function in Equation 1.9. A brief description of the modeling methods employed in [15] to compute the optimal inputs is provided below.

1.3.3.1 Cooling Coil

The output of interest of the cooling coil model is the cooling energy required to cool the air through the AHU. In [15], a constant efficiency cooling coil model is assumed. The energy expended is proportional to the cooling load on the air side of the heat exchanger. Thus the cooling coil energy is obtained as a function of the mass flow rate of the air through the AHU $\dot{m}_{ahu}$, the inlet air temperature $T_{a,in}$, and the discharge air temperature T_{chw}. Since a static model is assumed, the discharge air temperature is assumed to be equal to its setpoint T_{chw}^{set}.

1.3.3.2 Fan Model

The output of interest in the fan model is the energy consumed by the fan. In [15] the fan energy is assumed to be a quadratic function of the mass flow rate $\dot{m}_{ahu}$. The parameters to the quadratic function were determined through regression analysis with measured data.

1.3.3.3 Thermal Zone

The thermal zones in [15] were modeled as an RC equivalent circuit described in Section 1.2.2. The output of interest in the thermal zone models is the temperature of the zone.

The description of the objective function, and the component models showed the steps taken in [15] in order to implement MPC on a real building. The models of the components of ECL were developed using nonlinear regression of data obtained from either experiments or high fidelity simulation models. The models and the experimental data were specific to the building under study. As the experiments and simulation models cor-

responded specifically to the building on which MPC was being tested, the process cannot be adopted for automation and wide spread implementation.

When considering MPC on only the EDL components, gray-box models were used in modeling the temperature of the zones. For the reasons specified in Section 1.2.2 gray-box models are not feasible for the application of MPC being automated. Chapter 3 of the thesis makes a contribution towards automating the process of model development of the components of EDL, and in Chapter 4, MPC is applied on a high-fidelity simulation model to demonstrated the usefulness of the developed models for model-based control.

1.4 Modeling Architecture

The previous section described how MPC has the potential to improve the efficiency of building HVAC systems, and also the bottlenecks that still remain in its widespread implementation. This dissertation seeks to address the challenges in implementing MPC by proposing an automated modeling method, and verifying its use in model-based control on a high-fidelity simulation model.

Another question that this dissertation seeks to address is the control architecture that is most suitable for the implementation of MPC. Implementing MPC on building HVAC systems requires regulators that perform a large variety of operations. Some of these operations include storing the identified models of the subsystems, receiving real-time data from the BEMS sensors, updating the model states, computing the optimal control trajectory, relaying the computed setpoints to the actuators etc. Due to the large scale of building HVAC systems, performing the aforementioned operations results in a big storage, computational, and communication burden on the regulators. Depending upon the number of regulators used, and how information is exchanged between them, the control architecture used to solve the MPC problem can be broadly divided into three categories namely centralized, decentralized and distributed. A description of the architectures is

provided below.

1.4.1 Centralized MPC

Figure 1.9 shows an example of a centralized control architecture used to implement MPC on building HVAC systems. A centralized architecture consists of a single regulator that solves the MPC problem of the entire system. The MPC problem corresponding to a centralized architecture constitutes computing optimal setpoints over the prediction horizon for each subsystem by minimizing a single centralized objective function. For example in Figure 1.9, a single regulator is used to find the optimal setpoints for the chiller subsystem T_{chw}^{set}, AHU heat exchanger T_{ahu}^{set}, fan subsystem P_{eds}^{set}, and the thermal zones T_i^{set}, where $i \in 1, 2, ...n_r$ and n_r is the number of thermal zones. A list of the tasks performed by the regulator is shown in Table 1.1.

Table 1.1: Tasks Performed by the Regulator in the Centralized MPC Architecture

Task	**Description**
Storage	Storing Subsystem Models and historical data
Computation	Update system states for each subsystem
Computation	Compute optimal setpoints for each subsystem
Communication	Output data from BMS sensors to Regulator
Communication	Setpoint values from regulator to actuators

Implementation of centralized MPC for large scale systems such as building HVAC becomes difficult as the complexity of each of the tasks increases exponentially as the number of subsystems increase. A centralized architecture also results in reliability and robustness problems [36]. For example if one of the components of the HVAC system is replaced, the entire centralized model and objective function needs to be updated.

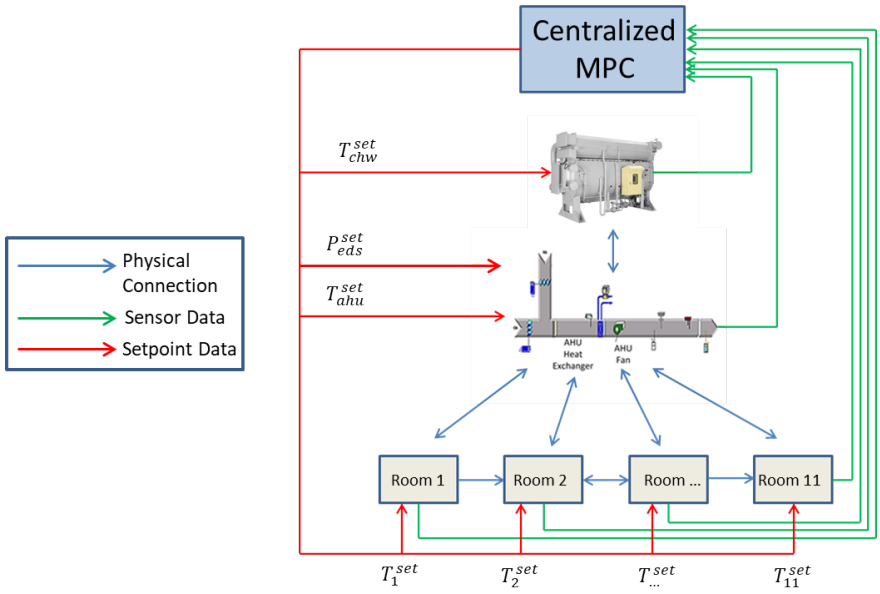


Figure 1.9: Layout Showing a Centralized MPC Architecture

1.4.2 Decentralized MPC

The layout of a decentralized control architecture is shown in Figure 1.10. In a decentralized control architecture there is a regulator associated with each component of the HVAC system. There is however, no information exchange that occurs between the regulators. Each regulator optimizes the component to which it has been assigned. A decentralized control architecture may work well only in the cases where the coupling between the dynamics of the various subsystems is small.

1.4.3 Distributed MPC

Figure 1.11 shows a layout of the distributed control architecture. A distributed architecture is similar to decentralized, except that some amount of information is exchanged between the regulators of each subsystem. The information that is exchanged between the regulators generally consists of data corresponding to the interacting dynamics of the subsystems. Depending upon the protocol used to exchange information, the distributed

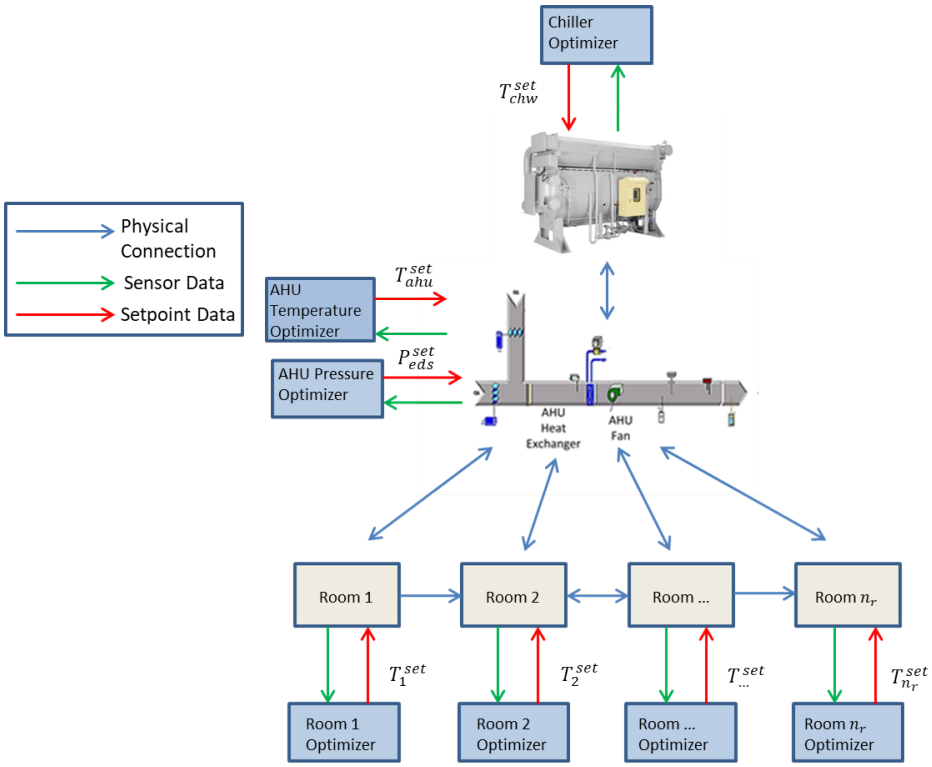


Figure 1.10: Layout Showing a Decentralized MPC Architecture

control architecture can be divided into non iterative and iterative systems [36]. In a non iterative system information is exchanged between the regulators only once every time step, whereas, in an iterative system information is exchanged several times in a single time step.

The distributed control architecture can also be divided according to the type of objective function that is minimized by the local regulators. Each regulator may minimize its own objective function (independent algorithm) or all regulators may minimize a single global objective function (cooperating algorithms) [36]. An independent distributive MPC may reach a Nash equilibrium which is not a global optimal solution. In a Nash equilibrium each subsystem is at its own local optimal position. Once a Nash equilibrium point is reached, a global optimal position cannot be reached since that would require the operat-

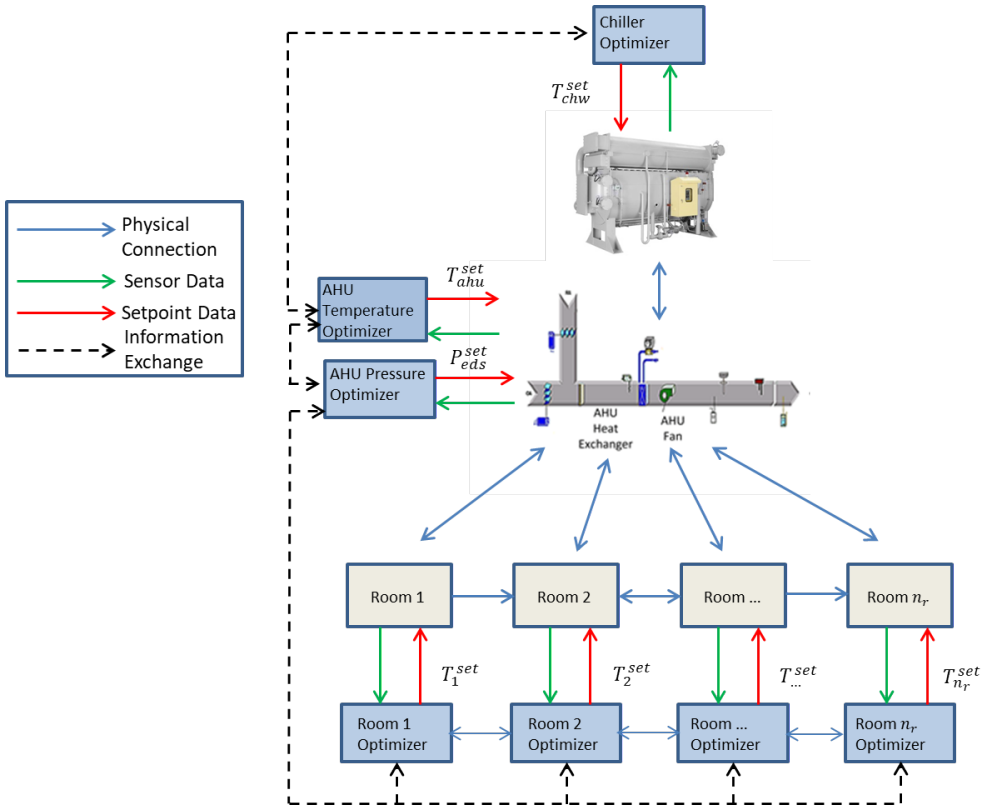


Figure 1.11: Layout Showing a Distributed MPC Architecture

ing condition to move away from the local optimum position thereby increasing the local cost function. An iterative independent distributive MPC was implemented in a simulation environment in [37]. An RC-equivalent gray-box model of a three zone building was used as the simulation test bed. Regulators corresponding to each thermal zone minimized their own local objective function to compute the optimal heating. Information of the predicted temperatures over the horizon was exchanged between the regulators of the neighboring thermal zones.

An alternative iterative setup called a cooperative distributive control was proposed in [38] and [39] where all the subsystems minimize a single global objective function. In order to achieve the global optimal solution, however, each regulator requires the model of

every other component in the system, and every iteration requires communication of the future optimal control set and states. In [40] a method called as Neighbor Communication Optimal method (NC Opt) to reach the global optimal solution was proposed. The method uses an iterative independent distributed approach. The authors were able to show that by judiciously designing the objective function of each component a global optimal solution in steady state can be achieved by exchanging information between regulators of neighboring subsystems. Each subsystem solves its own objective function. The cost function of the local subsystems is modified to include penalites based on how the output of the current local subsystem affects its neighbor. This method vastly reduces the communication burden of the conventional cooperative distributed algorithms.

The authors in [40] showed that a Pareto optimal solution can be obtained in steady state by judiciously designing the objective function of each component. In this dissertation, the algorithm proposed in [40] is adopted to apply dynamic MPC with a distributed control architecture. Chapter 5 of the dissertation provides the results of applying MPC with the distributed architecture on a high fidelity simulation model of an office building. Simulation results are compared with that of centralized MPC.

1.5 Organization of Dissertation

The remainder of the dissertation is organized as follows. Chapter 2 describes the phenomenon of hunting in building HVAC systems. The results of a survey are presented which shows the widespread nature of the phenomenon. An automated modeling algorithm is proposed and verified to detect hunting in real time. An alternative control architecture is offered as a solution along with a simulation example showcasing how the architecture can reduce the occurrence of hunting. Chapter 3 proposes an automated black-box modeling algorithm. The algorithm is then verified using data from BEMS of a real office building. Simulations of a high fidelity model of the office building are per-

formed where MPC uses the models developed from the proposed algorithm. Three types of black-box approaches are compared both in their prediction accuracy and their ability to be used in model-based control such as MPC. Chapter 4 provides a comparison of two types of control methodologies, the traditional PI-type control, and MPC. A description of the simulation environment used for the comparison, and the steps taken by MPC is also presented. Chapter 5 provides a description of an alternative control architecture through which MPC can be applied on building HVAC systems. A comparison of the performance achieved by the two types of control architectures in a simulation environment are also presnted. Conclusions and recommendations for future work are provided in Chapter 6.

2. UNDESIRED OSCILLATIONS IN HVAC SYSTEMS

[1] Some of the reasons for suboptimal performance of HVAC systems using conventional PI-type control algorithms were introduced in the previous chapter, undesired oscillations or hunting, the decentralized architecture, and lack of planning for future operating conditions. This chapter seeks to address one of those reasons for the inefficient operation, specifically undesired oscillations or hunting of HVAC components.

The chapter is organized as follows. First, a discussion of the probable causes for hunting is provided, followed by a methodology for identifying hunting in real-time. The algorithm is then applied to data obtained from the BEMS of 10 buildings at the Texas A&M university campus. The results of a survey documenting the prevalence of hunting in those buildings is provided. Finally a solution to the problem of hunting is offered in the form of an alternative control architecture, called cascaded control. A simulation example demonstrating how cascaded control can reduce hunting is also provided. The chapter is a summary of the joint work that was performed by the author and Christoper Price [5], [41]. From the aforementioned topics included in the chapter, the unique contribution of the author comprises the algorithm to identify the presence of hunting from real-time data, and the survey of HVAC components to show the prevalence of hunting in buildings.

2.1 Undesired Oscillations or Hunting

Most buildings employ PI-type control for HVAC systems, mainly due to the simplicity of its implementation [42]. The control outputs are easy to compute, and the data required for computation are easily obtained. HVAC systems, however, are time varying and inherently nonlinear. For example, consider the relationship between the steady state value

[1] Part of the data reported in this chapter is reprinted with permission from "Identification and elimination of hunting behavior in HVAC systems," by Chintala,R., Price, C., & Rasmussen, BP., 2015. *ASHRAE TRANSACTIONS*, 121, 294, ©2015 ASHRAE TRANSACTIONS.

of room temperature and the damper of the Variable Air Volume (VAV) box serving the room. Figure 2.1 shows the rate of change of the steady state value of the the temperature of room i (T_i) per percentage change in the damper position ($\delta T_i/\delta D_i$), as a function of the damper position D_i. As is evident from the figure, the steady state gain $\delta T_i/\delta D_i$ is much higher when the damper opening is small compared to when the opening is large.

In the traditional control architecture, damper position is regulated by PI control to track a given room temperature setpoint. The gains of the PI controller are generally designed for a fixed operating condition. If these design conditions correspond to when the damper opening is large, the PI gains would tend to be high as more control effort is needed to make a change in the steady state value of the room temperature. Over the course of normal operation, however, operating conditions change and the large PI gains may cause the measured temperature to oscillate about the set point temperature instead of accurately tracking it. The oscillations of the measured temperature in turn results in the oscillations of the damper position as it is dependent on the error between the measured and reference temperatures. These undesired oscillations in the output and input under non-design conditions are called hunting.

The example above showed how inherent nonlinearity in the system components can cause hunting. In addition to process nonlinearity, hunting can occur due to degradation in components such as valve stiction [43]. Thus presence of hunting can also be an indication that some of the system components may have to be replaced.

Other than nolinearity, there are several other causes of undesired oscillations in building control loops such as improper control tuning, oscillating disturbances or oscillations in neighboring control loops etc. For example, in the example above oscillations in the VAV damper due to high PI gains causes an oscillation in the measured room temperature. A cooling demand is generally calculated as a function of the error between the measured and reference room temperatures. The cooling demand is used in the calcula-

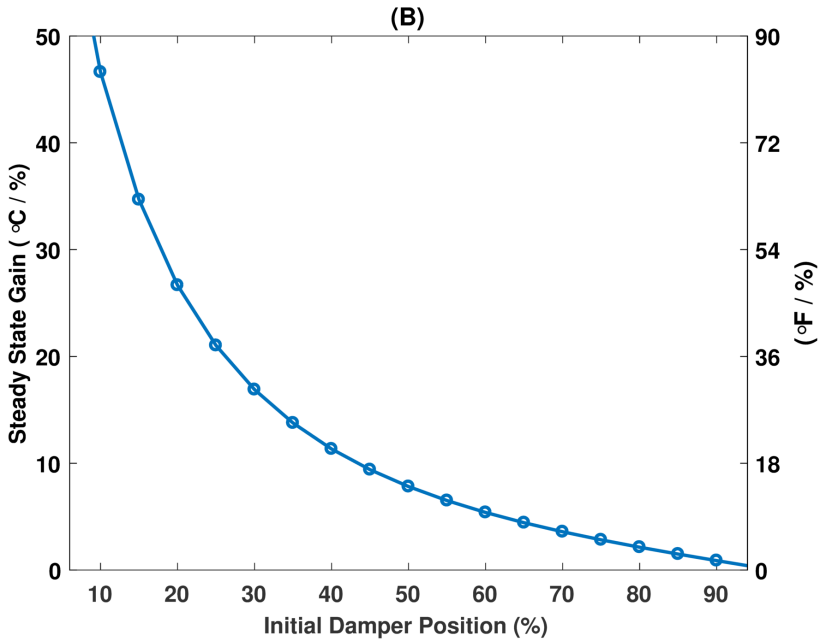


Figure 2.1: Volume Flow Rate as a Nonlinear Function of Damper Position

tion of the AHU discharge air temperature. Oscillations in measured room temperature may cause oscillations in the cooling demand calculation which in turn cause oscillations in the CHW valve of the AHU which regulates the discharge air temperature. The example shows how oscillations in one control loop can cause oscillations in another. Such oscillations are undesirable since they result in increased energy consumption such as in fans, component wear and tear in valves and dampers, and unsatisfactory performance in regulating the desired signals.

2.2 Detecting the Presence of Hunting

As is stated in the previous section, hunting causes increased energy consumption, and wear and tare of components of valves and dampers. In addition, hunting also results in poor control performance. Hunting causes oscillations in the value of the output thereby not tracking the assigned set point value accurately [44]. Hence preventing the occurrence of hunting is essential for a more efficient operation of building HVAC systems.

The first step in reducing the occurrence of hunting, however, is to detect its presence. Building HVAC systems contain hundreds of control loops and components that may be susceptible to hunting. Individually monitoring each of theses components is infeasible. Several methods of detecting oscillations have been presented in literature [43], [44],[45], [46]. A brief discussion of one of these methods is provided below.

Detection of hunting requires more than just checking for the presence of oscillations. Not all oscillations in control loops and the components involved are undesired. Some oscillations may occur due to the presence of oscillating disturbances which would require the control actions itself to be oscillatory. For example, the cooling load on buildings is influenced by outside weather conditions which for the most part are oscillatory with a time period of one day. The algorithm for detecting hunting should be able to distinguish between oscillations which are necessary to meet the objectives of the control system, and the undesired oscillations or hunting behavior.

The available literature on hunting mostly deals with process industries. The principal obstacles of detecting undesired oscillations in process industries is the fact that the frequency of oscillations vary widely, and the oscillations are not necessarily sinusoidal. In [43], Hagguland proposed a simple online oscillation detection algorithm to identify the undesired oscillations in control loops in process industries. The algorithm proposed by Hagguland is mainly used to detect the presence of stiction in valves. Hagglund argues that in process industries, PI controllers are conservatively tuned, and is unlikely for poor controller tuning to cause hunting. The oscillations due to stiction in valves correspond to low and mid range frequencies. The undesired oscillations are thus detected by placing an upper and lower bound on the frequencies of oscillations of a certain aspect of the operation. The frequencies of oscillations within these bounds are identified as hunting. The steps taken by Hagglund to detect these frequencies is provided below.

In order to detect the oscillations, the algorithm looks at the behavior of the control

error $e(t)$ defined in Equation 2.1

$$e(t) = y_r(t) - y_m(t) \tag{2.1}$$

where $y_r(t)$ is the reference signal, and $y_m(t)$ is the measured output at time t. The error signal is first used to compute an Integrated Absolute Error (IAE) denoted by I_e , which is the integral of the error signal between two successive zero crossings as shown in Equation 2.2

$$I_e = \int_{t_{i-1}}^{t_i} e(t)dt \tag{2.2}$$

where t_{i-1} and t_i correspond to the time instances of the zero crossings.

The algorithm then looks into two aspects of IAE to determine whether there are undesired oscillations in the control loop, namely, the amplitude and frequency. If the control performance is good, then the time elapsed between successive zero crossings of the error signals is small thereby resulting in a small amplitude of IAE. A threshold value denoted by I_e^{lim} is selected, and a load disturbance is said to have occurred whenever the IAE is greater than the threshold, i.e. $I_e > I_e^{lim}$. I_e^{lim} also serves as an upper bound on the frequencies of oscillations detected by the algorithm. When the frequency of oscillation is high, there is less time elapsed between the zero crossings resulting in a smaller IAE value. Hence, high frequency oscillations which have an IAE value less than I_e^{lim} are excluded.

The second aspect that the algorithm looks at is the frequency of the load disturbances. As stated in the preceding paragraph I_e^{lim} places an upper bound on the frequency of load disturbances. In addition to the upper bound, a lower bound is also placed on the frequency of load disturbances. This is done by selecting another design parameter n_{lim} which sets the lower threshold of the number of load disturbances that need to occur over

an observation period T_{obs} before the oscillations can be classified as hunting.

The simplicity of the algorithm lies in the fact that it requires just two parameters to be specified by the user. The user just needs to determine the range of frequencies which could possibly occur due to the presence of stiction in valves.

2.2.1 Hunting in Building HVAC Systems

The previous section provided a description of how hunting can be identified in process industries. The main obstacle in detecting hunting in process industries is that the operations may be subjected to a wide range of frequencies, which makes it difficult to distinguish between desired and undesired oscillations. This dissertation places a focus primarily on the oscillations that are present in building HVAC systems. An argument is made in this chapter that detecting hunting in building HVAC systems is much easier than in process industries as they are not subject to a wide range of oscillating disturbances.

As is stated in Section 2.2, the hunting algorithm should be able to distinguish between oscillations in the control loop due to load disturbances and the undesired oscillations. Oscillations due to load disturbances are part of the control system design. The main sources of load disturbances that occur in building HVAC systems are due to the changes in outside weather conditions, and internal heat loads. Oscillating disturbances due to outside weather conditions have a time period of 1 day, whereas internal heat loads due to occupancy, equipment etc. are non-oscillatory in nature. Internal heat loads generally contain large periods of constancy interspersed with step changes in values. Control loops responding to these disturbances would also exhibit similar behavior.

Figure 2.2 shows oscillations of the fan speed and the CHW valve of an AHU serving one of the buildings at the Texas A&M University campus. As is evident from the figure, the CHW valve and fan speed oscillate with two distinct frequencies. The time period corresponding to the fan speed oscillations is 1 day and the time period corresponding

to the CHW valve is approximately 1 hour. The oscillations in fan speed occur in order to meet the cooling demands of the building which oscillate at the same frequency of 1 day due to external weather. There are however, no load disturbances that oscillate with the frequency corresponding to the CHW valve. These oscillations are thus most likely a result of hunting behavior.

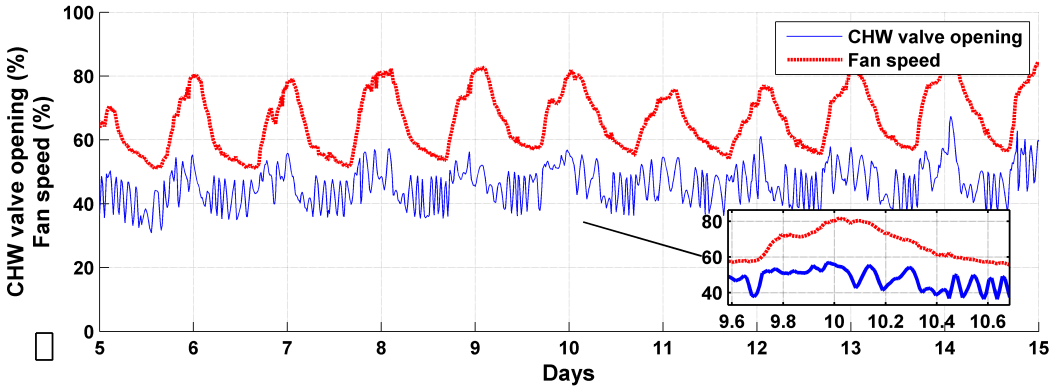


Figure 2.2: Oscillations Showing Fan Speed and CHW Valve Having Different Frequencies

2.2.2 Algorithm to Detect Hunting in Building HVAC Systems

In this chapter, a simple algorithm is proposed which detects the presence of the hunting in building HVAC systems. There are two aspects which make the algorithm much easier than the one proposed by Hagguland in [43] for detecting hunting in process industries. First, the distinct nature of the frequency of the load disturbances makes it easier to separate the desired and undesired oscillations. Secondly, the algorithm uses just the measured values of the process variables, instead of using the error function as was proposed by Hagguland in [43] which would require that the set point values of the variables also be recorded by the BEMS.

The proposed algorithm looks into two aspects of the process variable being studied in order to determine whether the oscillations corresponding to it constitute hunting, namely, the amplitude and frequency. There are three criteria based on the amplitude and frequency of oscillations that are placed by the hunting detection algorithm. First, the peak-to-peak amplitude of the oscillations should be greater than a certain design value A_{lim}. If the peak-to-peak magnitude of the oscillation is less than A_{lim} then the oscillations are considered to be negligible for the purpose of detecting hunting. The second criteria is that there needs to be a minimum number of oscillations n_{lim} in a certain observation period T_{lim} for the oscillations to be detected by the algorithm. This criteria is included to eliminate oscillations due to random disturbances such as occupancy. The third and final criteria for the algorithm to detect hunting is that the frequency of oscillation f_{osc} must be greater than $\frac{1}{day}$. A detailed description of the algorithm is provided as follows.

The first step of the algorithm is the introduction of three iteration variables, namely, i_{hunt} which is the index of the starting point of the data set being investigated for hunting, i_{osc} which is the index for the starting point of a subset of the data being studied for the presence of oscillations, and j which is the index of the current data point being investigated. All three iteration variables are initialized to 1.

In the second step of the algorithm, four more variables are initialized, $A_{max} = 0$, $A_{min} = 0$, $n_{osc} = 0$, and $n_{sgn} = 0$. The variables A_{max} and A_{min} keep track of the maximum and minimum value of the process variable in the subset which is being studied for the detection of an oscillation. After the detection of the oscillation, the variables are reset to 0. The variable n_{osc} keeps track of the number of oscillations that have been detected in the observation period T_{obs}. The variable n_{sgn} keeps track of the number of sign changes in the process variable. The value of n_{sgn} is reset to 0 every time an oscillation is detected.

The algorithm then proceeds by determining the sign (positive or negative) of the

change in the value of the process variable between two successive time instances. The sign corresponding to the current time instant j is determined by the variable $S_{new} = Y_{j+1} - Y_j$ and the sign corresponding to the previous time instant $j-1$ is stored in the variable $S_{old} = Y_j - Y_{j-1}$. The symbol Y represents the process variable being studied, and the subscript j denotes the time step. A sign change is said to have occurred if S_{new} and S_{old} have opposite signs. When a sign change occurs the value of the variable n_{sgn} is incremented by 1.

An oscillation constitutes three sign changes. Hence when the value of n_{sgn} is 3, an oscillation is said to have occurred. When detecting a new oscillation, the time step at which the first sign change occurs is assigned to the variable i_{osc}. If j is the index of the data point at which n_{sgn} is incremented to 3, then all the data points from the index i_{osc} to j are considered to be part of the oscillation. The value of the variables A_{min} and A_{max} are assigned the maximum and minimum values of the process variable in the time span between i_{osc} and j. If the peak-to-peak amplitude is greater than the threshold value A_{lim}, then the oscillation is considered significant and the variable n_{osc} is incremented by 1.

When the number of oscillations detected is equal to 2, then the algorithm checks if the 2 oscillations occurred within the observation period T_{lim}. If yes, then hunting is said to have occurred, and all the data points within the two oscillations (i_{hunt} to j) are included in the set of data points constituting hunting. When the next sign change occurs, the index i_{hunt} and i_{osc} are assigned the value j corresponding to the data point under consideration.

If the second oscillation is detected after T_{lim}, however, hunting is not said to have occurred yet. The variable i_{hunt} is assigned the index corresponding to the start of the next oscillation. The process continues till the entire data set is analysed.

A summary of the steps taken to detect the algorithm is provided below. Figure 2.3 contains a schematic representation of the algorithm.

- *Step 1:* Initialize $i_{hunt} = 1$, $i_{osc} = 1$, $j = 1$.

- *Step 2:* Initialize $A_{max} = 0$, $A_{min} = 0$, and $n_{osc} = 0$.

- *Step 3:* Compute the variables used in the detection of sign changes in the difference of successive values of the process variables.

$$\begin{aligned} S_{old} &= S_{new} \\ S_{new} &= Y_{j+1} - Y_j \end{aligned} \tag{2.3}$$

- *Step 4:* Check if S_{old} and S_{new} have the same sign. If they have a different sign then increment the value of the variable n_{sgn} as shown below.

$$n_{sgn} = \begin{cases} n_{sgn} & (S_{old} > 0 \;\&\&\; S_{new} > 0) \;||\; (S_{old} < 0 \;\&\&\; S_{new} < 0) \\ n_{sgn} + 1 & (S_{old} < 0 \;\&\&\; S_{new} > 0) \;||\; (S_{old} < 0 \;\&\&\; S_{new} > 0) \end{cases} \tag{2.4}$$

- *Step 5:*: If the number of sign changes = 3 then an oscillation is said to have occurred. The peak-to-peak value of the oscillation is computed by determining the maximum and minimum values of the process variable in the oscillation as shown below.

$$A_{max} = \begin{cases} A_{max} & n_{sgn} < 3 \\ max\{Y(i_{osc}), Y(i_{osc}+1), ..., Y(j)\} & n_{sgn} = 3 \end{cases} \tag{2.5}$$

$$A_{min} = \begin{cases} A_{min} & n_{sgn} < 3 \\ min\{Y(i_{osc}), Y(i_{osc}+1), ..., Y(j)\} & n_{sgn} = 3 \end{cases} \tag{2.6}$$

- *Step 6:* Increment the number of oscillations if the peak-to-peak value of the oscillation is greater than the threshold. If an oscillation has occurred but the peak-to-peak

value is less than the threshold then reset the value of the iteration variables i_{osc} and i_{hnt} as shown below.

$$\begin{cases} n_{osc} = n_{osc} + 1 & A_{max} - A_{min} > A_{lim} \;\&\&\; n_{sgn} = 3 \\ i_{osc} = j,\ i_{hunt} = j & A_{max} - A_{min} < A_{lim} \;\&\&\; n_{sgn} = 3 \\ A_{min} = 0,\ A_{max} = 0 & A_{max} - A_{min} < A_{lim} \;\&\&\; n_{sgn} = 3 \end{cases} \tag{2.7}$$

- *Step 7:* If 2 oscillations occur within the observable time period T_{lim} then hunting is said to have occurred, and the hunting data set Y_{hunt} is augmented as shown below.

$$\begin{cases} Y_{hnt} = [Y_{hnt}, \{Y(i_{hnt}), Y(i_{hnt}+1), ..., Y(j)\}] & n_{osc} = 2 \;\&\&\; t < T_{lim} \\ i_{hunt} = i_{osc}, i_{osc} = j & n_{osc} = 2 \;\&\&\; t > T_{lim} \\ A_{min} = 0, A_{max} = 0, n_{osc} = 1 & n_{osc} = 2 \;\&\&\; t > T_{lim} \end{cases} \tag{2.8}$$

- *Step 8:* Increment the iteration variable $j = j + 1$. Repeat steps 3 through 8, until data set is analyzed.

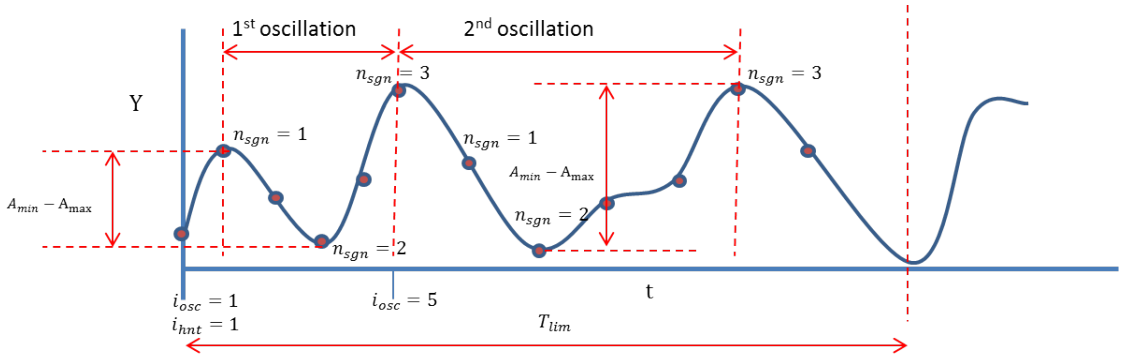


Figure 2.3: Flow Diagram Showing the Application of the Hunting Algorithm

2.2.3 Application of the Algorithm on Real Building Data

A description of the proposed algorithm to detect hunting in real time for process variables corresponding to building HVAC systems is provided in the preceding subsection. In this subsection, the algorithm is applied on data obtained from BEMS of a real building. The process variable on which the hunting algorithm is applied is the CHW valve opening. The BEMS sensors sample data once every 15 minutes. CHW valve data was collected for a period of 20 days. The two design parameters of the algorithm A_{lim} and T_{lim} are assigned the values 15% and 2 hours, respectively. Figure 2.4 shows the results of applying the algorithm. The hunting algorithm identifies the times when oscillations with a peak-to-peak magnitude of greater than 15% occur.

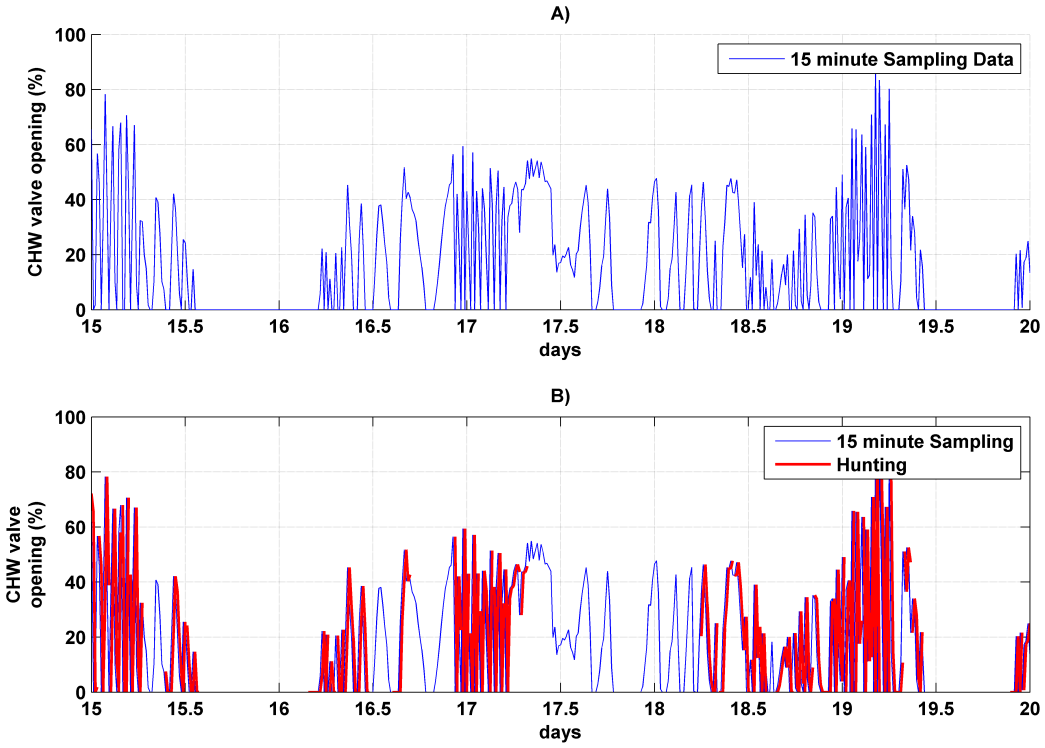


Figure 2.4: Results of Applying the Algorithm on CHW Valve Opening Data Sampled Once Every 15 Minutes

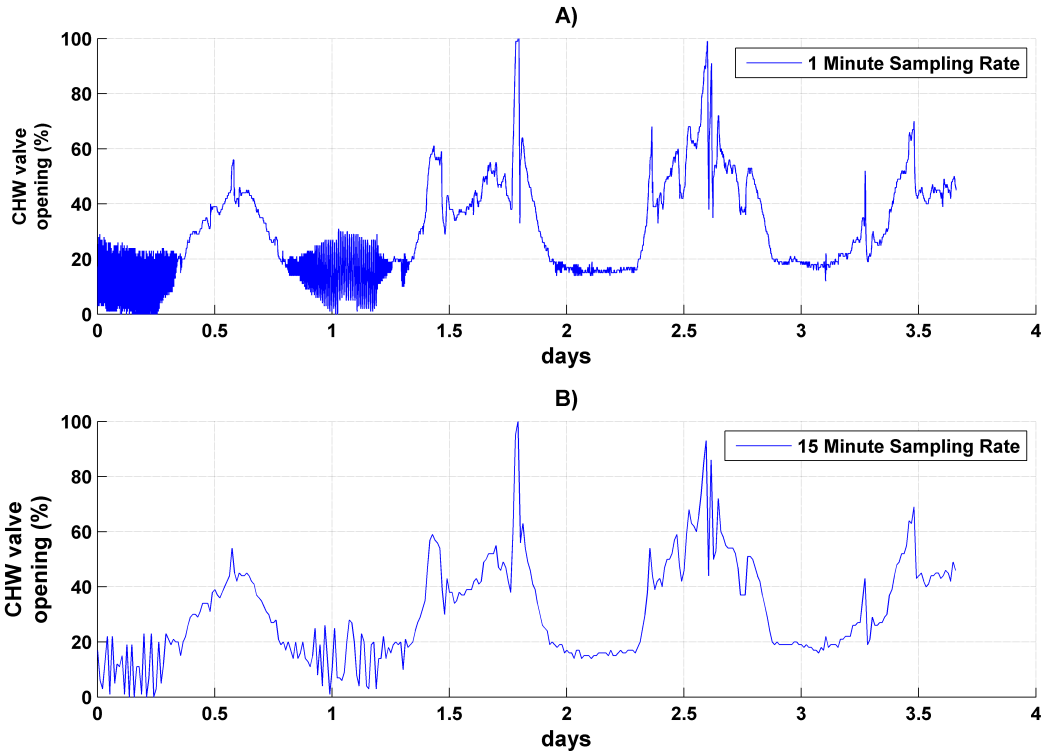


Figure 2.5: CHW valve Opening Data Sampled at 1 Minute and 15 Minute Intervals

The preceding paragraph shows the results of applying the data that is sampled at 15 minute intervals. A slow sampling rate may lead to aliasing of the property being measured. A 15 minute sampling rate may not be able to completely capture the dynamics of a higher frequency due to aliasing. In order to determine whether the proposed algorithm can identify the presence of hunting when the sampling frequency is high, a comparison is made between the results obtained by applying the algorithm on two data sets measuring the same process variable but with different frequencies. The CHW valve opening of an AHU is sampled at 1 minute intervals for a duration of 4 days. Another set of data was created by taking every 15th data point of the data set sampled at 1 minute. Figure 2.5 shows the CHW opening data corresponding to sampling rates of 1 minute and 15 minutes. The hunting algorithm is applied to both sets of data. The design parameter A_{lim}

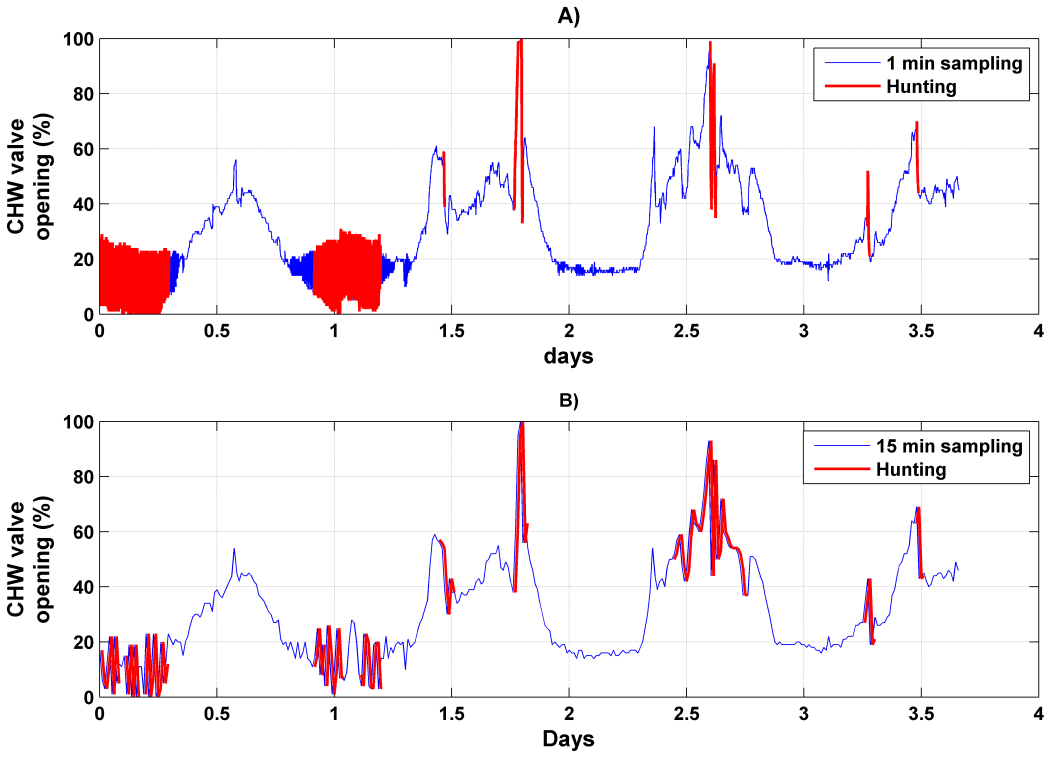


Figure 2.6: Results of Applying the Hunting Algorithm on CHW Valve Opening Data Sampled at One Minute and 15 Minute Intervals

for both sets of data is assigned 15%, and the parameter T_{lim} is assigned a value 2 hrs for the data sampled at 15 minute intervals, and 8 minutes for the data sampled at one minute intervals. The values 2 hrs and 8 minutes both correspond to 8 sample steps of their respective data sets. Figure 2.6 shows the periods of operation where hunting is identified by the algorithm in the one minute and 15 minute sampling data. Although, the 15 minute data does not capture all the dynamics of the oscillation, the algorithm is still able to identify the periods of time when hunting is observed.

In order to demonstrate how hunting in one control loop can cause hunting in another control loop, the alogrithm is applied on data collected over period of 60 days of two process variables, the average damper opening of the VAVs of all rooms, and the AHU fan

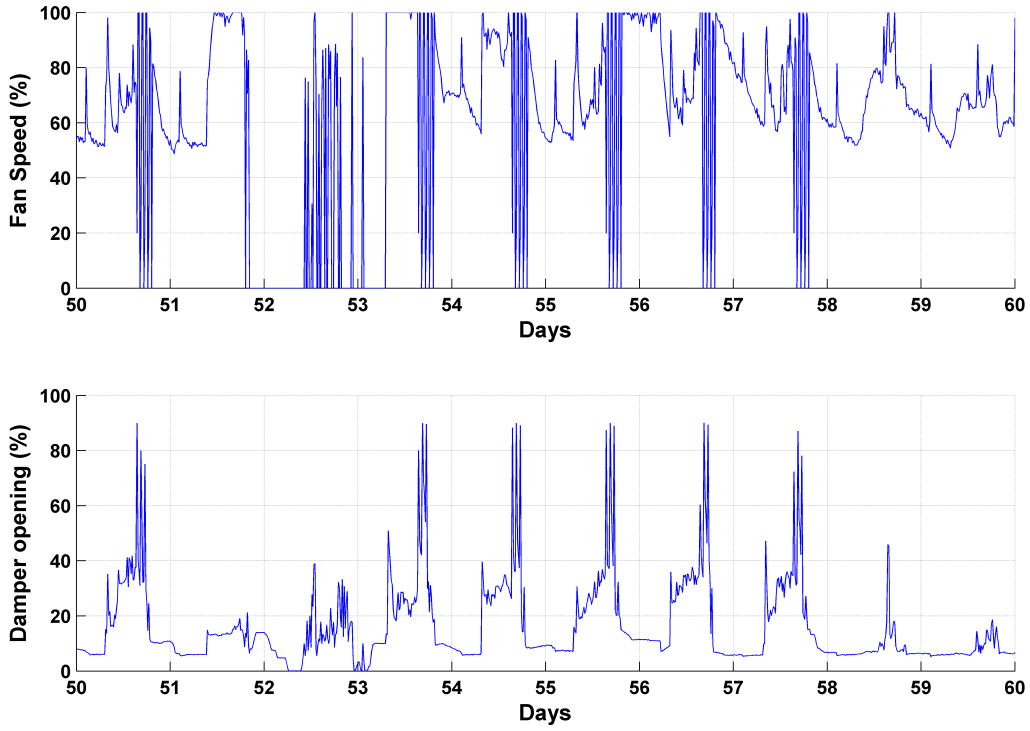


Figure 2.7: Fan Speed and Average Damper Opening Data Sampled Once at 15 Minute Intervals

speed. Figure 2.7 shows a 10 day period of the data of the fan speed and average damper opening. Figure 2.8 shows the results of applying the algorithm on the fan and damper data. As is evident from Figure 2.8, the periods of time for which the hunting is observed is same for both the process variables.

2.3 Survey of Hunting Behavior in Campus Buildings

In the previous section, a simple algorithm for the detection of hunting in building HVAC systems in real time is proposed. The algorithm is then verified by applying it on CHW opening data sampled at different frequencies. A demonstration is also made to show that due to the interconnected nature of HVAC systems, hunting might be present simultaneously in several control loops. In this section the results of a survey that was

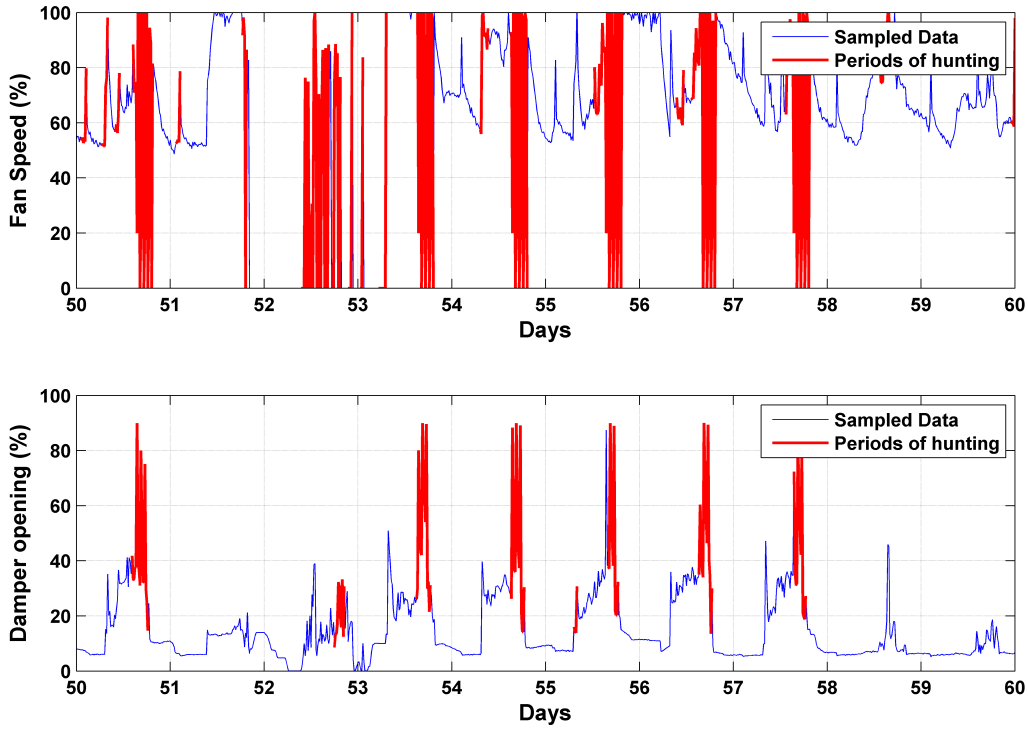


Figure 2.8: Fan Speed and Average Damper Opening Data Showing the Occurrence of Simultaneous Hunting

performed to determine the prevalence of hunting in building HVAC systems are provided.

Ten buildings of the Texas A&M University campus were selected for the purpose of the survey. From among the several components corresponding to the HVAC systems of the buildings being studied, AHUs are selected as they are the primary source for providing conditioned air to all the building zones. Presence of hunting phenomenon in AHUs probably indicates hunting in other components of the HVAC system. A schematic of the AHU serving the buildings is provided in Figure 2.9. AHUs take in a mixture of outside air and return air with the help of a pressure differential that is created by the AHU fan. The mixed air passes through a heat exchanger that uses CHW for conditioning. The amount of CHW used for cooling is regulated by a CHW valve. The hunting algorithm is applied

Table 2.1: Prevalence of Hunting in Fans in Buildings

Building Number	Number of Fans Observed	Number of Fans Exhibiting Hunting	% Duration for which Hunting was Observed
1	3	1	11
2	10	0	–
3	2	0	–
4	1	1	23
5	3	0	–
6	2	0	–
7	2	0	–
8	8	7	6-26
9	6	0	–
10	3	0	–

Table 2.2: Prevalence of Hunting in CHW Valves in Buildings

Building Number	Number of Fans Observed	Number of Fans Exhibiting Hunting	% Duration for which Hunting was Observed
1	3	2	6-19
2	10	10	33-78
3	2	2	6-7
4	1	1	27
5	5	4	7-31
6	6	2	12-39
7	2	0	–
8	8	6	7-12
9	6	5	14-31
10	3	0	1–

to two of the process variables of the AHU, the fan speed, and the CHW valve opening.

Data sampled at 15 minute intervals were collected for two of the process variables corresponding to the AHUs serving the buildings, the AHU fan speed, and the AHU CHW valve. The algorithm is applied on the collected data to determine the number of components that exhibit hunting, and also the percentage duration for which the phenomenon

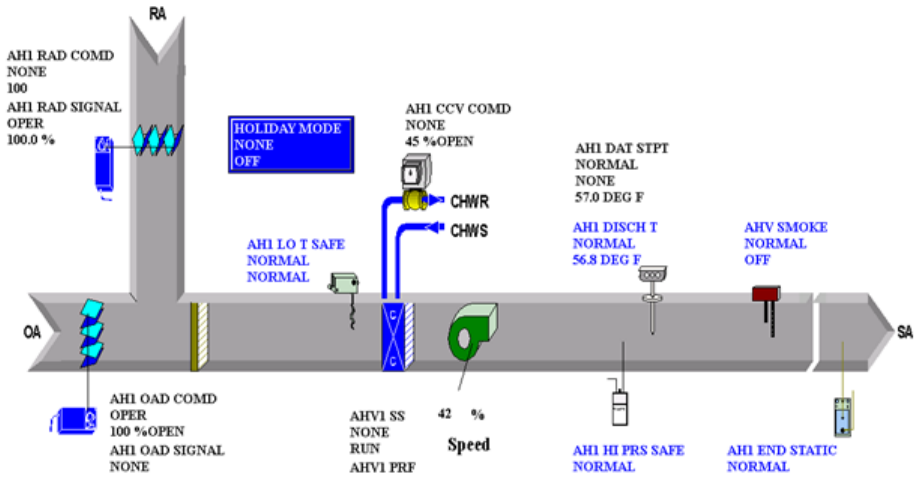


Figure 2.9: Schematic Representation of a Typical AHU

is observed. Tables 2.1 and 2.2 show the results of the survey of AHU fans and CHW valves, respectively. The results of the survey show that 70% of the CHW valves exhibited hunting for 6% to 78% of the time, and 22% of the fans studied exhibited hunting for 6% to 26% of their operation time. The results indicate that hunting is a widespread problem and that measures to minimize the occurrence of hunting have a significant potential to improve HVAC operations and reduce energy.

2.4 Cascaded Control Strategy

The previous section provided the results of a survey that demonstrated the widespread nature of the problem of hunting in building HVAC systems. This section is a summary of the contribution by Christopher Price which was performed as part of a joint work along with the author in [5]. A solution to the problem of hunting is proposed in the form of an alternative control architecture to the traditional feedback control, called as cascaded control. The cascaded control architecture is applied on a simulation model of an AHU to demonstrate its capacity to reduce the phenomenon of hunting.

Cascaded control architectures have been shown to reduce the phenomenon of hunting

in building HVAC systems [47]. In [47] Elliot and Rasmussen demonstrated that the cascaded control architecture reduces hunting in Vapor Compression Cycle (VCC) systems. The first step in VCC is to pass the refrigerant used for cooling through an expansion valve that causes an abrupt change in pressure. The refrigerant which was initially in gaseous phase converts into a cool two-phase liquid. The two-phase liquid is then passed through the evaporator where through the process of forced convection provides the necessary cooling to the liquid to be cooled. The heat transfer process is most efficient when the refrigerant is transformed from the liquid phase to vapor phase. Minimizing the refrigerant superheat (SH) by the expansion valve would result in a more efficient operation, however, the refrigerant may enter the compressor in liquid phase which is undesirable. Two sources of nonlinearity exist while regulating the expansion valve, the relationship between the refrigerant mass flow rate, and the super heat dynamics which vary drastically with operating conditions. The presence of these nonlinearities result in a well documented phenomenon of hunting in VCC systems.

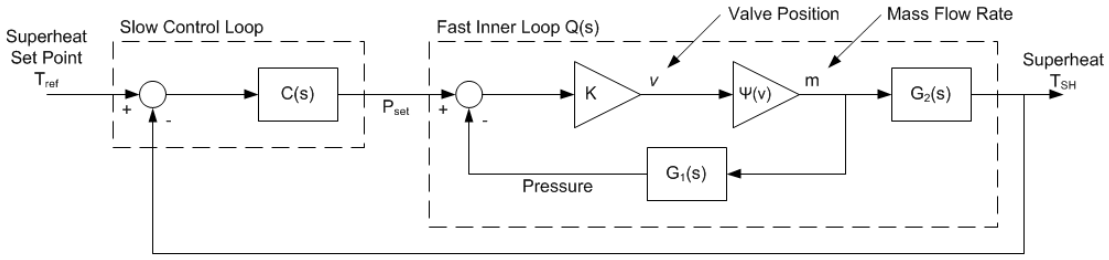


Figure 2.10: Block Diagram Representation of VCC Cascaded Loop

In [47], the authors show that implementing a cascaded control architecture can eliminate the phenomenon of hunting in VCC systems. The cascaded control architecture in this section is explained with the help of the VCC system. The cascaded control architecture corresponding to the VCC system is shown in Figure 2.10. As is shown in the figure,

the cascaded control loop comprises a fast inner loop, and a slow outer loop. The inner loops comprises a proportional gain K, and the outer loop is a PID controller $C(s)$. The transfer function corresponding to the faster inner loop is shown in Equation 2.9

$$Q(s) = \frac{K \cdot \Psi(\nu) \cdot G_2(s)}{1 + K \cdot \Psi(\nu) \cdot G_1(s)} \tag{2.9}$$

where $\Psi(\nu)$ represents the nonlinear relationship between the vavlve position and the mass flow rate of the refrigerant of the VCC system, $G_1(s)$ represents the transfer function between the mass flow rate of the refrigerant and the superheat temperature T_{SH}, and $G_2(s)$ represents the transfer function between the pressure P_{set} and T_{SH}. The equation shows that the nonlinear characteristics of the valve $\Psi(\nu)$ appears both in the numerator and denominator of the inner loop transfer function. If the magnitude of the inner loop proportional gain K is sufficiently large, then the transfer function $Q(s)$ approaches 1 as shown in Equation 2.10.

$$\lim_{K \to \infty} Q(0) = 1 \tag{2.10}$$

Since the transfer function of the inner loop approach a constant steady-state value as the magnitude of the proportional gain K is increased, the dynamics as seen by the outer loop become linear. As the inner loop dynamics become linear the system behaves in a similar fashion under varying operating conditions, thereby, reducing the phenomenon of hunting. A simulation example is provided in the following section where cascaded control is applied on an AHU model to eliminate the occurrence of hunting in CHW valves.

2.4.1 Cascaded Control on an AHU Simulation Model

Before providing the simulation results of applying the cascaded control architecture, a brief description of the AHU model is provided. Warm air passes through a finned heat exchanger. The heat exchanger is supplied with CHW with a mass flow rate of $q(\delta)$ where

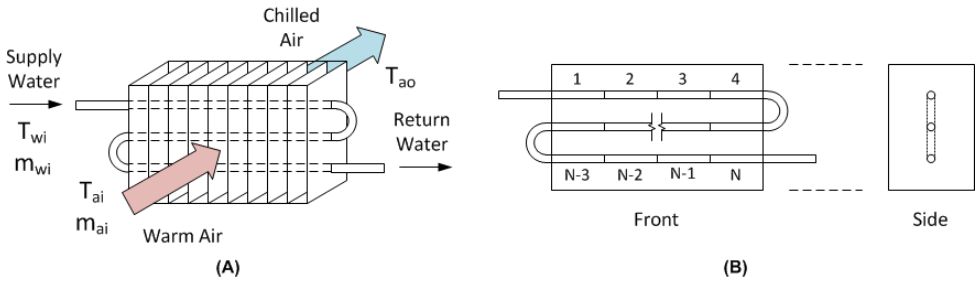


Figure 2.11: A Schematic of the AHU Heat Exchanger Simulation Model

δ is the valve position. CHW cools the warm air through a process of forced convection. The dynamics of the heat exchanger are modeled using a finite volume approximation which uses the model of a single row cooling coil proposed by Zhou in [48]. The tubing is divided into N sections, and the discharge air at the end of each finite volume is calculated with the help of the coil-side and water-side equations shown in 2.11.

$$
\begin{aligned}
&C_w \cdot \frac{\partial T_w}{\partial t} + c_{p,w} \cdot q(\delta) \cdot \frac{\partial T_w}{\partial x} + k_w(T_w - Tc) = 0 \\
&C_c \cdot \frac{\partial T_c}{\partial t} + \kappa(T_c - T_{a,in}) + \kappa(T_c - Tw) = 0 \\
&T_{a,out} = T_{a,in} + \epsilon_a \cdot (T_c - T_{a,in})
\end{aligned}
\tag{2.11}
$$

In the equations above, C_w and C_c are the heat capacities of the water and coil surface, respectively, $c_{p,w}$ is the specific heat capacity of water, and T_w and T_c are the water and coil surface temperatures, respectively. The variables $T_{a,in}$ and $T_{a,out}$ refer to the inlet and outlet temperature of the air at the finite volume. The heat transfer coefficient ϵ_a is computed using an NTU method, and the thermal conductivities κ of the air and water are a function of the inner tube diameter and heat transfer effectiveness. A more detailed description of the AHU cooling coil model is provided in [41]. Figure 2.11 shows a schematic of the AHU heat exchanger model.

The AHU model described above is used to run several simulations to demonstrate

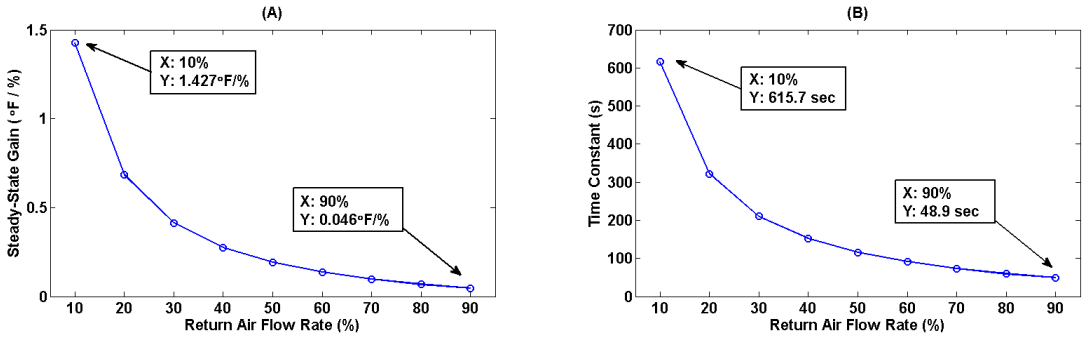


Figure 2.12: Response Characteristics of the AHU Heat Exchanger Under Different Mass Flow Conditions

the nonlinear characteristics of the relationship between chilled water valve position and the discharge air temperature. The simulations vary in the mass flow rate of the air that passes through the AHU heat exchanger. The mass flow rates are varied from 10-90 % of the maximum flow. The inlet temperature of the mass flow rate of the air is set at 72^0C. The change in the discharge air temperature for a negative step change in CHW flow rate is recorded. The same step change in CHW flow is applied to all the simulations. Figure 2.12 shows how the steady state gain of the change in discharge air temperature per unit decrease in CHW flow rate changes with the mass flow rate of the air. As is evident from the figure, the largest steady state gain is 30 times the smallest gain. Figure also shows the time constant of the heat exchanger dynamics associated with each simulation. The largest time constant is about 12 times the shortest. Due to the nonlinear nature of the heat exchanger dynamics, the performance of a PI controller used to regulate the discharge air temperature would heavily depend on the operating condition. The variations in the operating conditions thus results in hunting phenomenon.

A cascaded control loop architecture is then applied to the simulation model to regulate the discharge air temperature. A schematic showing the application of cascaded control on the AHU heat exchanger is shown in Figure 2.13. The simulations are run with the discharge air temperature set at 57^0F. Outside air conditions and mass flow rate of the inlet

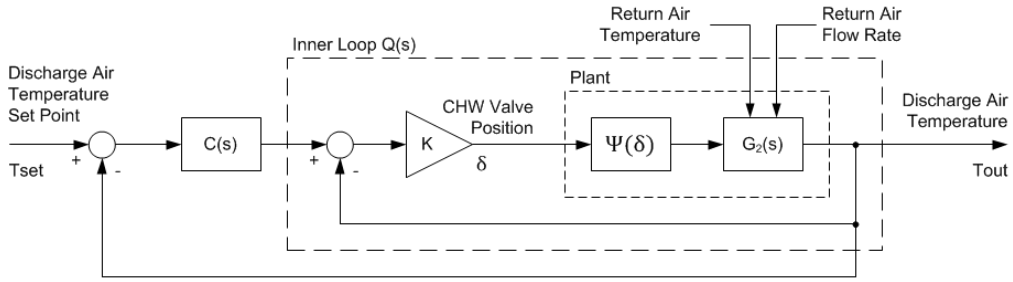


Figure 2.13: Flow Diagram Showing the Cascaded Control Loop to Regulate Dsicharge Air Temperature

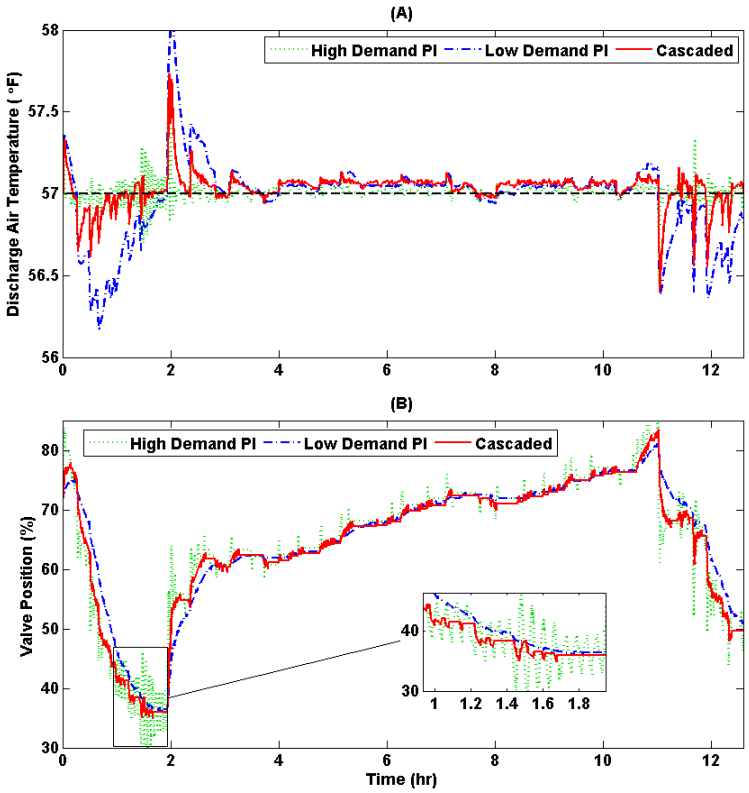


Figure 2.14: Simulation Results Showing the Discharge Air Temperature Using Different Controls

air are varied based on measurements made by the BEMS of an office building between 6 AM and 6:30 PM on 5/21/2014. Simulation results are shown in Figure 2.14. For

Table 2.3: Simulation Results Corresponding to Low Demand PI, High Demand PI, and Cascaded Control

Metric	Low Demand PI	High Demand PI	Cascaded
RMS	0.249^0F	0.062^0F	0.089^0F
MAE	1.23^0F	0.55^0F	0.73^0F
Total Valve Travel	214	999	474
Hunting	No	Yes	No

comparison purposes, simulations are also run with PI controllers which are tuned under high demand and low demand conditions. The figure shows that the simulations which are run with a PI controller tuned for high demand conditions exhibits hunting behavior with a magnitude of 5-15% during times when the chilled water demand is low. The simulations which are run with PI controller tuned under low demand conditions exhibits no hunting, but shows poor control performance. The cascaded controller provides the most consistent performance of the three controllers. Table 2.3 shows the Root Mean Squared Error (RMS), Maximum Absolute Error (MAE), and total travel time associated with each type of controller.

2.5 Conclusion

Hunting is one of the aspects of improper functioning of building HVAC components that may result in suboptimal performance and increased energy costs. Detection is the first step that should be taken in tackling this problem. A simple detection algorithm for the presence of hunting is presented in this paper. The detection is performed by using only the measured value of the process variable being observed and does not require that the set point values be recorded. The algorithm makes use of the fact that the oscillations, which are part of the control design of the HVAC components, have much smaller frequencies than those caused by variations in outside weather conditions.

Using the detection algorithm, a survey was performed on the AHUs of 10 buildings

at Texas A& M University finding that 70% of the CHW valves for 6% to 78% of the time, and 22% of the fans studied exhibited hunting for 6% to 26% of their operation time. It was also observed that due to system interconnections, hunting was observed in more than one subsystem associated with the same AHU during the same time periods. The survey demonstrates that hunting is a significant problem in building HVAC components, and reducing its occurrence would allow for significant improvements in performance and reducing costs.

A cascaded loop architecture is proposed in this chapter to deal with the nonlinearities inherent in the HVAC system dynamics and reduce the occurrence of hunting. Simulation using a finite volume model of AHU forced convection dynamics were able to reproduce the hunting behavior identified in a majority of units found on the campus of Texas A& M University. Large variations in system steady-state gains and time constants as well as nonlinearities due to heat transfer combine to cause oscillatory behavior in chilled water valves. These changing gains make performance of static PI controllers heavily dependent on conditions during controller tuning. The cascaded loop architecture successfully eliminates hunting behavior in simulation while providing consistent performance across all operating conditions. By reducing total actuator travel, component lifespan will be lengthened. Consistent performance means that HVAC technicians will not be required to re-tune control loops seasonally thus substantially reducing service overhead. Overall, the cascaded approach offers a cost savings approach that eliminates undue ware on system actuators while eliminating the need for seasonal tuning of PI control loops. Also because implementation only involves the addition of one standard proportional control loop, the cascaded approach is readily implementable in the HVAC field.

Future research in this area will include the implementation of the cascaded loop on real building systems including some of those surveyed in this paper. Real cost savings and performance associated with this control strategy will be recorded and compared with

traditional PI control. Tuning methods for the cascaded loop will also be investigated in terms of effective procedures for optimal performance. Application of the cascaded approach will also be investigated in other HVAC systems due to similar nonlinear behavior. Control problems such as VAV damper or fan speed control display the same condition dependent behavior and will benefit from cascaded loop control.

3. BLACK-BOX LINEAR SYSTEM IDENTIFICATION OF BUILDING HVAC SYSTEMS

In Chapter 1 of the dissertation, three reasons for inefficient operation of traditional control practices in building HVAC systems were introduced. The first reason is the improper tuning of the gains of the PI-type controllers. One of the consequences of improper tuning is the phenomenon of hunting. The preceding chapter provided the results of a survey that showed the widespread nature of the hunting phenomenon in buildings, and also offered a solution in the form of an alternate control architecture, namely cascaded control. A simulation example was provided that demonstrated how the cascaded control architecture has the potential to reduce the amount of hunting.

The second reason for inefficient operation is that there is no optimization performed at the global level. This is due to the decentralized nature of the control architecture employed. In decentralized control each component is actuated to a setpoint value which is determined without consideration of its effects on the neighboring subsystems. For example, consider the chilled water (CHW) discharge temperature setpoint assigned to the chiller system. Whilst a higher setpoint temperature leads to a more efficient operation of the chiller, it also results in increased pump energy use to satisfy the cooling demands of the building. The chiller systems are commonly designed to provide full cooling load at around 42^0F [49]. Building operators generally fix the CHW setpoint temperature at 42^0F to reduce the pump energy consumption and meet the building cooling load requirements. This leads to inefficient operation as the cooling load requirements are below the maximum for most of the time. Varying the CHW setpoint temperature based on the cooling load requirements would allow for a more efficient operation.

The third reason for inefficient operation is the lack of planning for future operating

conditions. For instance, say the occupancy of a room in the building starts at 8:00 am. The PI-type control methodologies employed in most buildings begin cooling the room from 8:00 am. Due to the thermal inertia of the walls,floor etc., however, the temperature of the room does not reach the desired temperature until some time after. This results in an initial period of time where the occupant thermal comfort needs are not met. If the occupancy schedule of the building is known in advance, strategies such as precooling the building can reduce these periods of discomfort.

The second and third reasons for inefficient operation described in the preceding paragraphs can be addressed by using more advanced control algorithms such as MPC. One of the biggest challenges of implementing MPC, however, is the development of models that can accurately predict the behavior of building HVAC systems with computational requirements that would allow for real-time control. This chapter seeks to address this problem by proposing an algorithm that automates the process of model development making the implementation of MPC more feasible and scalable.

The remainder of the chapter is organized as follows. First, a discussion of the various modeling approaches for building HVAC systems and the problems associated with each approach is presented in Section 1. The discussion of the approaches is followed in Section 2 by a mathematical description of the modeling method that has been adopted. Section 3 then presents the proposed algorithm for automating the process of model development, followed by a verification of the algorithm by using data obtained from a real working office building. Since the primary purpose of model development in this dissertation is the application of model-based control, the proposed algorithm is verified by applying MPC on a high fidelity simulation model of the office building in Section 4. The chapter ends with a summary and analysis of the results obtained from the simulations in Section 5.

3.1 Introduction

The modeling methods available in literature can be broadly be classified as white-box, gray-box, and black-box methods. The modeling methods differ in the amount of information required to develop them, and the computational burden required for their simulation and prediction of outputs. Figure 3.1 shows the different modeling approaches available in the current literature.

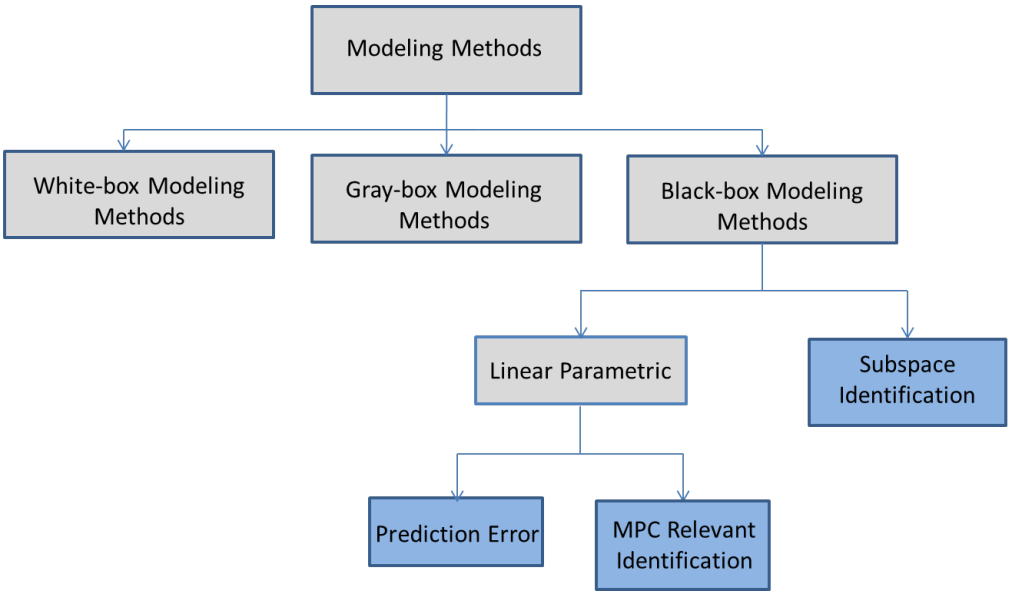


Figure 3.1: Modeling Approaches for Building HVAC Systems in Literature

3.1.1 White-Box Modeling Method

White-box models are developed from a detailed understanding of the underlying physics. They provide the most accurate descriptions of the thermal dynamics of the buildings. There are two reasons, however, as to why white-box models are not suitable for application of MPC on real building systems. Firstly, the accuracy of the models is contingent on the accuracy of the information required to develop them. In addition,

the information required, is not easily available. For example, the EnergyPlus white box models require information such as building material, building area and dimensions, placements of doors and windows, insulation levels, etc., which are not easy to obtain.

Secondly, the simulation of white-box models is computationally intensive. This makes the implementation of MPC infeasible for the following reason. Consider a solution space which includes the set of all possible values of the control setpoints that need to be optimized. For example if there are n control setpoints represented by the vector $\boldsymbol{U}^{set} = [u_1^{set}, u_2^{set}, ..., u_n^{set}]$, then the solution space is n-dimensional. Each point in the solution space represents a control setpoint vector with unique set of values. The objective of MPC is to determine the optimal point $\boldsymbol{U}^{mpc} \in \boldsymbol{U}^{set}$ that minimizes a certain objective function $\boldsymbol{J}^{mpc}$. White-box models do not provide any simple analytical relationship between the objective function $\boldsymbol{J}^{mpc}$ and the control setpoint vector $\boldsymbol{U}^{set}$. Hence the standard mathematical optimization tools cannot be used to compute $\boldsymbol{U}^{mpc}$. Optimization using white-box models can be performed, however, by running simulations corresponding to each point in the solution space, and selecting the control input set that results in the least value of $\boldsymbol{J}^{mpc}$. As the number of control inputs to be optimized increases, however, such a method quickly becomes infeasible since the number of points to be tested grows exponentially large with the increase in the number of inputs to be optimized.

3.1.2 Gray-Box Modeling Method

Gray-box models are a hybrid between white-box models which are built from first principles of physics, and an empirical approach where the models are developed purely from data obtained through Building Energy Management Systems (BEMS). In the gray-box approach, building thermodynamics are represented by an equivalent Resistance-Capacitance (RC) circuit. For example, the building walls and floors, and the room air are treated as thermal capacitors and form the nodes of the circuit. The flow of heat from

any two nodes, for instance between the room air and the room walls, or the room walls and the outside air, is assumed to pass through a thermal resistor. The modeling approach then entails the determination of the values of the resistances and capacitance through an empirical process involving nonlinear regression. An example of an equivalent RC circuit of a two-zone building is reproduced in Figure 3.2. In the figure C_{11} and C_{22}, and C_{1a} and C_{2a} are the thermal capacitance of the walls and air, respectively, of the two thermal zones. The symbols $R_{11}, R_{22}, R_{12}, R_{1a}$, and R_{2a} are the thermal resistances.

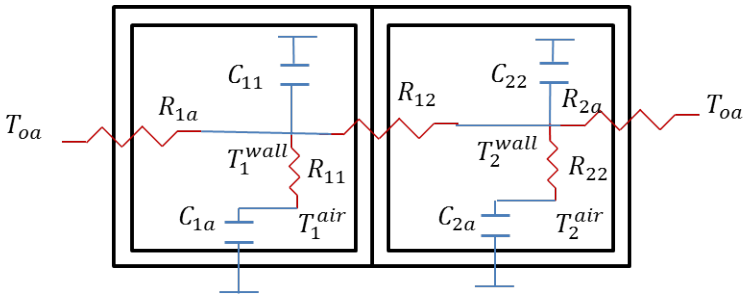


Figure 3.2: Schematic Showing an Equivalent RC Circuit for a Two Zone Building

There are again two reasons why the gray-box modeling method is not suitable for the purpose of automating the process of model development of building HVAC systems. Firstly, building the equivalent RC circuit requires a complete knowledge of the internal structure of the rooms. The placement of the thermal resistors requires a knowledge of how the rooms are located in relation to each other. The second reason for not using the gray-box approach is that there are no standard regression methods available for the determination for the values of the thermal resistors and capacitors.

3.1.3 Black-Box Modeling Method

Unlike white-box and gray-box models, black-box models are developed purely from data. Outputs and inputs of the system to be modeled are first defined, followed by

the identification procedure which involves determining a statistical relationship between them. The growing use of BEMS that measure various aspects of the building HVAC system such as room temperatures, flow rates, AHU discharge air temperature and pressure, etc. makes the development of black-box models suitable for automating model development. Due to the ease of development and the ability to automate and scale the process, the black-box modeling approach is selected in this dissertation for the purpose of implementing MPC.

Although black-box models are easy to develop, there are two inherent drawbacks associated with the approach. Firstly, the data used to develop the models need to be of a certain quality, i.e. the inputs and outputs need to be sufficiently perturbed for the identification procedure to capture the dynamics of the system being modeled. Under the normal course of building operation, the inputs might not be sufficiently perturbed for the identification of a reliable model. Intentionally perturbing the inputs under experimental conditions may not always be feasible.

Secondly, as the building HVAC system employs hundreds of components there exists a high degree of collinearity between various subsystems. Since the black-box approach is purely statistical, the identification process may result in a model that shows a causal relationship between two subsystems with high collinearity but no interconnected dynamics. For example, consider two rooms in a multi-zone building which are not physically adjacent to each other. The two rooms may have high degree of collienarity as they may share similar occupancy schedules, setpoint temperatures, etc. The black-box approach may lead to a model that shows interconnected dynamics between the two rooms which would be highly unlikely given that they are not neighbors.

In addition to the two problems mentioned above, there are two additional areas in which the current literature on black-box modeling is lacking. Firstly, none of the papers include both buildings and AHU with all of the details [28]. Most of the papers model only

the room temperatures. Optimizing the building HVAC operation requires models of other components such as the AHU fan, heat exchanger etc. Secondly, the model is developed with little or no underlying physics. The predictions made by the model correspond to a specific set of operating conditions under which the model has been trained. There is no guarantee that the predictions made by the model would be accurate under a different set of operating conditions. This generates a question that remains largely unanswered in literature. Does a black-box model with good predictive properties translate into a good model for control?

In this research an automated modeling algorithm is proposed that seeks to address the aforementioned problems in the black-box approach and the shortcomings in the literature. The modeling algorithm is first verified by showing good prediction properties on data obtained from a real working office building comprising 11 rooms. The algorithm is then verified for its usefulness in applying model-based control by applying MPC on high-fidelity simulation models of the office building. A comparison of the predictive properties and the MPC results of three different black-box approaches is also presented.

3.2 Black-Box Modeling Theory

The output being modeled by the black-box approach is generally continuous in nature like room temperature, humidity etc. The sensors employed by BEMS, however, take measurements at fixed intervals of time giving the output a discrete nature. The discrete output is modeled as a function of the previously measured output and input values. For example, sensors at an office building at the Texas A&M University campus in [50] measure room temperatures, volume flow rates, discharge air temperature, and weather conditions. By using the black-box modeling approach the room temperature at time instant t is expressed as a function its past values, and the other aforementioned input values measured before t.

Not all the factors that influence the output being modeled are generally captured by

BEMS sensor data. For instance in the above example where the room temperature is being modeled, factors such as solar incidence, cloud cover, wind-driven infiltration, etc. affect the output but are not measured by the sensors. In the black-box modeling approach the unmeasured inputs are assumed to be stochastic in nature, generally Gaussian with zero mean and variance 1. The modeling approach uses statistical tools to extract the influence of these stochastic inputs from historical data. Thus the output modeled by the linear parametric black-box approach is written as a linear function of deterministic and stochastic inputs.

Based on the structure of the function used to describe the relationship between the inputs and the outputs, the linear black-box modeling methods can be classified into two categories, linear parametric and subspace identification as shown in Figure 3.1. A description of the underlying theory behind each of the two approaches is provided as follows.

3.2.1 Linear Parameteric Approach

As stated in the preceding section, the discrete-time output can be expressed as a sum of deterministic and stochastic components as shown in Equation 3.1

$$y(t) = G(q)u(t) + H(q)e(t) \tag{3.1}$$

where u is the deterministic input, e is the stochastic input, and $G(q)$ and $H(q)$ correspond to the deterministic and stochastic transfer functions, respectively. The symbol q is a time shift operator. For example, $q^{-2}y(t)$ refers to $y(t-2)$. The objective of the modeling method is to determine the transfer functions $G(q)$ and $H(q)$.

In the linear parametric approach, the transfer functions G(q) and H(q) are expressed

as rational functions of the time shift operator q as shown in Equation 3.2.

$$G(q) = \frac{B(q)}{F(q) \cdot A(q)} \; ; \; H(q) = \frac{C(q)}{D(q) \cdot A(q)} \tag{3.2}$$

The numerator and denominator functions $A(q), B(q), C(q), D(q)$, and $F(q)$ are polynomials in the time shift operator q as shown in Equation 3.3

$$\begin{aligned} A(q) &= 1 + a_1 q^{-1} + a_2 q^{-2} + ...a_{n_a} q^{-n_a} \\ B(q) &= 1 + b_1 q^{-1} + b_2 q^{-2} + ...b_{n_b} q^{-n_b} \\ C(q) &= 1 + c_1 q^{-1} + c_2 q^{-2} + ...c_{n_c} q^{-n_c} \\ D(q) &= 1 + d_1 q^{-1} + d_2 q^{-2} + ...d_{n_d} q^{-n_d} \\ F(q) &= 1 + f_1 q^{-1} + f_2 q^{-2} + ...f_{n_f} q^{-n_f} \end{aligned} \tag{3.3}$$

where n_a, n_b, n_c, n_d, and n_f are parameters that determine the model order. Model order is the determination of how many past values of the input or output that need to be considered. For example, consider n_b which is the model order corresponding to input u. If $n_b = 2$, then the output at time instant t is influenced by two past input values, i.e, $u(t-1)$ and $u(t-2)$.

The discrete time output expressed in Equation 3.2 can be expressed using a subset of the functions $A(q), B(q), C(q), D(q)$, and $F(q)$. Not all the functions need to be present at the same time. Depending on which functions are selected to represent the output, the parametric equation in 3.2 can be classified into different structures, with ARX, ARMAX, Box-Jenkins (BJ), and Output Error (OE) being the most widely used. The model structures differ in how the deterministic and stochastic inputs affect the dynamics of the output.

An example of the ARX model structure is shown in Equation 3.4.

$$y(t) = \frac{B(q)}{A(q}u(t) + \frac{1}{A(q)}e(t) \tag{3.4}$$

In the ARX model structure, the deterministic and stochastic inputs are assumed to have the same system poles. The linear functions used in building the ARMAX, BJ, and OE model structures are shown in Table 3.1. The linear parametric approach for system iden-

Table 3.1: Polynomial Values for Different Model Structures

Model Structure	Functions Used
ARX	$A(q), B(q)$
ARMAX	$A(q), B(q), C(q)$
BJ	$B(q), C(q), D(q), F(q)$
OE	$B(q), F(q)$

tification thus entails the process of first selecting the model order and structure, followed by the parameters corresponding to it. For example, if the ARX model structure is selected with model orders n_a and n_b, then regression analysis is used to identify ^{arx} where ^{arx} is the parameter set shown in Equation 3.5.

$$\theta_{arx} = [a_1, a_2, ...a_{n_a}, b_1, b_2, ..., b_{n_b}]^T \tag{3.5}$$

The algorithm used to select the model order and structure is provided in Section 3. The model parameters are computed by performing regression analysis between measured outputs and the outputs predicted by the model. Before providing a description of the regression analysis, the analytical relationship between the output predictions of the model and the model parameters is presented below.

3.2.1.1 One-Step-Ahead Prediction

A description of the symbols and notations used in the computation of the predicted output is first provided. Let the current time instant be t, then a k-step-ahead predicted output is denoted by $\hat{y}(t|t+k)$. The symbol$\hat{}$is used to indicate a predicted value, and the notation $t|t+k$ is used to indicate a k-step-ahead prediction made at time instant t.

If an assumption is made that $y(s)$ and $u(s)$ are known for $s \leq t$, then the one step ahead predicted value of the output $\hat{y}(t|t+1)$ can be expressed in terms of the trasnfer functions $G(q)$ and $H(q)$ in Equation 3.2 as follows.

$$\hat{y}(t|t-1) = H^{-1}(q)G(q)u(t) + [1 - H^{-1}(q)]y(t) \tag{3.6}$$

The derivation of the relationship shown above is provided in [51]. The one-step-ahead predicted output corresponding to each of the four model structures studied in this paper, can be obtained by expressing $G(q)$ and $H(q)$ in terms of the corresponding linear equations $A(q), B(q), C(q), D(q)$, and $F(q)$ substituting it in Equation 3.6. An example of the one-step-ahead prediction corresponding to the ARX model is shown in Equation 3.7.

$$\hat{y}(t|t-1) = B(q)u(t) + [1 - A(q)]y(t) \tag{3.7}$$

3.2.1.2 *k*-Step-Ahead Prediction

The preceding section provided a description of the one-step-ahead predicted value of the output. For the application of MPC, the output predictions beyond just one time step are required. An analytical relationship between a k-step-ahead predicted output and the model structure functions $A(q), B(q), C(q), D(q)$, and $F(q)$ is difficult to derive. Hence, in this chapter, the a k-step-ahead predicted output is obtained through an iterative process.

The iteration equations corresponding to the ARMAX model structure are as shown below.

$$\begin{aligned}
\hat{y}_{t+1|t} =& B(q)u(t) + [1 - A(q)]y(t) + [C(q) - 1]\epsilon(t) \\
\hat{y}_{t+2|t} =& B(q)u(t+1) + [1 - A(q)]\hat{y}_{t+1|t} + [C(q) - 1]\hat{\epsilon}_{t+1|t} \\
& \vdots \\
\hat{y}_{t+n_p|t} =& B(q)u(t+n_p-1) + [1 - A(q)]\hat{y}_{t+n_p-1|t} + \\
& [C(q) - 1]\hat{\epsilon}_{t+n_p-1|t}
\end{aligned} \tag{3.8}$$

In the iterative process shown above, the estimated value of the stochastic input $\hat{e}$ is computed as shown in Equation 3.9.

$$\hat{e}(t|t+k) = \begin{cases} 0 & k \geq 0 \\ y(t+k) - \hat{y}_{t+k-1|t+k} & k < 0 \end{cases} \tag{3.9}$$

Since the stochastic input is assumed to have zero mean, in Equation 3.9 the estimated value of e for all future time steps is assumed to be 0. The estimated value of the stochastic inputs for the past time steps is the error between the measured and one-step-ahead predicted value.

With the help of the relationships shown in Equation 3.6, and 3.8 regression analysis can be performed to determine the model parameters (the coefficients of the functions $A(q), B(q), C(q), D(q)$, and $F(q)$). Depending on the objective function minimized in the regression analysis, the linear parametric approach is further classified into Prediction Error Method (PEM) and MPC Relevant Identification Method (MRI). A mathematical description of the two methods is provided below.

3.2.1.3 Prediction Error Method

In PEM, regression analysis is performed between the measured output and the one-step-ahead predicted value of the output. The parameters of the model are computed by minimizing the quadratic norm of the error between the measured and predicted values. For example the parameters corresponding to the ARX model structure θ_{arx}^{pem} shown in Equation 3.5 are obtained as shown in Equation 3.10.

$$\theta_{arx}^{pem} = \underset{\theta_{arx}}{argmin}\, V^{pem}(\theta_{arx}, Z^{N^{est}}) \tag{3.10}$$

The superscript pem is used to refer to the fact that the parameters were obtained using PEM. In the equation above, N^{est} is the number of samples of the output in the training data set used to estimate the parameters, $Z^{N^{est}}$ contains the input-output data corresponding to the training data set. V^{pem} is the quadratic norm associated with PEM and is a function of the error between the measured and one-step-ahead predicted values of the output as shown in Equation 3.11.

$$V^{pem} = \frac{1}{N^{est}} \sum_{i=1}^{N^{est}} \{y(i) - \hat{y}(i-1|i)\}^2 \tag{3.11}$$

3.2.1.4 MPC Relevant Identification

In PEM, the model parameters are identified by minimizing the norm of the error between the measured and one-step-ahead predicted output. MPC uses output predictions of not just one time step ahead, but over a future time period called as prediction horizon. For example if the prediction horizon spans n_p time steps, and say the current time interval is t, MPC uses the predictions $\hat{y}(t|t+1), \hat{y}(t|t+2), ..., \hat{y}(t|t+n_p)$. Hence in the MRI method, the parameter values are determined by taking into consideration the prediction

errors over the entire horizon. The norm corresponding to the MRI method is shown in Equation 3.12. .

$$V^{mri} = \frac{1}{n_p \cdot (N^{est} - n_p)} \sum_{i=1}^{N^{est}} \sum_{i=1}^{N^{est}} \{y(i) - \hat{y}(i-1|i)\}^2 \tag{3.12}$$

The model parameters corresponding to the ARMAX model structure obtained by minimizing the norm above are shown in Equation 3.13.

$$\theta_{armax}^{mri} = \underset{\theta_{armax}}{argmin}\ V^{mri}(\theta_{armax}, Z^{N^{est}}) \tag{3.13}$$

The MRI norm is nonlinear and non-convex, hence a global optimal solution is not guaranteed by the optimization algorithm used to determine the model parameters.

3.2.2 Subspace Identification

The objective of the subspace identification algorithm is to find a state-space representation of the system. Whereas, conventional algorithm such as PEM, and MRI, try to find a relationship between the inputs and outputs, subspace algorithms places emphasis on system states [52]. A summary of the various subspace identification algorithms is provided in [53]. In this thesis the *N4SID* algorithm in MATLAB was used to determine the state space matrices.

3.3 Automated Black-Box Modeling Algorithm

The previous section provided a mathematical description of the various black-box modeling approaches available in literature. In this section, an algorithm is proposed that uses these approaches to automate the process of model development for building HVAC systems. The section also presents the results obtained by implementing the algorithm on data obtained from a real working office building.

The first step in the development of a model of the dynamics of a system is to select its inputs and outputs. Since the primary purpose of the models is for their application in model-based control, in this research the outputs are selected based on the quantities that are of interest to MPC. For example, while modeling the thermal dynamics of the room, the room temperature is selected as an output since it is used by MPC to measure thermal comfort. The input selection, however, is not as trivial. There may be several factors that may affect the dynamics of the output and identifying these factors is not straightforward. The proposed modeling algorithm thus seeks to determine the inputs which would help capture most of the dynamics of the output. In addition, the determination of these inputs is performed in a way that would allow the process to be easily automated.

3.3.1 Initializing the Set of Selected Inputs

A set of all possible inputs u_i^{pos} that affect the dynamics of the output is first constructed. The subscript i refers to the subsystem being modeled. There are two criteria that are applied while constructing u_i^{pos}. Firstly, including the input must make physical sense, and secondly, the input is measured by one of the sensors of BEMS.

Another set of inputs is constructed u_i^{sel} which contains the list of inputs that have been selected by the modeling algorithm. The set of selected inputs is a subset of u_i^{pos}, i.e., $u_i^{sel} \subset u_i^{pos}$. The set of selected inputs is populated in an iterative manner and is first initialized to contain only the outside air temperature T_{oa} and the control input u_i^c corresponding to the subsystem, i.e. $u_i^{sel(0)} = [T_{oa}, u_i^c]$. The outside air temperature is selected as a default input since it affects the dynamics of all the building HVAC components. The number in parenthesis as a superscript to u_i^{sel} indicates the iteration number.

3.3.2 Selecting the Best Model Order and Structure

After initializing the set of selected inputs, the identification process entails the selection of a model order and structure that best represents the dynamics of the system. In this

dissertation, the selection is made using an exhaustive process. Several model order and structure combinations are generated. The model structures are restricted to those listed in Table 3.1, and the model orders are restricted to a maximum value of 2. For example, in the BJ model structure, the model order is determined by 4 parameters n_b, n_c, n_d, and n_f. Each of these parameters can take a value either 1 or 2, thus giving a total of 16 possible BJ models. Similarly, the ARX, ARMAX, and OE model structures can each have 4, 8, and 4 models up to order 2, respectively. Thus a total of 32 model structure and order combinations are investigated. Model orders only up to 2 were investigated since in [26] and [54] the authors demonstrated that while modeling the thermal dynamics of the building, data older than 2 time steps did not have a significant impact on the output.

For each combination of model order and structure, the model parameters are identified using either the PEM or MRI methods, by minimizing the norm in Equation 3.11 or 3.12, respectively. The training data set $Z^{N^{est}(j)}$ used in the computations of the aforementioned norms contains the output values $\boldsymbol{Y}_i^{est} = [y_i(1), y_i(2), ..., y_i(N^{est})]$, and the input values $\boldsymbol{U}_i^{sel(j),est} = [u_i^{sel(j)}(1), u_i^{sel(j)}(2), , u_i^{sel(j)}(N^{est})]'$, where j is the current iteration number. The set of 32 models under investigation are denoted by the vector $\boldsymbol{M^{(j)}}$.

The models $\boldsymbol{M^{(j)}}$ are then tested based on their accuracy of output predictions. If the models are intended for use by MPC which optimizes control input values over a prediction horizon spanning n_p time-steps ahead, then n_p-step-ahead output predictions are computed corresponding to each model in $\boldsymbol{M}$. The predictions are computed using an iterative process such as in Equation 3.8. The data set used to make the output predictions represented by the vector $Z^{N^{val}}$ which is different from the traingin data comprises the output values $\boldsymbol{Y}_i^{val} = [y_i(1+N^{est}), y_i(2+N^est), ..., y_i(N^{val}+N^{est})]$, and the input values $\boldsymbol{U}_i^{sel(j),val} = [u_i^{sel(j)}(1 + N^{est}), u_i^{sel(j)}(2 + N^{est}), , u_i^{sel(j)}(N^{val} + N^{est})]'$. The predicted values from each model are then compared to the corresponding measured output values. The accuracy of prediction is measured using a fit percentage computed with the help of

Equation The model that provides the best fit percentage f_{fp}, is selected.

$$f_{fp} = \left(1 - \sqrt{\frac{\sum_{i=N^{est}+1}^{N^{val}-n_p} \{y(i+n_p) - \hat{y}(i|i+n_p)\}^2}{\sum_{i=N^{est}+1}^{N^{val}-n_p} \{y(i+n_p) - \frac{1}{N^{val}-n_p}\sum_{i=n_p}^{N} y(i)\}^2}}\right) \tag{3.14}$$

The model with the highest fit percentage is selected as a candidate model as shown in Equation 3.15.

$$M^{*(j)} = \underset{M^{(j)} \in \boldsymbol{M}^{(j)}}{argmax}\ f_{f_p}(M^{(j)}) \tag{3.15}$$

The fit percentage corresponding to the chosen model $M^{*(j)}$ is represented by $f_{fp}^{*(j)}$. The fit percentage of the candidate model of the current iteration is compared to the fit percentage of the candidate model from the previous iteration. The candidate model of the current iteration is then kept or discarded based on the comparison of the fit percentages as shown in Equation 3.16.

$$M^{*(j)} = \begin{cases} M^{*(j)} & f_{fp}^{*(j)} \geq f_{fp}^{*(j-1)} \\ M^{*(j-1)} & f_{fp}^{*(j)} < f_{fp}^{*(j-1)} \end{cases} \tag{3.16}$$

If the candidate model at the current iteration is the same as that of the previous iteration, then the iterations are terminated and $M^{*(j)}$ is chosen as the to represent the dynamics of the system being modeled. The selected input set corresponds to that of the previous iteration, i.e. $u_i^{sel} = u_i^{sel(j-1)}$.

3.3.3 Augmenting the Set of Selected Inputs

The proposed algorithm uses an additive approach in selecting the set of significant inputs, i.e., the algorithms starts with an initial set of inputs $u_i^{sel(0)}$, and is augmented by a single input every iteration. The process for selecting the candidate input to be augmented in u_i^{sel} is described below.

Consider the set of selected inputs in the j^{th} iteration $u_i^{sel(j)}$, and the set of possible

inputs u_i^{pos}. A third set of inputs referred to as the unselected inputs $u_i^{unsel(j)}$ is constructed from the aforementioned sets by including all the inputs in u_i^{pos} that are not present in $u_i^{sel(j)}$ as shown in Equation 3.17.

$$u_i^{unsel(j)} = u_i^{pos} \cap (u_i^{sel(j)})^{\complement} \tag{3.17}$$

In the $j+1^{th}$ iteration, $u_i^{sel(j)}$ is augmented by one of the inputs from $u_i^{unsel(j)}$. The input that is augmented is denoted by $u_i^{aug(j)}$. The augmentation only happens, however, if the inclusion of $u_i^{aug(j)}$ increases the accuracy of output prediction. Before providing a description of how the accuracy of output prediction is measured, the method for determining u_i^{aug} is first presented as follows.

The training data set $Z^{N_{est}(j)}$ and the model $M^{*(j)}$ are used to compute one-step-ahead predicted values of the output, $\hat{\boldsymbol{Y}}_i^{est}(t|t+1)$ using Equation 3.6. The predicted values are used in the computation of the residual vector (error between measured and predicted values) $\hat{\boldsymbol{e}}^{est}$ as follows.

$$\hat{\boldsymbol{e}}^{est}(t|t+1) = \boldsymbol{Y}_i^{est} - \hat{\boldsymbol{Y}}_i^{est}(t|t+1) \tag{3.18}$$

If a causal relationship exists between a particular input not included in the set u_i^{sel} and the output, then the predicted output $\hat{\boldsymbol{Y}}_i^{est}(t|t+1)$ does not capture the dynamics associated with that input. The uncaptured dynamics thus forms a part of the residuals $\hat{\boldsymbol{e}}^{est}$. There is expected to be a strong cross correlation between the input not included and the residuals. Thus the algorithm uses cross correlation values between the input vector $\boldsymbol{U}_i^{unsel(j),est} = [u_i^{unsel(j)}(1), u_i^{unsel(j)}(2), , u_i^{unsel(j)}(N^{est})]'$ and the residuals $\hat{\boldsymbol{e}}^{est}$.

Instead of considering the absolute values of these vectors,however, difference vectors corresponding to $\boldsymbol{U}_i^{unsel(j)}$ and $\hat{\boldsymbol{e}}^{est}$ are constructed. A difference vector is created by

taking the difference between successive elements of the vector and is denoted by placing the symbol δ before the vector. For example, the difference vectors of $\boldsymbol{U}_i^{unsel(j)}$ and $\hat{\boldsymbol{e}}^{est}$ are denoted by $\delta\boldsymbol{U}_i^{unsel(j)}$ and $\delta\hat{\boldsymbol{e}}^{est}$. An example for calculating the k^{th} element of these vectors is shown below.

$$\begin{aligned}\delta\hat{e}(k|k+1) =& \hat{e}(k|k+1) - \hat{e}(k-1|k)\\ \delta u_i^{unsel(j)}(k) =& u_i^{unsel(j)}(k) - u_i^{unsel(j)}(k-1)\end{aligned} \tag{3.19}$$

The reason for considering difference vectors is to reduce the possibility of including an input which has high collinearity with the output but has no causal relationship. The difference vector is associated with changes in values over successive time steps. It is unlikely for the output to consistently change in the same direction as the input under consideration if there exists no causal relationship. The input that has the maximum crosscorrelation value with the residuals represented by $u_i^{aug(j)}$ is augmented to u_i^{sel} as shown in Equation 3.20.

$$u_i^{aug} = \underset{u_i^{pos}-u_i^{sel}}{argmax}\ r(\delta\boldsymbol{e}^{est}, \delta U_i^{sel(j)}) \tag{3.20}$$

where the function $r(x, y)$ represents the cross correlation between vectors x and y. The augmented input set shown in Equation 3.21 is used in determining the candidate model for the next iteration.

$$u_i^{sel(j+1)} = [u_i^{sel(j)}, u_i^{aug}] \tag{3.21}$$

An example of how cross correlation of difference vectors can be used in selecting the inputs is shown with the help of the following example. The temperature of room 1 of an office building (described in the next section) is being modeled. The set of selected inputs is initialized to include only the outside air temperature, i.e. $u^{sel(1)_1} = [T_{oa}]$. The training data set obtained from the BEMS of the office building corresponds to a duration of 14

days. The data set is broken down into 14 subsets denoted by $n_{slot} = 14$ and cross correlation values are found between difference vectors of the residuals $\delta\hat{\boldsymbol{e}}^{est(1)}$ and the control input $\delta\boldsymbol{U}_1^c$ corresponding to each data subset. For comparison purposes cross correlation values are also shown between the residuals and the difference vector of the temperature of room 6 $\delta\boldsymbol{T}_6$. Room 6 is not adjacent to room 1, hence, the cross correlation values are expected to be small. Figure 3.3 shows the cross correlation values corresponding to the aforementioned sets of data. As expected, the cross correlation values between $\delta\hat{\boldsymbol{e}}^{est(1)}$ and $\delta\boldsymbol{U}_1^c$ is much greater than the cross correlation values between the $\delta\hat{\boldsymbol{e}}^{est(1)}$ and $\delta\boldsymbol{T}_6$.

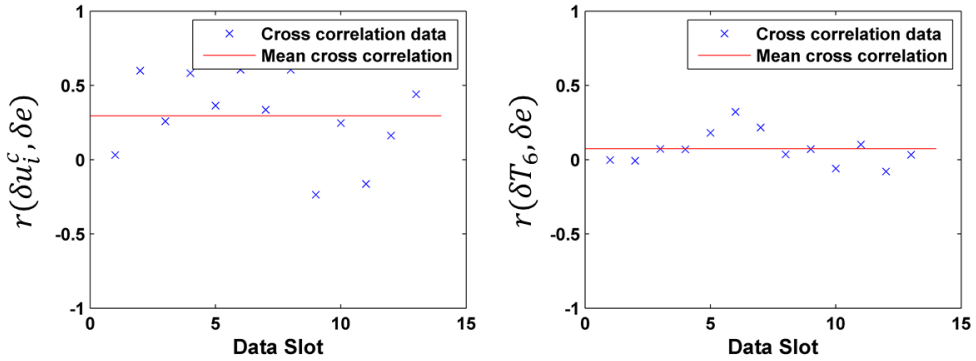


Figure 3.3: Cross Correlation Values Between Difference Vectors of Residuals and Inputs

3.3.4 Summary of the Automated Linear Parametric Black-Box Modeling Algorithm

The steps taken to develop the automated linear parametric black-box model of subsystem i is summarized as follows.

- *Step 0:* Construct the set of possible inputs u_i^{pos} based on the available BEMS data, and a physical understanding of the dynamics of the subsystem. Initialize the iteration number $j = 1$, and the set of selected inputs $u_i^{sel(0)} = [T_{oa}, u_i^c]$.

- *Step 1:* Construct a set $\boldsymbol{M}^{(j)}$ which contains 32 models with unique model structure and model order combinations. Model structures belong to either the ARX, ARMAX, BJ, or OE, and the model order parameters n_a, n_b, n_c, n_d, n_f can take the value either 1 or 2. The set of inputs of the models in $\boldsymbol{M}^{(j)}$ is $u_i^{sel(j)}$.

- *Step 2:* Construct the training data set $Z^{N^{est(j)}} = [\boldsymbol{Y}_i^{est}, \boldsymbol{U}_i^{sel(j),est}]$, and the validation data set $Z^{N^{est(j)}} = [\boldsymbol{Y}_i^{est}, \boldsymbol{U}_i^{sel(j),est}]$. The parameters N^{est} and N^{val} are the number of samples in the estimation and validation data set, respectively.

- *Step 3:* Compute the parameters of the 32 models of $\boldsymbol{M}^{(j)}$ using the PEM or MRI methods as shown below.

$$\begin{aligned} \theta^{pem} &= \underset{\theta}{argmin}\, V^{pem}(\theta, Z^{N_{est}}) \\ \theta^{mri} &= \underset{\theta}{argmin}\, V^{mri}(\theta, Z^{N_{est}}) \end{aligned} \tag{3.22}$$

- *Step 4:* Compute the n_p-step-ahead predicted output corresponding to each model of $\boldsymbol{M}^{(j)}$ using the iterative process such as in Equation 3.8 on the validation data set. Fit percentages corresponding to each of the predictions is computed using Equation 3.14. The model that provides the best fit percentage on the validation data set is selected as the candidate model as shown below.

$$M^{*(j)} = \underset{M \in \boldsymbol{M}^{(j)}}{argmax}\, f_{fp}(M) \tag{3.23}$$

The fit percentage corresponding to the candidate model is denoted by $f_{fp}^{*(j)}$. If the fit percentage of the candidate model of the current iteration is less than that of the past iteration, then iterations are stopped and the model corresponding to the previous iteration is selected to represent the dynamics of the subsystem, i.e. $M_i = M^{*(j-1)}$.

- *Step 5:* Compute the residual vector $\hat{\boldsymbol{e}}_{t|t+1}^{est}$ corresponding to the model $M^{*(j)}$ using the equation shown below.

$$\hat{\boldsymbol{e}}_{t|t+1}^{est} = \boldsymbol{Y}^{est} - \hat{\boldsymbol{Y}}_{t|t+1}^{est} \tag{3.24}$$

Construct the set of unselected inputs $u_i^{unsel(j)} = u_i^{pos} \cap (u_i^{sel(j)})^{\complement}$. Construct the difference vectors corresponding to $\hat{\boldsymbol{e}}_{t|t+1}^{est}$ and $\boldsymbol{U}_i^{unsel(j)}$ using Equation 3.19.

- *Step 6:* The difference vectors computed in Step 6, $\delta\hat{\boldsymbol{e}}_{t|t+1}^{est}$ and $\delta\boldsymbol{U}_i^{unsel(j)}$ are divided into subsets, with each subset containing n_{slots} data points. Cross correlation values are found between $\delta\hat{\boldsymbol{e}}_{t|t+1}^{est}$ and each input of $\delta\boldsymbol{U}_i^{unsel(j)}$ corresponding to each subset. The input with the maximum average cross correlation value is selected to augment the input set $u_i^{sel(j)}$ as shown in the equations below.

$$u_i^{aug} = \underset{u_i^{pos} - u_i^{sel}}{argmax}\ r(\delta\boldsymbol{e}^{est}, \delta U_i^{sel(j)}) \tag{3.25}$$

$$u_i^{sel(j+1)} = [u_i^{sel(j)}, u_i^{aug}] \tag{3.26}$$

- *Step 7:* Increment the value of the iteration variable j and repeat Steps 1 through 6. The algorithm stops at Step 4 when the augmented input set $u_i^{sel(j+1)}$ does not increase the fit percentage of the candidate model $M^{*(j)}$.

3.4 Black-Box Modeling Results

In this section, the results obtained by applying the automated modeling algorithm described in the previous section are presented. The modeling algorithm uses the linear parametric method, and the parameters of the model can be identified either by PEM or the MRI approaches. A comparison of the identification results obtained from models whose parameters are identified using the PEM and MRI methods, respectively are provided. In

addition to the two linear parametric approaches, the results obtained from the subspace identification method described in Section 3.2.2 are also provided for comparison. The models are developed by using data from the BEMS of a real working office building. A description of the office building and its HVAC system is first provided.

3.4.1 Description of the Office Building

Chapter 1 of the dissertation provided a classification of the building HVAC systems, where the components were divided into two levels, the Energy Conversion Level (ECL) and the Energy Distribution Level (EDL). The ECL corresponds to the components dealing with the primary cooling fluid such as chilled water (CHW). These components include the chiller, cooling towers, etc. The EDS components are associated with the conditioned air that is distributed to the building zones to provide occupancy comfort. The components of EDS include the AHU heat exchanger, the AHU fan, the thermal zones of the building, etc. In this dissertation only the components corresponding to EDS are considered. The reasons for including only the EDS components is provided in Chapter 1.

The office building for which black-box models are developed in the Chapter, is single-storied comprising 11 rooms. The floor plan of the building is shown in Figure 3.4. The EDS components of the office building being are shown with the help of a schematic in Figure 3.5 reproduced from Chapter 1. The office building is served by a single AHU that provides conditioned air to the rooms of the building. The AHU comprises a heat exchanger that utilizes CHW supplied at a fixed temperature of 44^0F by a campus-wide CHW distribution system. The CHW supply is used to cool a mixture of outside air and return air that is drawn by the AHU system. In order to maintain the air flow through the AHU a pressure differential is created with the help of a fan. The air that is cooled by the AHU heat exchanger is distributed with the help of Variable Air Volume (VAV) boxes placed in each room. The data corresponding to the EDL components available through

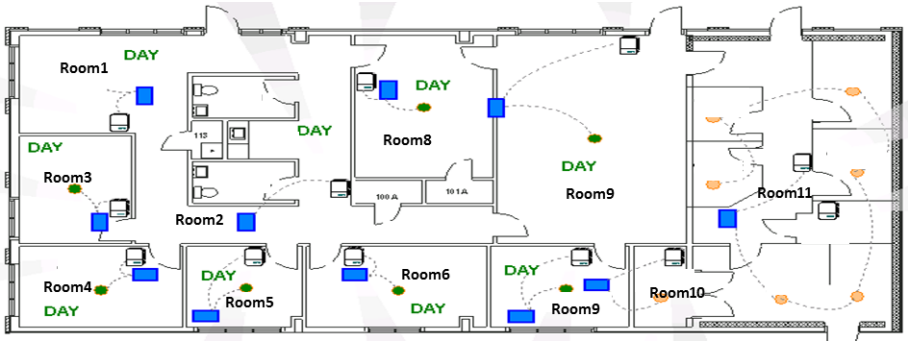


Figure 3.4: Layout of the EMO Building

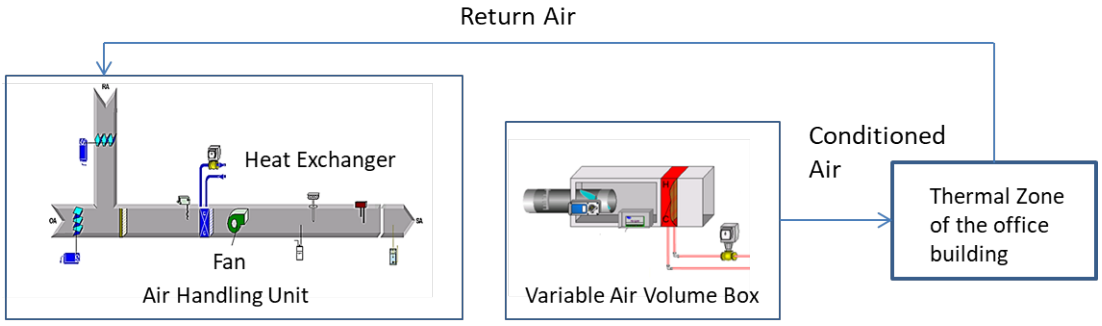


Figure 3.5: Components of the Office Building Corresponding to EDL

the BEMS of the office building is shown in Table.

In order to regulate the EDL components in Figure 3.5, the building employs a PI-type supervisory control that computes three setpoints, the end static pressure setpoint P_{eds}^{set} required to maintain the pressure differential of the AHU system, the AHU discharge air temperature T_{ahu}^{set} measured at the exit of the heat exchanger, and the volume flow rate setpoint through the VAV box of each room. The volume flow rate setpoints is represented by the vector $\boldsymbol{V}^{set} = [V_1^{set}, V_2^{set}, ..., V_{n_r}^{set}]$, where the number in the subscript is the room index and the n_r is the total number of rooms. A detailed description of how the setpoints are computed by the traditional PI-type control, and the MPC algorithms is provided in Chapter 4. The computation of the setpoints by MPC, however, requires models that describe

the behavior of the system. This section shows how the proposed modeling algorithm can be used to develop model for the HVAC components in corresponding to EDL.

The modeling of the dynamics of the components corresponding to EDL is performed by first constructing three subsystems, the AHU pressure subsystem, the AHU tempreature subsystem, and the room temperature subsystem. The subsystems are constructed based on the outputs which are of interest for implementing MPC. For example, one of the outputs of interest for implementing MPC is the fan energy cost. Modeling the fan energy cost requires a model of the end static pressure, and the total volume flow rate through the AHU. Hence the AHU pressure subsystem models the outputs, end static pressure P_{eds}, and the volume flow rate through the AHU V_{ahu} which is the sum of the volume flow rate through each of the VAV boxes. A description of the inputs and outputs corresponding to the aforementioned subsystems, and the results of applying the model on data from the real building are provided as follows.

3.4.2 AHU Pressure Subsystem

Figure 3.6 shows a schematic of the AHU pressure subsystem. The subsystem comprises the AHU fan, and the VAV boxes. The fan creates a pressure differential called the end static pressure P_eds which maintains a positive air flow through the AHU and VAV boxes. The outputs of interest of the AHU pressure subsystem, are the end static pressure P_{eds}, and the volume flow rate through the AHU V_{ahu}.

The end static pressure P_{eds} depends on several factors such as the fan speed, the construction of the AHU, the damper openings etc. Obtaining a black-box model by taking into consideration is unfeasible. The dynamics of the fan, however, are fast. A static model of the pressure can be used to sufficiently model the pressure subsystem over time scales that are of relevance for the building HVAC system. Hence in this research, An approximation is made by assuming that P_{eds} is actuated to its setpoint value P_{eds}^{set} by the

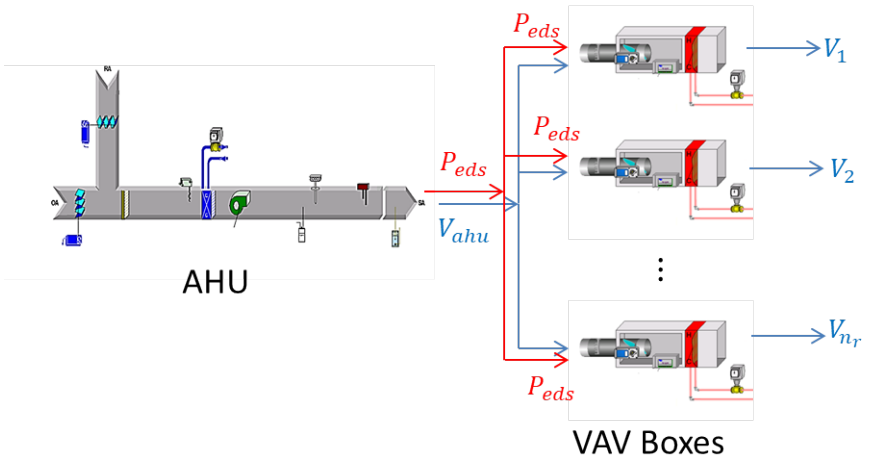


Figure 3.6: A Schematic of the AHU Pressure Subsystem

next control timestep as shown in Equation 3.27

$$\hat{P}_{eds}(t|t+1) = P^{set}_{eds}(t) \tag{3.27}$$

While modeling the volume flow rate through the AHU V_{ahu} an assumption is made that there is no loss of air through the duct system. Hence V_{ahu} is equal to the sum of the volume flow rates through the VAV boxes of each room as shown below.

$$\hat{V}_{ahu}(t|t+1) = \sum_{i=1}^{n_r} \hat{V}_i \tag{3.28}$$

Following the same reasoning as the one for modeling P_{eds}, only a static model for the volume flow rate is considered. The flow rate through the VAV box of room i (say) is dependent on the damper position D_i, and the pressure difference across the VAV P_{eds}. The flow rate V_i is proportional to the square root of the pressure P_eds and to the cross-

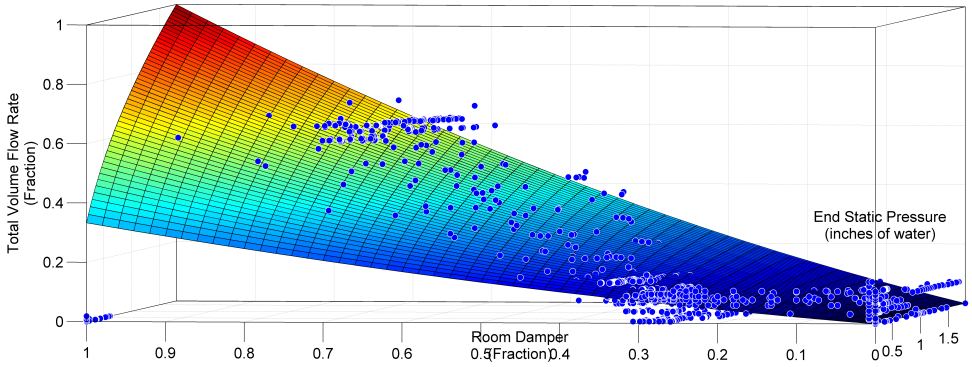


Figure 3.7: Regression Analysis Between Measured and Predicted Values of Volume Flow Rate of Room 1

sectional area of the damper opening. Hence, V_i is modeled as shown in Equation 3.29.

$$\hat{V}_i(t|t+1) = (a_i \cdot D_i(t|t+1) + b_i \cdot D_i^2(t|t+1)\hat{P}_{eds}(t|t+1) \tag{3.29}$$

The parameters a_i and b_i are found through regression analysis.

Data sampled at 5 minute intervals of the end static pressure setpoint P_{eds}^{set}, volume flow rate V_1, and damper position D_1 of the office building described in the previous section were collected for a period of 7 days. Since the end static pressure was not sampled by the BEMS, the setpoint values P_{eds}^{set} were used for regression analysis. The parameters a_1 and b_1 were obtaining by minimizing the quadratic norm of the error between predicted and measured volume flow rates. Figure 3.7 shows a comparison of the surface representing the predicted volume flow rates as a function of P_{eds} and D_1 with the measured values. Figure 3.8 shows a comparison of the measured and predicted values of the volume flow rate of room 1.

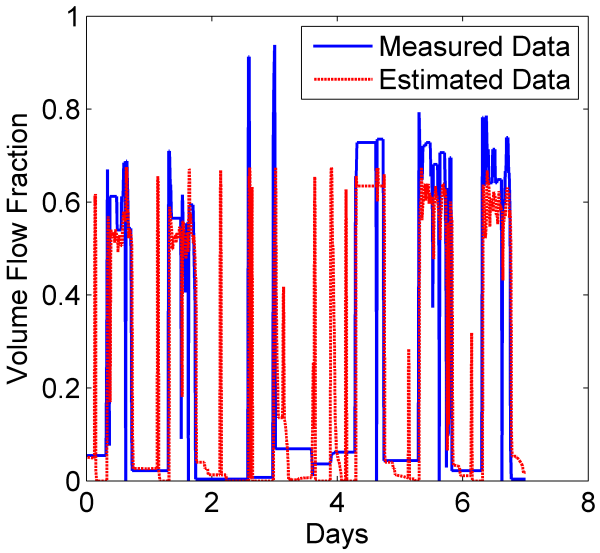


Figure 3.8: A Comparison of the Measured and Predicted Values of the Volume Flow Rate of Room 1

Table 3.2: Regression Values for Volume Flow Model of VAV 1

Parameter	Value
a_1	0.2665
b_1	0.4786

3.4.3 AHU Temperature Subsystem

Figure 3.9 shows the inputs and outputs of the AHU temperature subsystem. The output of interest of the AHU temperature subsystem is the discharge air temperature T_{ahu}. In this disseration a model of the dynamics of T_{ahu} is developed by applying the algorithm proposed in Section 3.

The first step in the algorithm after identifying the output is to create a list of possible inputs that affect its dynamics. In order to identify the list of possible inputs, the equation describing the conservation of energy principle across the AHU heat exchanger shown in

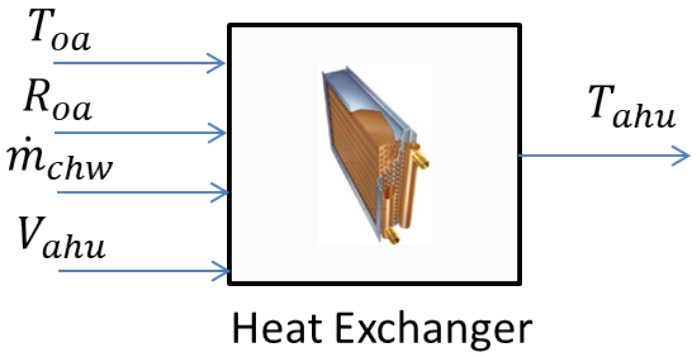


Figure 3.9: A Schematic of the AHU Temperature Subsystem

Equation is investigated

$$\dot{m}_{ahu} \cdot c_{air} \cdot \delta T_{ahu} = \dot{m}_{chw} \cdot c_{chw} \cdot \delta T_{chw} + \dot{Q}_{s,ahu} \tag{3.30}$$

where $\dot{m}_{ahu}$ and $\dot{m}_{chw}$ are the mass flow rates through the AHU heat exchanger of the air and CHW respectively, c_{air} and c_{chw} are the specific heat capacities of air and water repsectively, δT_{ahu} and δT_{chw} are the change in temperatures from outlet to inlet of the heat exchanger of the air and CHW, respectively. The symbol $\dot{Q}_{s,ahu}$ is used to represent the rate of heat loss to the surroundings. The factors affecting the discharge air temperature of the air thus include the ratio of the flow rates of CHW and air, i.e. $\frac{\dot{m}_{chw}}{\dot{m}_{ahu}}$ and the inlet water temperature $T_{chw,in}$. The mass flow rate of CHW is not available through the BEMS sensors of the office building being studied, hence it is expressed as a quadratic function of the CHW valve opening V_{chw}. In addition, the heat lost to the surroundings is assumed to be a function of the outside air temperature and humidity. The list of possible inputs thus considered in modeling the AHU temperature subsystem is shown in Table 3.3.

After constructing the list of possible inputs u_{ahu}^{pos} of the AHU temperature subsystem, data sampled at 5 minute intervals is collected for a period of 28 days. The first 21 days

Table 3.3: Inputs and Output for AHU Temperature Subsystem

Inputs u_i^{pos}	Output
T_{oa}	
V_{chw}/V_{ahu}	T_{ahu}
V_{chw}^2/V_{ahu}	
RH_{oa}	

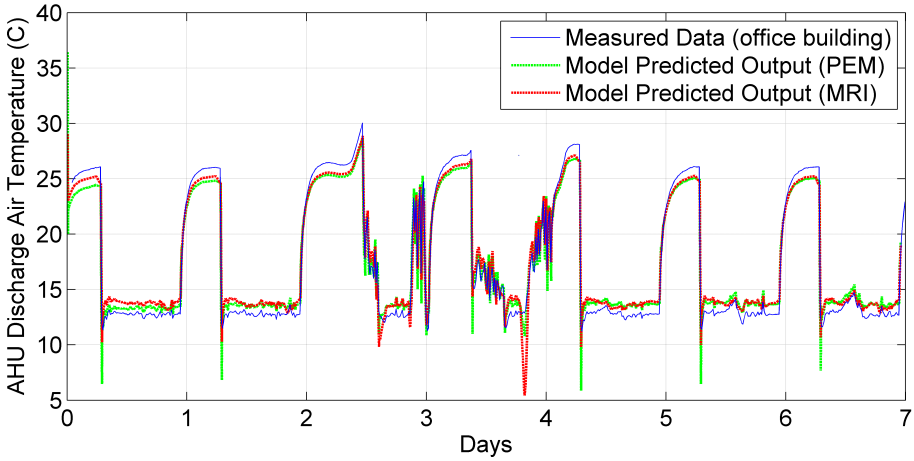


Figure 3.10: Measured and Predicted Values of the Discharge Air Temperature

are used for training the data, and the last 7 days are used for model verification. The result of applying the proposed modeling algorithm with PEM and MRI approaches, and the subspace identification method is shown in Figure 3.10. The model structure and order selected, and the fit percentages obtained corresponding to each approach is provided in Table 3.5.

3.4.4 Room Temperature Subsystem

The output of interest of the room temperature subsystem is as the name indicates the temperature of the room. Consider the differential equation used in [55] used to describe

the room temperature dynamics

$$M_i c_p \frac{\partial T_i}{\partial t} = m_{vav} c_p (T_{ahu} - T_i) + Q_{s,i} \tag{3.31}$$

where M_i is the thermal mass of the room i, c_p is the heat capacity, m_{vav}, and T_{vav} are the mass flow rate, and temperature of the conditioned air through the VAV of the room, and $Q_{s,i}$ are the heat loads on the room. From Equation 3.31 we see that the room temperature is impacted by the product of the mass flow rate and the discharge air temperature. Hence $V_i \cdot T_{ahu}$ is selected as the control input (assuming the density of air remains constant). The factors that could potentially impact $Q_{s,i}$ and are also measured by the BEMS sensors are the outside air temperature and humidity (T_{oa} and R_{oa}), and the temperature of the neighboring rooms (T_j, $j \in \{j_1, j_2, ...j_{n_i^{sur}}\}$). j_k is the k^{th} surrounding room,, and n_{sur} is the total number of surrounding rooms of room i. Hence the list of possible inputs considered in this paper while modeling the temperature of room i are $T_{oa}, RH_{oa}, T_j s$, and $V_i \cdot T_{ahu}$.

Another aspect to consider while modeling room temperatures is the high degree of collinearity which may exist between the different subsystems, which can result in numerical problems [56]. For example, two rooms in a building say room i, and j, may have the same temperature set points and work schedules. Since black-box modeling is a purely statistical technique, while modeling the temperature of room i, the model may incorrectly identify a large input gain corresponding to room j. In order to reduce the possibility of misidentifying gains due to collinearity, the difference in temperatures $T_j - T_i$ (corresponding to the example above) is chosen as the input. Table 3.4 contains the list of inputs considered for the room temperature subsystem.

After constructing the list of possible inputs, the modeling algorithm is applied on the data that is obtained through the BEMS of the office building. The duration of the training

Table 3.4: Inputs and Output for Room Temperature Subsystem

Inputs u_i^{pos}	Output
$T_{oa} - T_i$	
$V_i \cdot T_{ahu}$	T_i
$T_j - T_i \; j \in \{j_1, j_2, ..., j_{n_{sur}}\}$	

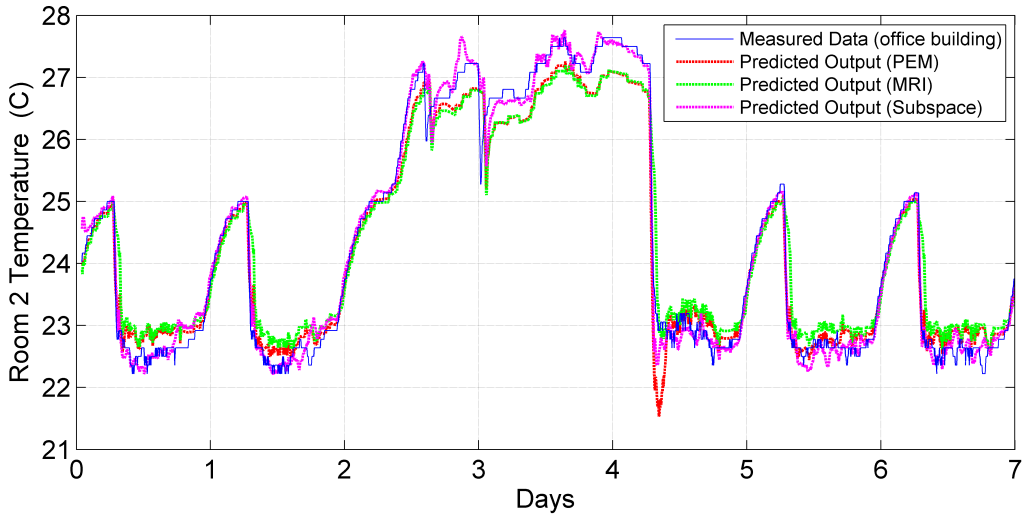


Figure 3.11: Measured and Predicted Temperature of Room 1

and validation data is the same as that for the AHU temperature subsystem. The results obtained by applying the proposed modeling algorithm with PEM and MRI approaches, and the subspace identification method is shown in Table 3.5. A comparison between the 12-step-ahead predicted output comoputed by the identified model, and the measured temperatures of room 1 on the validation

It is convenient to represent them by using equivalent state-space matrices as shown in

Table 3.5: Summary of Results Obtained by Implementing the Proposed Algorithm on the Different Subsystems

	PEM				MRI				Subspace
Room	u_i^{sel}	Structure	Order	Fit	u_i^{sel}	Structure	Order	Fit	Fit
1	$[T_{oa}, u_1^{ctr}]$	BJ	$[2,1,2,1]$	74.0	$[T_{oa}, u_1^{ctr}]$	ARX	$[2,1]$	78.5	85.6
2	$[T_{oa}, u_2^{ctr}, T_1]$	ARX	$[1,1]$	86.0	$[T_{oa}, u_2^{ctr}, T_1]$	ARMAX	$[1,2,2]$	86.0	88.5
3	$[T_{oa}, u_3^{ctr}, T_5]$	BJ	$[1,1,1,1]$	79.6	$[T_{oa}, u_3^{ctr}, T_1]$	ARMAX	$[2,2,2]$	75.9	83.1
4	$[T_{oa}, u_1^{ctr}]$	BJ	$[1,1,1,1]$	68.0	$[T_{oa}, u_4^{ctr}]$	ARMAX	$[2,1,1]$	68.4	76.1
5	$[T_{oa}, u_1^{ctr}, T_8]$	BJ	$[1,1,2,2]$	77.3	$[T_{oa}, u_1^{ctr}, T_2, T_3]$	ARX	$[2,2]$	77.7	84.9
6	$[T_{oa}, u_1^{ctr}, T_8]$	BJ	$[1,1,1,1]$	85.0	$[T_{oa}, u_1^{ctr}, T_2, T_3]$	ARMAX	$[2,1,1]$	77.9	85.5
7	$[T_{oa}, u_1^{ctr}, T_11]$	BJ	$[1,1,2,1]$	66.4	$[T_{oa}, u_1^{ctr}, T_2]$	ARMAX	$[1,2,2]$	63.8	70.3
8	$[T_{oa}, u_1^{ctr}]$	ARMAX	$[2,1,2]$	79.9	$[T_{oa}, u_1^{ctr}]$	ARX	$[2,1]$	79.2	85.6
9	$[T_{oa}, u_1^{ctr}]$	ARMAX	$[2,1,2]$	71.8	$[T_{oa}, u_1^{ctr}]$	ARMAX	$[2,1,1]$	71.2	82.9
AHU	$[T_{oa}, \frac{V_{chw}}{V_{ahu}}, \frac{V_{chw}^2}{V_{ahu}}]$	BJ	$[1,1,2,1]$	46.5	$[T_{oa}, u_1^{ctr}]$	ARMAX	$[2,1,1]$	71.2	82.9

Equation 3.32.

$$\begin{aligned} x(k+1) =& A_i x(k) + B_i u_i^{sel}(k) + K_i e(k) \\ y(k) =& C_i x(k) \end{aligned} \tag{3.32}$$

In order to reduce the possibility of collinearity, some of the inputs were expressed as a difference in temperature with the output of the subsystem. For example, while modeling the temperature of room 1, say the inputs selected were $u_1^{sel} = [T_{oa}, u_1^c, T_2 - T_1, T_3 - T_1]'$. The output of the subsystem is T_1, and some of the selected inputs are also defined in terms of the output. A more convenient state-space representation would be to express the subsystem inputs without containing the output term. So instead $T2 - T1$, the input would just be $T2$.

In order to be able to express the inputs independent of the output, the vector E^i is introduced to identify the inputs which contain T_1 in their expression. For the example provided above, E^1 would be as shown below.

$$E^1 = [0\ 0\ 1\ 1]' \tag{3.33}$$

Substituting the value of $T_i = C_i x(k) + e(k)$ in Equation 3.32, we obtain the state space representation of the output of subsystem i as shown in Equation 3.34.

$$\begin{aligned} x(k+1) =& (A_i - B_i E_i C_i)x(k) + B_i u_i^{sel}(k) + (K_i - B_i E_i)e(k) \\ y(k) =& C_i x(k) \end{aligned} \tag{3.34}$$

3.5 Model Verification by Applying MPC in a Simulation Environment

The previous section provided the results of applying the modeling algorithm on data obtained through the BEMS of a real office building. Black-box models are developed for the AHU pressure subsystem, AHU temperature subsystem, and the room temperature subsystem. A static model is used to predict the behavior of the AHU pressure subsystem, and the modeling algorithm proposed in Section 3 is used to develop dynamic models of the AHU discharge air temperature, and room temperatures. The dynamic models were developed using the PEM and MRI approach, and the subspace identification method. The modeling approaches provided good predictions of the output when applied to data that was collected from the BEMS of an office building.

The main objective of developing the models, though, is not just to be able to predict the outputs but to be able to use them in model-based control. Since the models were developed corresponding to a certain operating condition, there is no guarantee that the models will have good prediction or control capabilities when a new control methodology is employed or when the operating conditions change. Hence in this section the modeling methods are tested in their ability to be used for model-based control under different operating conditions. Since testing the modeling and control methods under different operating conditions is not feasible on a real working office building, a high-fidelity simulation system is used instead. A schematic of the simulation environment corresponding to EDL is shown in Figure 3.12. The simulation environment comprises white-box models of the

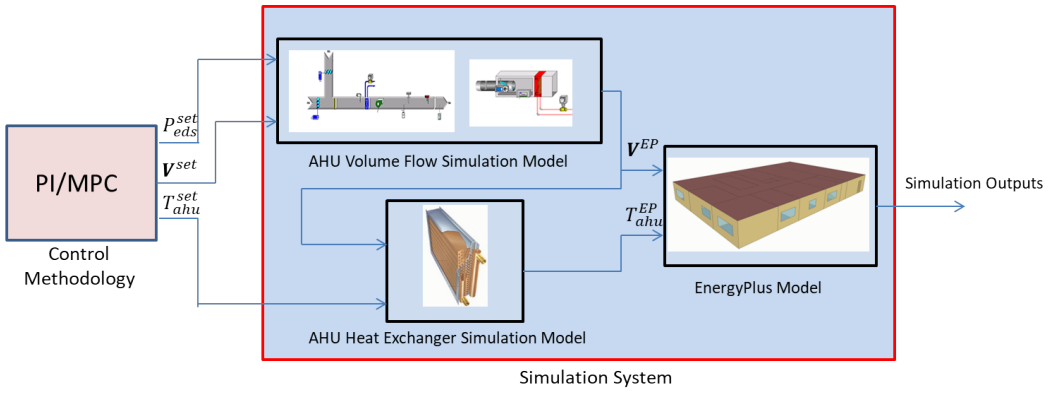


Figure 3.12: Simulation System to Test the Performance of Different Modeling Methods by Applying MPC

AHU heat exchanger, and an EnergyPlus model of the office building described in Section 3. The The AHU pressure system is simulated using the static model described in Section 4.

The simulation environment is regulated by a control methodology that computes the end static pressure setpoint P_{eds}^{set}, the flow setpoints to each room $\boldsymbol{V}^{set} = [V_1^{set}, V_2^{set}, ..., V_{n_r}^{set}]$, and the AHU discharge air temperature setpoint T_{ahu}^{set}. In Chapter 4 two types of control methodologies are tested on the simulation environment, traditional PI-type control, and MPC. A detailed description of the simulation environment, and the process by which the control methodologies compute the aforementioned setpoints are provided.

Results of applying MPC are included in this chapter to compare the modeling methods for their use in model-based control under different operating conditions. Models of the room temperatures are developed from the simulation data using the proposed algorithm with PEM and MRI approaches, and the subspace identification method. The three sets of models are then used to implement MPC on the simulation system. The comparison of the models is based on the cost of operation as computed by the simulation system.

3.5.1 Modeling Results on Simulation Environment

Simulations are run using the traditional PI-type control methodology for a period of 30 days from June 1st to June 30th of a typical year in College Station, Texas. The three approaches of developing black-box models (PEM, MRI, and Subspace Identification) are used to develop models from the simulation data. Figure shows the simulated and a 12-step-ahead predicted temperature of room 4 using the PEM approach. Table 3.6 shows a comparison of the fit percentages obtained while computing the 12-step-ahead predicted on the validation data. Although, all the modeling approaches provide accurate predictions, the subspace identification method provides the best average fit percentage of 82.8%.

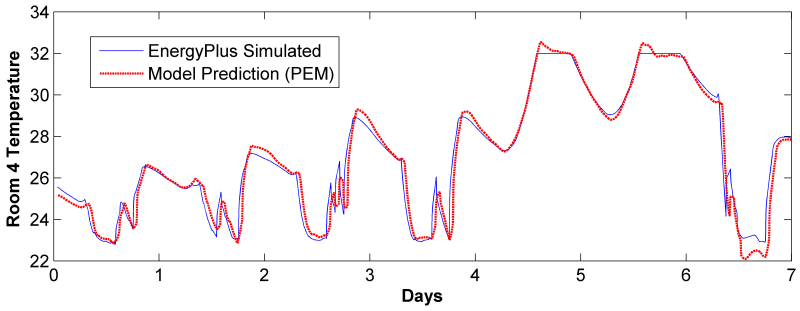


Figure 3.13: Model Predicted and EnergyPlus Simulated Temperature of Room 4

3.5.2 Description of the Operating Conditions used for Simulation

The models from the three approaches above are then tested for their use in model-based control. The tests include applying the three sets of models to implement MPC on the simulation system under different operating conditions. A total of 9 simulation experiments with unique operating conditions are performed using each of the three types of models. The 9 unique operating conditions are obtained by varying three aspects of the

Table 3.6: Prediction Fit Percentages Obtained by Applying the Modeling Algorithm on EnergyPlus Data

Room Number /Modeling Approach	PEM	MRI	SS
1	80.1	81.6	81.5
2	82.5	83.7	83.9
3	80.7	80.7	82.9
4	79.1	80.5	80.4
5	82.1	82.5	82.6
6	83.8	82.6	84.8
7	82.8	83.9	83.9
8	82.7	84.8	85.6
9	79.8	80.2	81.3
10	79.0	79.2	81.4

simulation, the objective function parameters, the outside air temperature and humidity, and occupancy. Before providing a description of the operating conditions, a brief description of the objective function minimized by MPC is provided. In this dissertation, the cost of operation of the building HVAC system is measured by taking into consideration the cost of running the AHU fan and heat exchanger, and the cost of not meeting the thermal comfort requirements of the occupants. The cost of operation at time step t is thus expressed as shown in Equation 3.35.

$$\boldsymbol{J}^{op}(t) = r_{fan} \cdot \boldsymbol{E}^{fan}(t) + r_{he} \cdot \boldsymbol{E}^{fan}(t) + r_{sal}\boldsymbol{E}^{comf}(t) \tag{3.35}$$

where $\boldsymbol{J}^{op}$ is the total cost of operation, $\boldsymbol{E}^{fan}$ and $\boldsymbol{E}^{he}$ are the energy associated with the AHU fan and heat exchanger operation, and $\boldsymbol{E}^{comf}$ is a measure of occupancy comfort. The parameters r_{fan} and r_{he} correspond to the unit price of fan and heat exchanger energy, and r_{sal} is the weight put on the occupants comfort. In this dissertation r_{sal} is computed based on the occupant's yearly salary.

MPC seeks to minimize the cost of operation over a future time period called as a prediction horizon. If n_p is the number of time steps in the prediction horizon, then the estimated cost over that period can be expressed as shown in Equation 3.36.

$$\hat{\boldsymbol{J}}^{op}(t) = \sum_{i=1}^{n_p} r_{fan} \cdot \hat{\boldsymbol{E}}^{fan}(t|t+i) + r_{he} \cdot \hat{\boldsymbol{E}}^{fan}(t|t+i) + r_{sal}\hat{\boldsymbol{E}}^{comf}(t|t+i) \tag{3.36}$$

MPC uses predictions of future disturbances such as outside air temperature and humidity, occupancy, etc. and the black-box models to obtain a relationship between the estimated cost of operation and the control setpoints to be optimized. Mathematical optimization is then performed to obtain the set of control inputs that provide the least value of $\hat{\boldsymbol{J}}^{op}(t)$ as shown in Equation 3.37.

$$\boldsymbol{u}^{mpc}(t) = \underset{\boldsymbol{u}^{set}}{argmin}\ \hat{\boldsymbol{J}}^{op}(t) \tag{3.37}$$

where $\boldsymbol{u}^{mpc}$ is the optimal values corresponding to the set of control setpoints $\boldsymbol{u}^{set}$ computed by MPC. In this chapter the set of control setpoints to be optimized $\boldsymbol{u}^{set}$ comprises P_{eds}^{set}, $\boldsymbol{V}^{set}$, and T_{ahu}^{set}.

Simulation experiments 1 through 4 are performed corresponding to the weather conditions on June 1^{st} of the EnergyPlus weather file of College Station, TX. The weather conditions represent a typical hot and humid day in Texas. The simulations 1 through 4 differ in the objective function parameters r_{fan}, r_{he}, and r_sal. The parameters r_{fan}, and r_{he} are assigned the values that were determined to be the best estimates of real world unit costs of fan and cooling energy per time step. The parameter r_{sal} is the average salary of the occupant per time step if the yearly salary is r_{sal}^{yr}. In simulations 1 through 4 the parameter r_{sal}^{yr} is changed by an order of magnitude, with a smaller value of r_{sal}^{yr} placing a smaller penalty on occupancy discomfort.

The process of varying r_{sal}^{yr} is repeated in simulations 5 through 8. The simulations

differ from the first four in the outside weather conditions. The weather conditions correspond to August 1^{st} of the EnergyPlus weather file of College Station. Finally, simulation 9 represents a special occupancy condition where only room 2 is occupied. A summary of the operating conditions and cost parameters corresponding to each simulation is summarized in Table 3.7.

Table 3.7: Parameters Corresponding to Various Operating Conditions

Operating Condition	Max Temp (^0C)	Max Humidity (%)	r_{fan}	r_{he}	r_{sal}^{yr}	Occupancy
1	35	93	1.101e-6	1.078e-7	20,000	Full
2	35	93	1.101e-6	1.078e-7	2,000	Full
3	35	93	1.101e-6	1.078e-7	200	Full
4	35	93	1.101e-6	1.078e-7	20	Full
5	40	89	1.101e-6	1.078e-7	20,000	Full
6	40	89	1.101e-6	1.078e-7	2,000	Full
7	40	89	1.101e-6	1.078e-7	200	Full
8	40	89	1.101e-6	1.078e-7	20	Full
9	35	93	1.101e-6	1.078e-7	20	Room 1

3.5.3 MPC Simulation Results

Three simulations are run for each operating condition in Table 3.7. The three simulations correspond to the three different black-box modeling approaches (PEM, MRI, and subspace identification) used to apply MPC to the simulation environment. Results corresponding to the operating condition 1 are first provided.

Operating condition 1 corresponds to a maximum outside air temperature of 35^0C and a maximum relative humidity of 93%. The cost parameters r_{fan}, and r_{he} represent the best estimates of the cost of operation of the AHU fan, and heat exchanger, respectively. The parameter r_{sal}^{yr} refers to the occupant's yearly salary which is set at \$20,000.

The parameter r_{sal}^{yr} is used to compute the cost of occupancy comfort. The occupancy comfort cost of a particular occupied room is designed so that if the temperature of that room lies between a high temperature setpoint, and a low temperature setpoint then the cost is zero. If the temperature of lies outside the two temperature setpoint bounds, then a nonlinear relationsip exists between the comfort cost and the temperature, and the occupant's yearly salary. Greater the salary greater the penalty for the room temperature not lying between the temperature bounds. Thus an efficient HVAC operation would correspond to the room temperature tracking the higher setpoint as this results in zero comfort cost and at the same time would require the least fan and cooling energy. The control performance of the MPC simulations corresponding to operating condition 1 using the three modeling approaches is presented below.

Figure 3.14 shows the simulated temperature of room 1 when MPC is applied using the three modeling approaches. The figure also shows the two temperature setpoint bounds, and the outside air temperature during the simulation period. MPC simulations that use the PEM and MRI approaches are able to closely track the upper bound of the temperature setpoint. By looking at the magnified portion of the figure, we see that MPC that uses the PEM approach is able to more closely track the higher temperature setpoint. When MPC is applied with the subspace models, however, no thermal comfort is provided to the occupants. The room temperature drifts towards the outside air temperature conditions.

Figure 3.15 shows the end static pressure and discharge air temperature setpoint computed by MPC using the different black-box modeling approaches. The figure shows MPC using the PEM and MRI approaches seeking the optimal combination of end static pressure and discharge air temperature setpoints, whereas, MPC with subspace models provides no cooling and assigns the setpoint of the end static pressure to the maximum permissible value. The comfort cost, the cost of the AHU fan and heat exchanger cooling energy, and the total cost of operation corresponding to the three modeling approaches are shown in

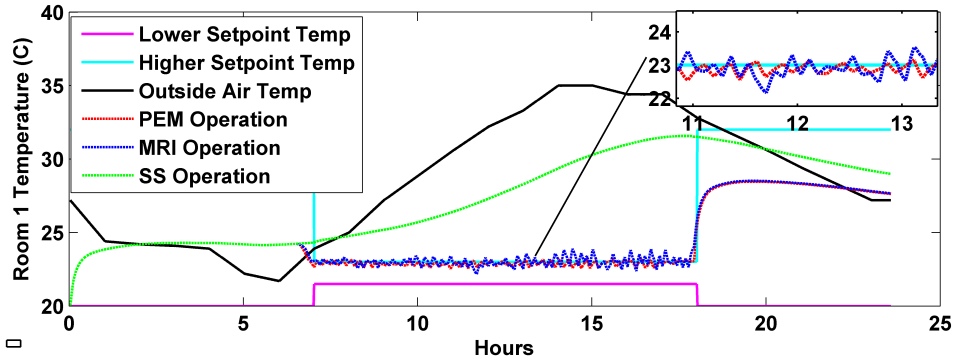


Figure 3.14: Simulation Temperatures of Room 1 with the Application of MPC Using Different Modeling Approaches

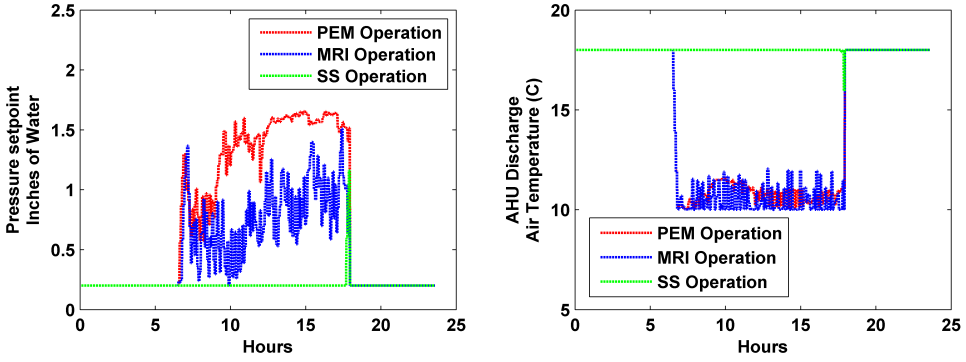


Figure 3.15: End Static Pressure and Discharge Air Temperature Computed by MPC Using Different Modeling Approaches

Figure

The results obtained by running simulations corresponding to the operating conditions 1 through 9 are summarized in Table 3.8. MPC applied using models developed by the PEM and MRI methods provided consistent good performance under varying operating condtions, whereas MPC applied using the subspace modeled fared poorly in all simulations as it did not provide any cooling to the thermal zones. The results indicate that a good prediction fit percentage is not a sufficient condition for a model to be used for purpose

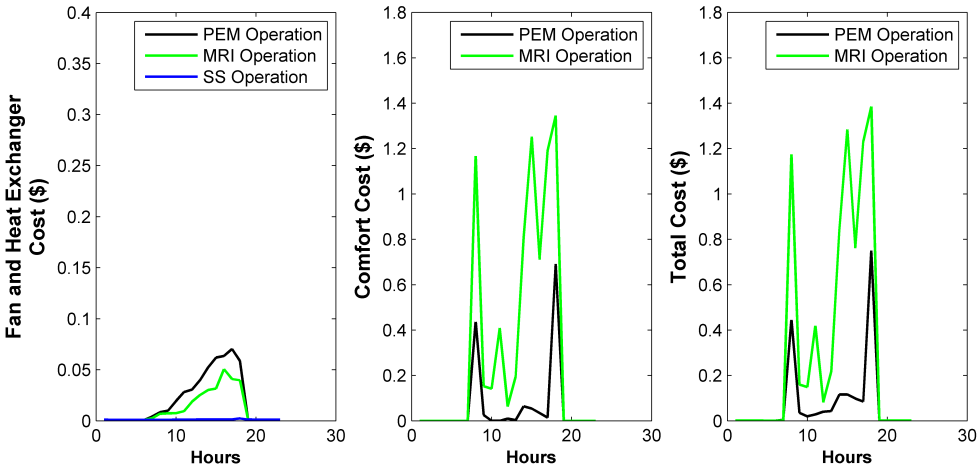


Figure 3.16: Comfort Cost, Cost of AHU Fan and Heat Exchanger Cooling Energy achieved by MPC Using Different Modeling Approaches

of application of control. The results in Table 3.8 show that the subspace identification percentage provided the best fit percentages on the data used to develop the model. The problem of collinearity that is associated with black-box modeling is probably the main reason for the subspace identification method to have identified a model not suited for the purpose of control.

Table 3.8: Costs Achieved by Applying MPC on the Simulation System Using Different Modeling Approaches

Operating Condition	Comfort PEM	Cost MRI	($) SS	AHU Fan PEM	and MRI	HE cost ($) SS	Total PEM	cost MRI	($) SS
1	1.34	7.44	273.25	0.46	0.28	0.03	1.79	7.72	273.27
2	0.27	1.36	27.53	0.35	0.20	0.03	0.63	1.56	27.56
3	0.09	1.73	2.75	0.17	0.05	0.03	0.26	1.78	2.78
4	0.26	0.17	0.27	0.03	0.05	0.03	0.29	0.22	0.30
5	1.8	8.09	277.96	0.48	0.31	0.03	2.28	8.40	278.00
6	0.28	1.71	27.80	0.35	0.35	0.03	0.64	1.93	27.83
7	0.11	1.80	2.78	0.18	0.18	0.03	0.29	1.85	2.81
8	0.26	0.27	0.28	0.03	0.03	0.03	0.30	0.30	0.31
9	0.11	0.27	31.94	0.43	0.16	0.03	0.53	0.43	31.94

3.6 Conclusions

Black-box modeling approaches offer the simplest methods of developing models for large scale building HVAC systems. In this thesis, a modeling algorithm was proposed that can automate the process of selecting the model order, structure and the inputs to the various subsystems either using the PEM, or MRI approaches. The subsystem models identified by the linear parametric methods can be combined to form a centralized model. The identified models were then verified by comparing multi-step-ahead predictions and measured values of the outputs of the subsystems using data from a real office building. A centralized room-temperature model was also identified using a MIMO approach with subspace identification algorithms. The results show that black-box modeling approaches have the capability to provide good predictions using data from normal working conditions. Subspace algorithms provided the best fit percentages among the three identification approaches.

A good prediction model does not necessarily translate into a good model for control. In order to reflect some of the underlying physics for each subsystem. The proposed algorithm using the linear parametric approach takes into account some of the underlying physics that govern each subsystem. This is done by including inputs that are most likely to impact the outputs, and using difference in temperatures as inputs in order to reduce the possibility of collinearity. The MIMO system identified using subspace identification algorithm however, did not consider any of the physics principles. As a result MPC implemented on a simulated model of the office building in EnergyPlus showed that the PEM and MRI models performed much better than subspace models in reducing the cost of the objective function under different operating conditions.

4. CENTRALIZED MPC

The preceding chapter provided a methodology to accurately predict building HVAC system behavior with the help of statistical models developed using black-box system identification techniques. The models were verified using both simulation data, and data from a real office building. The primary use of the black-box models, however, is not just to be able to predict system behavior, but to be able to use those predictions to implement optimal model based control methodologies such as MPC. This chapter provides a description of how the identified models are used to implement MPC to regulate the components corresponding to the energy conversion level of the building HVAC systems. Simulation experiments are then carried out using high-fidelity models of the HVAC components to demonstrate the capabilities of MPC where the predictions are obtained from black-box models. A comparison of MPC with the existing PI-type control methodology is also performed on the simulation test-bed.

4.1 Simulation Models

Chapter 1 of the thesis provided a description of how the building HVAC systems can be classifed into two levels, namely, the energy conversion level and the energy distribution level. This thesis seeks to demonstrate the utility of implementing MPC derived from black-box models by considering only the components in the energy distribution level. The components corresponding to the energy distribution level are reproduced in Figure 4.1 for ease of reference. Implementation of MPC only on the energy distribution level serves as a first step to a more widespread implementation where components from the energy conversion level are also regulated by MPC.

Ideally the experimental test-bed for demonstrating the utility of MPC would be a real building system as simulation models do not capture all the dynamics of the real system.

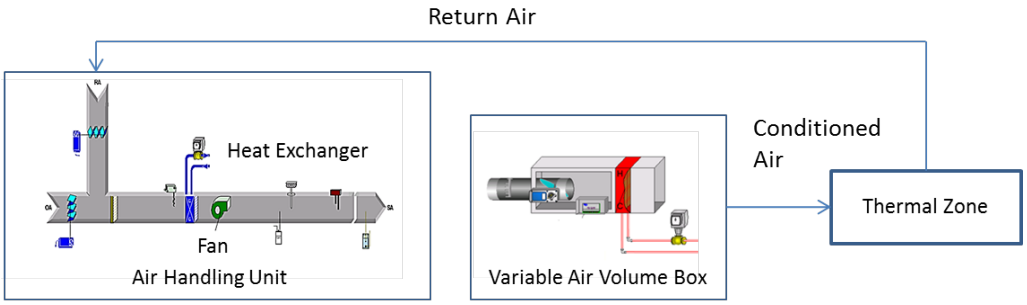


Figure 4.1: HVAC Components Corresponding to the Energy Distribution Level

In this thesis, however, high fidelity simulation models of the HVAC components are used as an experimental test-bed instead. To the best knowledge of the authors, the use of black-box models to implement centralized MPC on real HVAC buildings has not been performed before. Hence the use of simulation models would serve as a proof-of-concept before the control methodology can be applied to real buildings.

Successful demonstration of the proposed control methodology on simulation systems would make building managers more accepting to the idea of implementing black-box driven MPCs on real world building HVAC systems. In addition to serving as a stepping stone, the simulation environment allows the control methodologies to be tested under wide range of operating conditions. For example in the simulation environment, parameters such as outside air conditions, relative humidity, occupancy, etc can be varied between a wide range of values. Confidence in the control methodology increases when it can be demonstrated that a satisfactory performance of the HVAC components can be achieved under such wide range of conditions.

In this chapter, a juxtaposition of two types of control methodologies, namely, PI and MPC is made by comparing their performance using simulation models. Figure 4.2 shows a schematic of how either of the two aforementioned control systems are applied to the simulation system. The control methodologies are used to compute three types of set-

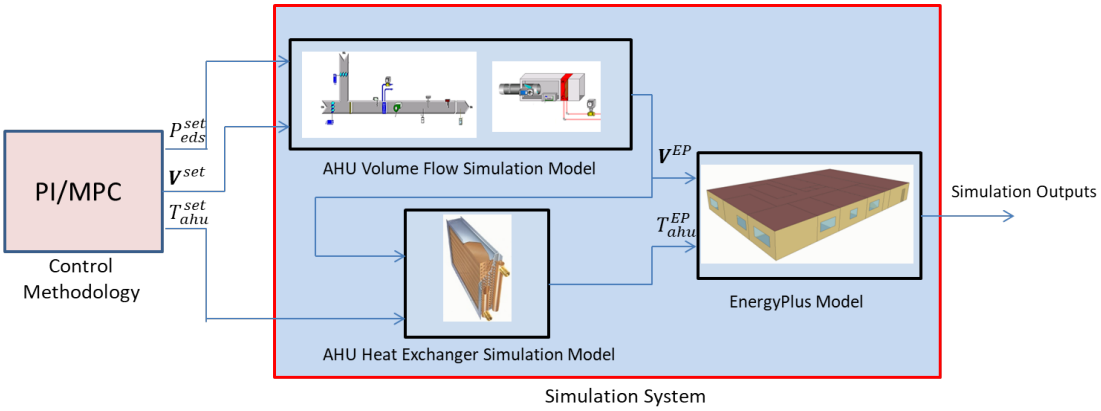


Figure 4.2: A Schematic of the Control Methodology and Simulation System

points namely, the AHU discharge air temperature setpoint T_{ahu}^{set}, the AHU end static pressure setpoint P_{eds}^{set}, and the volume flow rate setpoint through the VAV box of each room represented by the vector $\boldsymbol{V}^{set} = [V_1^{set}, V_2^{set}, ..., V_{n_{sel}}^{set}]$ where V_i^{set} is the flow rate setpoint of room i, and n_{sel} is the number of rooms of the office building whose flow rates are being optimized. The setpoints are computed at fixed intervals of time referred to as the control time step t_c.

The aforementioned setpoints are used to regulate the three subsystem of the energy distribution level. A brief description of the simulation models of the components of the energy distribution level used in this thesis is provided below.

4.1.1 AHU Volume Flow Simulation Model

The objective of the AHU volume flow simulation model is used to estimate the volume flow $V^{EP} = [V_1^{EP}, V_2^{EP}, ...V_{n_{sel}}^{EP}]$ of the conditioned air that is distributed through the VAV box placed in each room, given the volume flow setpoints $\boldsymbol{V}^{set}$ and the end static pressure setpoint P_{eds}^{set} computed by the control system. The superscript EP refers to the EnergyPlus environment. There are two main types of components of the AHU volume flow system, namely, the AHU fan, and the VAV box which is placed in each room. The

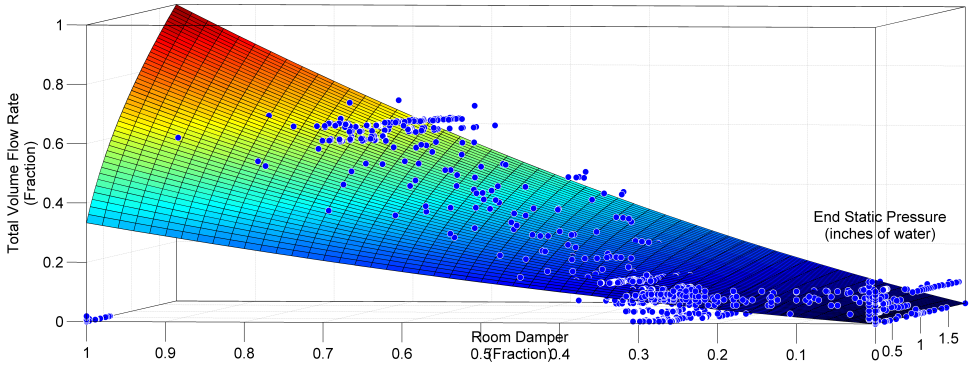


Figure 4.3: Recorded Data and Regression Fit of End Static Pressure, Damper Fraction, and Volume Flow Fraction

AHU fan creates a pressure differential P_{eds} which causes a mixture of outside and return air to be drawn to the AHU heat exchanger and finally through the VAV boxes. The fan speed is regulated by a PI controller in order for P_{eds} to track the setpoit P_{eds}^{set}. In this thesis, an assumption is made that the dynamics of the fan are much faster compared to the AHU heat exchanger, and the room temperature dynamics. As a result, while simulating the fan model it is assumed that the desired end static pressure setpoint is reached within the next control time step.

The next step in the simulation model is to estimate the volume flow rate through the VAV box of room i denoted by V_i^{EP} given an end static pressure P_{eds}, and a volume flow rate setpoint V_i^{set}. While the setpoints are computed at every control time step, the outputs of the simulation model are computed at ever simulation time step. The simulation time step referred as t_s corresponds to the time step used by the EnergyPlus simulation. In this thesis, the control time step is selected to be 5 minutes, whereas the simulation timestep is 1 minute.

The volume flow rate V_i^{EP} depends primarily on two factors, namely, the damper opening and the pressure differential created by the AHU fan P_{eds}. The flow rate through

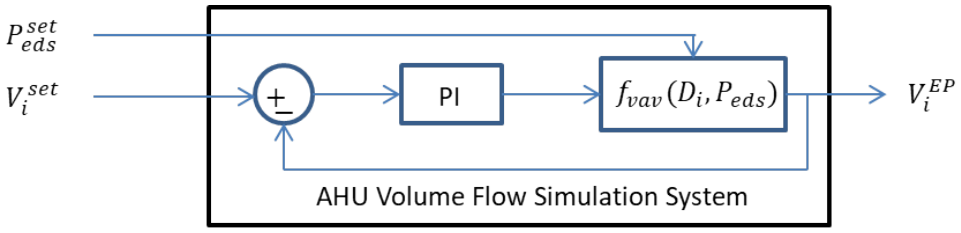


Figure 4.4: Flow Diagram Showing the AHU Volume Flow Simulation Model

the VAV box is proportional to the cross-sectional area of the damper opening, and to the square root of the pressure difference across the damper. The parameters corresponding to the relationship between volume flow V_i^{EP}, damper opening D_i, and P_{eds} were experimentally obtained by performing regression analysis on data obtained from a real office building in Chapter 3. The relationship is reproduced in Equation 4.1 for ease of reference. In the equation, a_i and b_i are parameters obtained from the regression analysis. Figure 4.3 shows the regression fit between the damper position, volume flow rate and end static pressure setpoints for one of the rooms obtained using data from the real office building .

$$V_i = (a \cdot D_i + b \cdot D_i^2) \cdot \sqrt{P_{eds}} \tag{4.1}$$

At each simulation time step, the damper position is computed with the help of a PI controller driven by the error between the setpoint V_i^{set} and the simulated output V_i^{EP} from the previous simulation time step. Substituting the damper position computed by the PI controller, and the end static pressure in Equation 4.1, the value of the current simulation flow rate is estimated. A schematic representation of the process used to compute V_i^{EP} is shown in Figure 4.4.

4.1.2 AHU Heat Exchanger Simulation Model

The heat exchanger of the AHU of the building under consideration uses chilled water (CHW) as a coolant that cools a mixture of outside air and return air drawn in by the pressure differential created by the AHU fan. CHW with a temperature of around 6.5^0C is supplied to the AHU heat exchanger from a campus-wide water cooling system. The objective of the AHU simulation model is to simulate the aforementioned water-cooled heat exchanger by estimating the discharge air temperature of the AHU T_{ahu} at every simulation time step, when a setpoint temperature T_{ahu}^{set} is provided by the control system. A local PI controller provides the actuation required for the discharge air temperature to track the setpoint value by regulating a CHW damper that controls the mass flow rate of the CHW (m_{chw}). Other than the mass flow rate of the CHW, T_{ahu} depends on other factors such as the supply water temperature $T_{w,in}$ and the mass flow rate $(\dot{m}_a)$, temperature $(T_{a,in})$, and humidity $H_{a,in}$ of the air being cooled by the heat exchanger. A schematic of the inputs and outputs of the AHU simulation model are shown in Figure 4.5 . The simulation model used in this thesis also provides as outputs the relative humidity of the discharge air $H_{a,out}$ and the outlet water temperature $T_{w,out}$. A brief description of the physics used to develop the model is provided below. A more complete description of the model can be obtained from [41].

The simulation model uses a finite volume approach to model the air and water temperatures through the heat exchanger. The basic equations used to describe the heat transfer process was derived by Hansen [57] and Zhou [48]. The chilled water tubing of the heat exchanger is divided into N sections, and for each section the water and coil temperature dynamics are calculated using Equations 4.2, and 4.3, respectively. The temperature of the water is affected by the convective heat transfer with the coil surface, and also the temperature and flow rate of the water entering the finite volume. The coil surface temperature

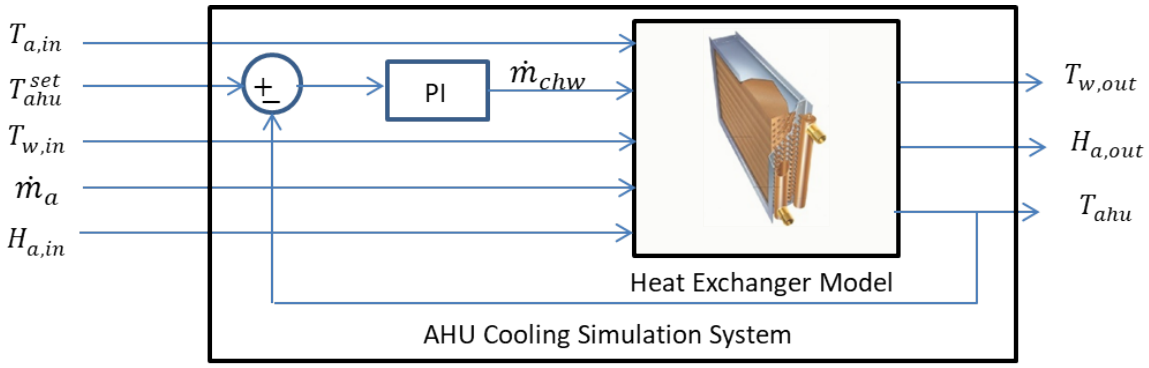


Figure 4.5: Flow Diagram Showing the AHU Cooling Simulation Model

at a particular section is affected by the the convective heat transfer with the air and water passing through each section.

$$C_w \cdot \frac{\partial T_w}{\partial t} + c_{p,w} \cdot q(\delta) \cdot \frac{\partial T_w}{\partial x} + h_w \cdot P_t(T_w - T_c) = 0 \tag{4.2}$$

$$C_c \cdot \frac{\partial T_c}{\partial t} + c_{p,a} \cdot \dot{m}_a \cdot \epsilon_a \cdot (T_c - T_{a,in}) + \kappa(T_c - T_w) = 0 \tag{4.3}$$

In the equations above symbols $C, c_p, T, q, \delta h, P$ and ϵ stand for specific heat capacity, heat capacity, temperature, mass flow rate, valve position, convective coefficient, inner perimeter, and heat transfer effectiveness, respectively. The subscripts, $w, t, c,$ and a stand for water, tubing, coil, and air, respectively.

The temperature of the air exiting each finite volume of the heat exchanger is given by Equation 4.4.

$$T_{a,out} = T_{a,in} + \epsilon \cdot (T_c - T_{a,in}) \tag{4.4}$$

The discharge air temperature exiting the heat exchanger is obtained by taking the average of all the N sections. A SIMULINK model that uses the above finite volume approach was developed by Liang [], and is used as the AHU heat exchanger model in this thesis. The gains of the PI controller used to regulate the mass flow rate were tuned by trial and error.

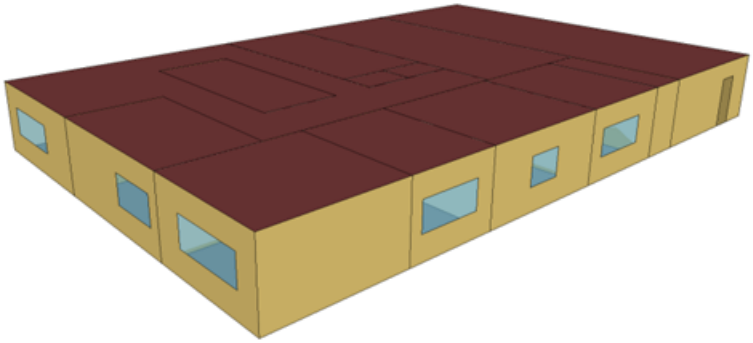


Figure 4.6: Isometric View of the EnergyPlus Model of the Office Building

4.1.3 EnergyPlus Model

An EnergyPlus model shown in Figure 4.6 is used to simulate the room temperatures of the office building described in Chapter 3. The first step in developing the EnergyPlus model is to define the parameters of the model that remain constant throughout the simulation period. These parameters can be broadly divided into three types, namely, building parameters, system parameters, and plant parameters. There are several parameters associated with each of the aforementioned type. For example, the building parameters include building dimensions, materials, placement of windows and doors, etc. The system parameters which describe the components of the energy distribution level include the size of the VAV boxes, the AHU fan power, the power rating of the heat exchangers etc. The plant parameters which describe the components of the energy conversion level include the chiller capacity, the cooling tower capacity, etc. The comprehensive list of parameters allows the EnergyPlus simulations to accurately capture the dynamics of the real building. In cases where information about the real building is not available default values are used to run the simulation.

In addition to the time-constant parameters described above, the room temperature dynamics also depends on time-varying inputs which can themselves be classified as dis-

turbance and control inputs. The disturbance inputs constitute all factors which influence the room temperature dynamics but whose values cannot be manipulated by the control system. For example, the outside air temperature, humidity, wind speed, occupancy etc. influence the room temperature dynamics but their values cannot be regulated. In this chapter the two control methodologies PI and MPC are compared by running several simulations under varying operating conditions achieved by using different sets of disturbance inputs.

Control inputs corresponding to the EnergyPlus simulations are inputs whose values can be varied during each simulation step. The purpose of using the EnergyPlus simulation model in this chapter is to study how the room temperature and humidity evolve when subject to the conditioned air provided by the volume flow and temperature simulation systems. Thus the control inputs to the EnergyPlus simulatin in this chapter are the volume flow rates $\boldsymbol{V}^{EP}$ and the discharge air temperature T_{ahu}.

The EnergyPlus software, however, is not built for testing and comparing control methodologies. $\boldsymbol{V}^{EP}$ and T_{ahu}^{EP} cannot be directly fed into the simulation. The control inputs that are generally used to run the simulations are instead room temperature setpoints. EnergyPlus assumes perfect knowledge of the environment and calculates a cooling load of each of the rooms based on the room temperature setpoints. The simulation engine then makes use of the plant and system description to compute its own discharge air temperature and flow rates. Since the cooling load calculations are based on a perfect knowledge of the thermal loads acting on the building, such a setup does not accurately reflect a real-world scenario.

In order to overcome this problem, the feedback between the room temperature system and the AHU system is removed by deleting the fields through which the room temperature setpoints are entered. This is represented by a red 'X' mark in Figure 4.7. With the feedback from between the AHU system and rooms removed, the user can provide their

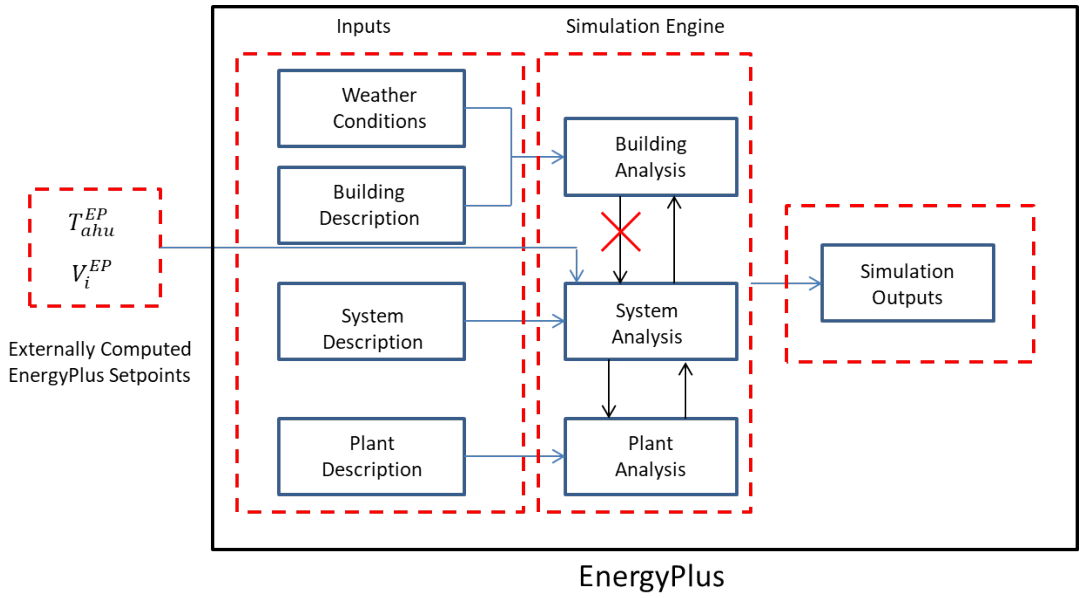


Figure 4.7: Schematic of the EnergyPlus Simulation

own external AHU discharge air temperature setpoint at every simulation time step. The flow rates to the rooms are instead controlled by manipulating the variable that governs the minimum permissible air through each of the VAV boxes. Thus by deleting the room temperature setpoint fields, and altering the minimum flow rate requirements, the EnergyPlus simulations in this chapter are run with $\boldsymbol{V}^{EP}$ and T_{ahu}^{EP} as control inputs. A schematic showing the EnergyPlus inputs, simulation engine, and outputs is shown in Figure 4.7.

4.2 Control Architectures

The previous section described the components of the simulation system, and how the setpoints computed by the control system are translated to simulation outputs. Figure 4.2 shows a schematic of how the setpoints computed in this chapter,$P_{eds}^{set}, T_{ahu}^{set}$, and $\boldsymbol{V}^{set}$ are implemented on the simulation system. This section provides a description of two control methodologies, namely, the existing PI-type control methodology and the proposed MPC methodology are used to compute the aforementioned setpoints.

4.2.1 PI Control Methodology

The existing PI-type control methodology is driven by user defined set points $\boldsymbol{T}^{ref} = [T_1^{ref}, T_2^{ref}, ...T_{n_{sel}}^{ref}]$ which constitute the desired room temperatures. The control methodology then uses the user-defined room temperature setpoints to first compute the flow setpoints $\boldsymbol{V}^{set}$ followed by the AHU end static pressure and discharge air setpoints P_{eds}^{set} and T_{ahu}^{set},,respectively. The algorithm used to compute the aforementioned setpoints is provided below.

The algorithm starts off by calculating a cooling demand $\boldsymbol{L} = [L_1, L_2, ...L_{n_{sel}}]$ for each room. Cooling demand is a numerical scale ranging from 0 to 100 and is computed as an output of a PI controller. The PI controller is driven by the error between the reference and measured room temperatures , i.e. $T_i^{ref} - T_i$ where i is the room under consideration. The cooling demand values are used in the computation of the flow setpoints for each room and the discharge air temperature. A higher cooling demand of a particular room results in higher flow setpoint, and a higher cumulative cooling demand (cooling demand from all the rooms being served by the AHU) results in a lower AHU discharge air temperature setpoint.

The VAV box of each room is associated with a minimum and maximum flow rate, V_i^{min} and V_i^{max}, respectively. A minimum flow rate is maintained in order to meet the fresh air requirements of the building. The maximum flow rate is a limitation placed due to physical constraints resulting from the construction of the VAV boxes, ducts, rated fan power etc. The flow setpoint of the VAV box i is calculated by using a linear relationship with the cooling demand of room i, with upper and lower bounds of V^{max} and V^{min} respectively, being placed on its value as shown in Equation 4.5.

$$f_{flow}(L_i) = V_i^{set} = L_i \cdot \frac{V_i^{max} - V_i^{min}}{100} + V_i^{min} \tag{4.5}$$

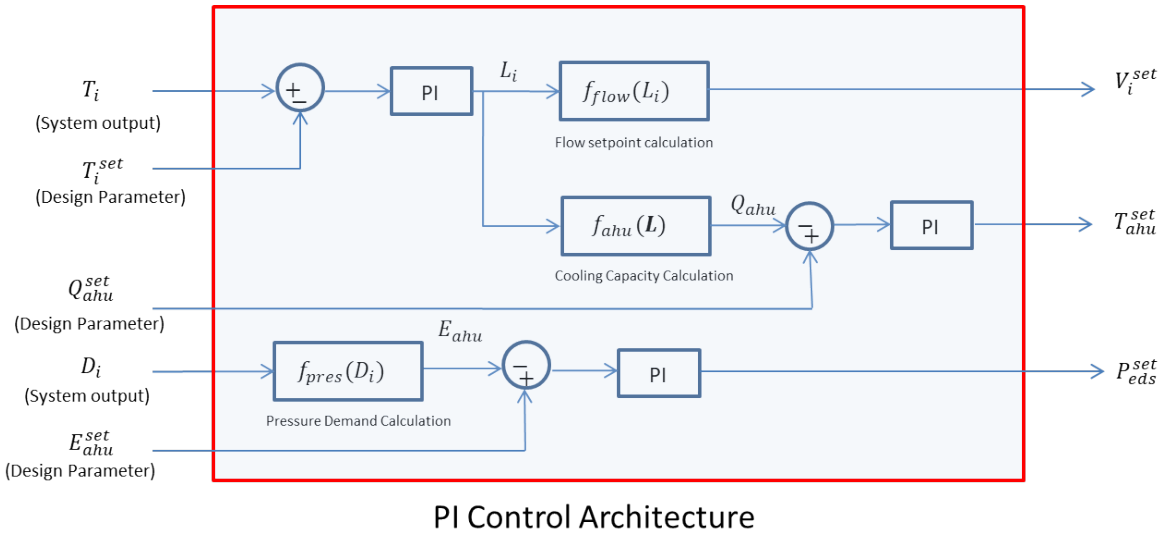


Figure 4.8: Flow Diagram of the Current PI Control Methodology Being Implemented on the Office Building

The flow rate of the conditioned air V_i is actuated to its setpoint value V_i^{set} is used by a local PI controller as discussed in Section 4.1.1. The PI controller is driven by the error between the V_i^{set} and V_i to regulate the damper position D_i. The control algorithm uses the damper position from all the rooms ($\boldsymbol{D}_{vav} = [D_1, D_2, ...D_{n_{sel}}]$) to compute the end static pressure setpoit P_{eds}^{set}.

In order to compute P_{eds}^{set} a pressure demand E_{ahu} is computed by taking the weighted average of the damper position of each room as shown in Equation 4.6.

$$f_{pres}(\boldsymbol{L}) = E_{ahu} = 0.4 \cdot max(\boldsymbol{D}) + 0.4 \cdot \frac{1}{n_{sel}} \sum_{i=1}^{n_{sel}} D_i \tag{4.6}$$

Pressure demand E_{ahu} is a numerical value between 0 and 100 and is a measure of the total air flow requirement of all the rooms. A higher value of E_{ahu} results in a higher P_{eds}^{set} value which in turn results in greater air flow through the VAV boxes. The setpoint P_{eds}^{set} is assigned the value equal to the output signal of a local PI controller which is driven by

the error between a setpoint value for the pressure demand E_{ahu}^{set} and the calculated value E_{ahu}. E_{ahu}^{set} is a design value which is selected by trial and error. For the simulations in this chapter and the real office building, E_{ahu}^{set} is assigned a value of 60. The value of P_{eds}^{set} is constrained between P_{eds}^{min} and P_{eds}^{max} to account for actuator saturation. Figure 4.8 contains a graphical representation of the procedure used to compute P_{eds}^{set}.

The next step in the algorithm is to compute the AHU discharge air temperature setpoint T_{ahu}^{set}. While the cooling demand value of a room is used in the computation of the flow setpoint of the corresponding VAV box, a weighted average of the cooling demand of each room is used in the calculation of the T_{ahu}^{set}. The weighted average of the cooling demand of each room is first used to compute a cooling capacity demand Q_{ahu} using Equation 4.7.

$$f_{ahu}(\boldsymbol{L}) = Q_{ahu} = 0.4 \cdot max(\boldsymbol{L}) + 0.4 \cdot \frac{1}{n_{sel}} \sum_{i=1}^{n_{sel}} L_i \tag{4.7}$$

Cooling capacity demand is also a numerical value between 0 and 100 and is a measure of the amount of cooling that is required to be performed by the AHU heat-exchanger. Greater the cooling capacity demand, lower is the value of $T_{ahu}set$ resulting in a greater cooling requirement. Similar to the method by which P_{eds}^{set} was computed, T_{ahu}^{set} is assigned the value corresponding to the output signal of a local PI controller that is driven by the error between the cooling capacity demand and its reference value Q_{ahu}^{set}. Q_{ahu}^{set} is also a design value and is selected by trial and error. In the real building system and the simulations Q_{ahu}^{ref} is assigned a value 60. The setpoint values are constrained between T_{ahu}^{min} and T_{ahu}^{max} to account for actuator saturation. Figure 4.8 shows a schematic of how the setpoint value T_{ahu}^{set} is computed.

4.2.2 MPC Architecture

The previous section provided a description of how PI controllers were used to compute the setpoints $P_{eds}^{set}, T_{ahu}^{set}$, and $\boldsymbol{V}^{set}$. Instead of using PI controllers, the MPC methodol-

ogy computes the aforementioned setpoints by minimizing an objective function denoted by $\boldsymbol{J}^{mpc}$. Before describing the objective function and the procedure adopted by MPC to compute the setpoints, a definition of the terms and symbols that are used in the description are first provided below.

At each control time instant say t, unlike the PI-type control methodology, MPC does not compute the setpoints just for the current control time step interval, but also for a future time period known as the prediction horizon. If the time step of the simulation is t_s, and the number of time instances for which MPC optimizes the setpoints is n_{pred}, then the prediction horizon of MPC is $t_{pred} = t_s \cdot n_{pred}$. MPC computes optimal setpoints for all the time instances in the prediction horizon $[t, t+1, ..., t+n_{pred}-1]$, but only the setpoints corresponding to the current time instant t are applied to the simulation system or the real building.

The setpoints corresponding to the time instant $t + k$ computed at time instant t are represented by the vector $\boldsymbol{U}^{set}_{t|t+k} = [\boldsymbol{V}^{set}(t|t + k)', P^{set}_{eds}(t|t + k), T^{set}_{ahu}(t|t + k)]'$. The representation $t|t + k$ is used to denote a k-step-ahead prediction made at time instant t. The set of setpoints corresponding to the entire prediction horizon are represented by the vector $\boldsymbol{U}^{set}_{h} = [\boldsymbol{U}^{c}_{t|t}, \boldsymbol{U}^{c}_{t|t+1}, ..., \boldsymbol{U}^{c}_{t|t+n_{pred}}]'$. The subscript h refers to the prediction horizon. If the setpoints over the prediction horizon are computed by MPC then they are represented by the vector $\boldsymbol{U}^{mpc}_{h} = [\boldsymbol{U}^{mpc}_{t|t}, \boldsymbol{U}^{mpc}_{t|t+1}, ..., \boldsymbol{U}^{mpc}_{t|t+n_{pred}}]'$. Figure 4.9 shows a schematic of the inputs and outputs corresponding to the application of MPC at time instant t. The outputs that are measured by the sensors at time instant t are denoted by $\boldsymbol{Y}_t$, and the outputs predicted by the model used for implementing MPC are represented by the vector $\hat{\boldsymbol{Y}}_h = [\boldsymbol{Y}_{t|t+1}, \boldsymbol{Y}_{t|t+2}, ..., \boldsymbol{Y}_{t|t+n_{pred}}]$.

At time instant t MPC computes the setpoints corresponding to the entire prediction

horizon by minimizing an objective function as shown in Equation 4.8.

$$\boldsymbol{U}_h^{mpc} = \underset{\boldsymbol{U}^{set}}{argmin}\ \boldsymbol{J}^{mpc} \tag{4.8}$$

In this thesis, the objective function is designed to be a function of the cost of energy consumption of the HVAC components of the energy distribution level, and a measure of occupancy comfort over the prediction horizon. A more detailed description is provided in the following sections. In order for the MPC solver to perform the minimization expressed in the equation above, a relationship f_{mpc} between the objective function, and the setpoints $\boldsymbol{U}_h^c$ is required as shown in Equation 4.9.

$$\boldsymbol{J}^{mpc} = f_{mpc}(\boldsymbol{U}_h^c) \tag{4.9}$$

Black-box models described in Chapter 3 are used in this thesis to identify statistical relationships between the outputs and inputs of the components of the energy distribution level. These black-box models provide output predictions $\hat{\boldsymbol{Y}}_h$ over the entire prediction horizon, which are used by the MPC solver to perform the objective minimization. Since the models don't accurately predict the outputs of the simulation system or the real building, the predictions made by the black-box models are updated at each time instant. For example at time instant $t+1$, the black-box models receive the simulation/real building outputs $\boldsymbol{Y}_{t+1}$ and the inputs corresponding to time instant t,i.e. $\boldsymbol{u}(t)$ in order to update its states $\boldsymbol{X}(t+1)$. The updated states are then used to make new predictions over the prediction horizon $\hat{Y}_h(t+1)$. A complete description of the inputs and outputs of each black-box subsystem model, and how the states corresponding to the model are updated is provided in the following section. Figure 4.10 shows a schematic of the functions performed by the MPC methodology.

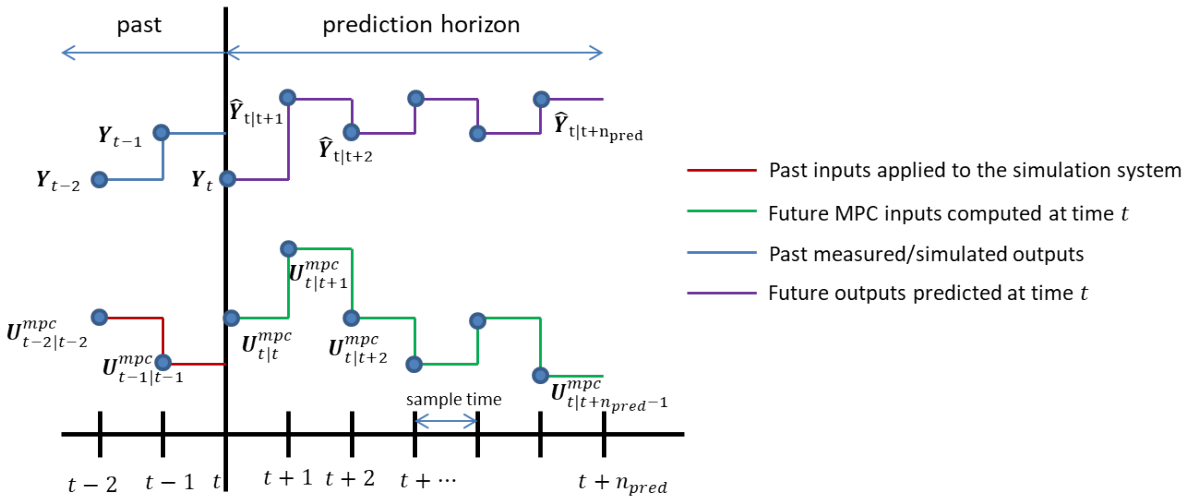


Figure 4.9: Schematic Showing the Measured and Predicted Inputs and Outputs Corresponding to the Application of MPC

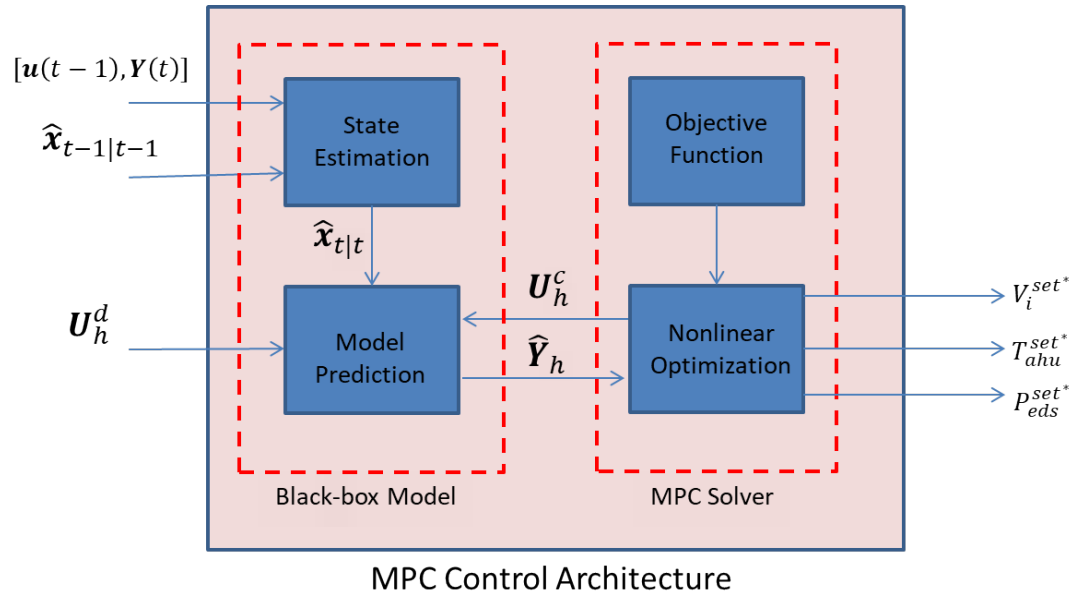


Figure 4.10: Flow Diagram of the Proposed MPC Control Methodology for the Office Building

4.3 Black-Box Models of the Simulated System

The previous section provided a description of how the MPC methodology uses predictions from black-box models to minimize an objective function and compute optimal setpoints over a prediction horizon., MPC solver minimizes the objective function J^{mpc} with the help of a relationship between the cost function and the setpoints f_{mpc} as shown in Equation 4.9. Since there are three main components of the energy distribution level, the mpc objective function is divided into three main components J_{fan}, J_{he}, and J_{comf} corresponding to the AHU volume flow subsystem, AHU heat exchanger subsystem, and the room temperature subsystem, respectively, as shown in Equation 4.10.

$$\boldsymbol{J}^{mpc} = \boldsymbol{J}^{fan} + \boldsymbol{J}^{he} + \boldsymbol{J}^{comf} \tag{4.10}$$

The relationship $fmpc$ can also similarly be divided into three components as shown in Equation 4.11.

$$f_{mpc} = f_{fan} + f_{he} + f_{comf} \tag{4.11}$$

The objective of the black-box models in this thesis is to identify the components of the function f_{mpc} shown above. This section provides a description of the black-box models which are used by the MPC. As was stated earlier.

4.3.1 AHU Volume Flow Black-Box Model

The principal component in the AHU volume flow system that consumes energy is the AHU fan. Thus the objective function corresponding to the AHU volume flow system $\boldsymbol{J}^{fan}$ is chosen to be a function of the energy consumed by the AHU fan over the prediction

horizon as shown in Equation 4.12

$$\boldsymbol{J}^{fan} = r_{fan} \cdot \sum_{i=1}^{n_{pred}} E^{fan}(i) \tag{4.12}$$

where r_{fan} is the energy cost, and $E^{fan}(i)$ is the energy consumed by the fan in the i^{th} time step. The energy consumed by the AHU fan is obtained from the fan law in Equation 4.13

$$E^{fan}(i) = \eta_{fan} \cdot V_{ahu}(i) \cdot P_{eds}(i) \tag{4.13}$$

where η_{fan} is the efficiency of the fan and $V_{ahu}(i) = \sum_{j=1}^{n_{sel}} V_j(i)$ is the volume flow rate across the fan and is the sum of the flow rates across the VAV boxes of all the rooms.

The objective of the black-box model is to estimate the energy consumption of the fan at time instant t over the prediction horizon as a function of the setpoints $\boldsymbol{U}_h^c$. In order to estimate the fan energy, an assumption is made that the volume flow rate across the fan at a future time instant $t+k$ is a scalar multiple of the actual volume as shown in Equation 4.14

$$\hat{V}_{ahu}(t|t+k) = k_{fan} V_{ahu}(t+k) \tag{4.14}$$

where k_{fan} is the scalar constant and $\hat{V}_{ahu}$ is the predicted volume flow rate across the fan. The energy consumed by the fan over the sample time t_k is then predicted by using the Equation 4.15.

$$\hat{E}^{fan}_{t|t+k} = k_{fan} \cdot \eta_{fan} \cdot \hat{V}_{ahu}(t|t+k) \cdot P^{set}_{eds}(t|t+k) \tag{4.15}$$

4.3.2 AHU Heat Exchanger Black-Box Model

In this thesis, the cost of operating the AHU heat exchanger is measured in terms of the amount of chilled water used for cooling. The objective function corresponding to the AHU heat exchanger $\boldsymbol{J}^{he}$ is chosen to be a function of the heat energy gained by CHW

used for cooling the AHU air over the prediction horizon as shown in Equation 4.16

$$\boldsymbol{J}^{he} = r_{chw} \cdot \sum_{i=1}^{n_{pred}} E^{he}(i) = r_{chw} \cdot \sum_{i=1}^{n_{pred}} m_{chw}(i) \cdot (T_{w,out}(i) - T_{w,in}(i) \tag{4.16}$$

where r_{chw} is the cost of CHW operation, m_{chw} is the mass of CHW used in the sample time i, are the outlet and inlet CHW temperatures, respectively.

The CHW energy gained over the prediction horizon is estimated by the black-box model, by using the energy conservation principle, i.e., the heat gained by the CHW system, is equal to the heat lost by the AHU air. In addition, while estimating the CHW energy, an assumption is made that the AHU discharge air setpoint is equal to the measured/simulated temperature. The CHW energy is thus estimated using Equation 4.17

$$\hat{E}^{he} = \sum_{i=1}^{n_{pred}} \rho_a \cdot \hat{V}_{ahu}(i) \cdot (T_{out}(i) - T^{set}_{ahu}(i) \tag{4.17}$$

where ρ_a is the density of air and T_{out} is the outside air temperature.

4.3.3 Room Temperature Subsystem

The third component of the objective function J^{comf} corresponds to the room temperature subsystem. As stated earlier, the minimization of the objective function requires a relationship between the objective function and the setpoints over the prediction horizon as shown in Equation 4.18.

$$\boldsymbol{J}^{comf} = f^{comf}(\boldsymbol{U}^c_h) \tag{4.18}$$

The comfort of the occupants of the room, however, is measured as a function of the temperature of the rooms. Hence in order to express the comfort cost J^{comf} in terms of the setpoints $\boldsymbol{U}^c_h$, a relationship of the setpoints with the temperature of the rooms $\boldsymbol{T} = [T_1, T2, ..., T_{n_{sel}}]$ is first required. The objective of the black-box models is thus to

Table 4.1: Inputs to the Room Temperature Subsystems

Inputs	Output
T_{oa}	
$-V_i * (T_i - T_{ahu})$	T_i
T_j, where $j \in R_i^{sur}$	
h_{oa}	

identify the relationship as shown in Equation 4.19

$$\boldsymbol{T}_h = f^{room}(\boldsymbol{U}_h^c) \tag{4.19}$$

where the subscript h in $\boldsymbol{T}_h$ is used to denote the temperature of the rooms over the prediction horizon. The relationship between the room temperatures and the comfort cost is described in the following section.

In order to identify the relationship f^{room}, black-box models of the room temperature dynamics of each room are identified using the algorithm proposed in Chapter 3. The relationship is expressed in terms of the state-space equation as shown in

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_i u_i(k) + K_i e(k) \\ T_i(k) &= C_i x_i(k) \end{aligned} \tag{4.20}$$

where x_i are the states, A_i, B_i, C_i, K_i are the state space matrices, and u_i are the inputs to the room temperature subsystem i. The list of possible inputs from which u_i is obtained are reproduced in Table 4.1 for ease of reference. In the table T_{oa} and h_{oa} refer to the outside air temperature and humidity, respectively, and R_i^{sur} is the list of surrounding rooms. A description of inputs, outputs and how the state space matrices are identified are provided in Chapter 3. A comprehensive state space equation is then created by combining the state

space matrices of each room as shown in Equation 4.21

$$\begin{aligned} x(k+1) &= Ax_(k) + Bu(k) + Ke(k) \\ \boldsymbol{T}(k) &= Cx(k) \end{aligned} \tag{4.21}$$

where x refers to the states and A, B, C, K are the state space matrices of the comprehensive model. The state space matrices A, B, C, K are built from the subsystem state space models using Equation 4.22

$$A = \begin{bmatrix} A_1 + \sum\limits_{j \in R_1^{sel}} B_1^j C_j & 0 & 0 & \dots & 0 \\ 0 & A_2 + \sum\limits_{j \in R_2^{sel}} B_2^j C_j & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \dots & \vdots \\ 0 & 0 & 0 & \dots & A_2 + \sum\limits_{j \in R_{n_{sel}}^{sel}} B_{n_{sel}}^j C_j \end{bmatrix}$$

$$B = \begin{bmatrix} B_1^{oa} & B_1^c & 0 & \dots & B_1^{h_{oa}} \\ B_2^{oa} & 0 & B_2^c & \dots & B_2^{h_{oa}} \\ \vdots & 0 & 0 & \ddots & \vdots \\ B_{n_{sel}}^{oa} & 0 & \dots & B_{n_{sel}}^c & B_{n_{sel}}^{h_{oa}} \end{bmatrix} \; C = diag(C_i) \; ; K = diag(K_i); i \in \{1, 2, ...n_{sel}\}$$

(4.22)

where R_i^{sel} is the set of surrounding rooms which are inputs to room i, B_i^{oa} is the column matrix of B_i corresponding to the input T_{oa}, B_i^c is the column matrix corresponding to the control input $u_i^{ctr} = -V_i \cdot (T_i - T_{ahu})$, and $B_i^{h_{oa}}$ is the column matrix corresponding to the input h_{oa}.

Since the black-box model does not accurately predict the simulation/real building outputs, the states of the centralized state space system are updated at each time instant

based on the most recent measured outputs and inputs as shown in Equation 4.23.

$$\begin{aligned} x(k) &= Ax_(k-1) + Bu(k-1) \\ x(k) &= x(k) + K(\boldsymbol{T}(k) - Cx(k) \end{aligned} \tag{4.23}$$

After updating the states, predictions of room temperatures over the prediction horizon in terms of the future disturbance inputs, and control setpoints can be made over the prediction horizon by using Equation 4.24 . While making the predictions an assumption is made that the VAV flows, and AHU discharge air temperature are equal to the setpoint values.

$$\begin{aligned} &x(k+1) = Ax(k) + Bu(k) \\ &\hat{\boldsymbol{T}}_{k|k+1} = Cx(k+1) \\ &x(k+2) = Ax(k+1) + Bu(k+1) \\ &\hat{\boldsymbol{T}}_{k|k+2} = Cx(k+2) \\ &\vdots \\ &x(k+n_{pred}) = Ax(k+n_{pred}-1) + Bu(k+n_{pred}-1) \\ &\hat{\boldsymbol{T}}_{k|k+n_{pred}} = Cx(k+n_{pred}) \end{aligned} \tag{4.24}$$

4.4 Implementing Centralized MPC

The previous section provided a description of how black-box models of HVAC systems provide predictions that can be used by MPC. This section provides a description of how the MPC solver shown in Figure 4.10 computes the optimal variables over the prediction horizon $\boldsymbol{U}_h^{mpc}$ by making use of the predictions from the black-box models. The section is divided into 4 parts. The first part contains a description of the optimization variables and the values within which they are constrained. The second part contains a description of the objective function being minimized. The third part provides a method to

compute the gradient and hessian of the objective function which is used by the optimizer to perform the minimization.

4.4.1 Optimization Variables and Constraints

The variables optimized by MPC correspond to the setpoints of the three subsystems of the energy distribution level over the entire prediction horizon t_{pred}. The variables are represented by the vector $\boldsymbol{U}_h^{mpc}$ as shown in Equation 4.25.

$$\begin{aligned}\boldsymbol{U}_h^{mpc} =&[V_1^{set}(k|k), V_2^{set}(k|k), \ldots, V_{n_{sel}}^{set}(k|k), \ldots, V_{n_{sel}}^{set}(k|k+n_{pred}-1),\\ &T_{ahu}^{set}(k|k), P_{eds}^{set}(k|k), \ldots, P_{eds}^{set}(k|k+n_{pred}-1)]'\end{aligned} \tag{4.25}$$

In order to reduce the computational burden, T_{ahu}^{set} is kept constant through out the prediction horizon. The number of optimization variables n_{var} are thus given by Equation 3.34.

$$n_{var} = n_{sel} \cdot n_{pred} + 1 + n_{pred} \tag{4.26}$$

The physical and actuator limitations subject the optimization variables ($\boldsymbol{U}_h^{mpc}$) to the nonlinear inequality constraints. For example the volume flow through the VAV boxes of each room is required to be greater than a certain minimum flow V^{min} in order to provide fresh air to the occupants. The maximum air flow through the VAV box is constrained by the physical limits of the VAV box, and the pressure differential P_{eds}. The relationship expressed in Equation 4.1 is used to set the upper bound on the volume flow of the VAV boxes V^{max}. The constraint inequality for the volume flow setpoints is expressed in 4.27.

$$\begin{aligned}V_i^{min} < V_i^{set}(t|t+k) < V_i^{max} = (a+b)\sqrt{P_{eds}^{set}(t|t+k)}\\ \forall t,\ i \in \{1,2,...,n_{sel}\},\ k \in \{1,2,...,n_{pred}\}\end{aligned} \tag{4.27}$$

Due to the physical constraints of the chiller providing CHW, and the AHU fan, the set-

points for the AHU discharge temperature, and the AHU end static pressure are constrained between upper and lower bounds T_{ahu}^{min} and P_{eds}^{min}, and T_{ahu}^{max} and P_{eds}^{max}, respectively, as shown in the inequality 4.28.

$$\begin{aligned} & T_{ahu}^{min} < T_{ahu}^{set} < T_{ahu}^{max} \\ & P_{eds}^{min} < P_{eds}^{set}(t|t+k) < P_{eds}^{max},\ \forall t,\ k \in \{1, 2, ..., n_{pred}\} \end{aligned} \tag{4.28}$$

4.4.2 MPC Objective Function

The main objective of the MPC solver is to compute setpoints that minimize energy consumption without compromising occupant comfort. To serve this end, a cost function is designed which penalizes the energy consumption of the AHU fan, the amount of CHW used by the heat exchanger, and departures from thermal comfort. The three components of the cost function shown in Equation 4.10 are computed by using Equations 4.12, 4.16, and 4.18. The aforementioned equations however depend on measurements made by the sensors during simulation or operation. The optimal setpoints, however, are obtained minimizing the objective function over a future time period. Hence, MPC solver uses predictions of the cost function components using the black-box models described in the previous section. A description of the predictions of the cost function components are provided below.

4.4.2.1 AHU Volume Flow Predicted Cost

A k-time-step ahead prediction of the AHU fan energy made at time instant t is expressed in Equation 4.15. The component of the cost function corresponding to the AHU volume flow simulation $\hat{\boldsymbol{J}}^{fan}$ is over the prediction horizon is estimated using the predicted

fan energy as shown in Equation 4.29

$$\hat{\boldsymbol{J}}^{fan} = r_{fan} \cdot \sum_{k=1}^{n_{pred}} \hat{E}^{fan}_{t|t+k} \tag{4.29}$$

In this thesis, the cost of energy r_fan is varied for the different simulation test cases discussed in the following section.

4.4.2.2 AHU Heat Exchanger Predicted Cost

The cost of operation of the AHU heat exchanger is measured in terms of the CHW used for cooling the outside air. A k-time-step ahead prediction of the CHW energy made at time instant t is expressed in Equation 4.17. The cost function estimate corresponding to the AHU heat exchanger over the prediction horizon is computed using the estimated CHW energy as shown in Equation 4.30.

$$\hat{\boldsymbol{J}}^{he} = r_{chw} \cdot \sum_{k=1}^{n_{pred}} \hat{E}^{he}_{t|t+k} \tag{4.30}$$

Similar to the fan energy cost, the CHW energy cost r_{chw} is varied for the different simulation test cases.

4.4.2.3 Comfort Cost

Whereby the cost components corresponding to the AHU fan and heat exchanger can be objectively measured in terms of the energy consumption and mass of CHW used, the thermal comfort cost is a subjective assessment of occupancy comfort. One way to measure comfort is to estimate the impact the environmental conditions may have on the productivity of the occupant. The environmental conditions that impact a person's productivity may entail several factors such as the task being performed, clothing, metabolic rate, wind speed, temperature, humidity etc. In order to make the computation of the thermal

cost easier, however, the aforementioned factors with the exception of temperature and humidity are assumed to be constant. A relationship between temperature and humidity with a person's productivity is then sought.

A summary of the effects of thermal comfort on productivity in a laboratory environment was provided by Wyon in [58] . The laboratory tasks were divided into two types, namely, thinking and typing. The results of the survey showed that for thinking tasks the productivity reduced by dropped by 30% at 27^0C for a neutral temperature of 21^0C, and for typing tasks the same loss of productivity occurred at 25^0C. As stated earlier, however, the loss in productivity does not just depend on temperature but a host of other factors.

The temperatures in the results summarized by Wyon were converted into a thermal scale called Predicted Mean Vote (PMV) by Kosonen in [59]. PMV is a scale developed by Fagner [60] that takes into account all the aforementioned factors that impact occupant comfort and represents them by a single scalar value. The scale ranges from -3 to 3, with a negative value indicating cold sensations and a positive value indicating hot sensations. A regression analysis was then performed by Kosonen to express the loss of productivity values in terms of PMV for thinking and typing tasks. The results of the regression analysis for thinking tasks is summarized in Equation

$$\begin{aligned} r_i(t|t+k) =& \beta_0 + \beta_1 \hat{p}_i(t|t+k) + \beta_2 \hat{p}_i(t|t+k) + \beta_3 \hat{p}_i^2(t|t+k) + \\ & \beta_4 \hat{p}_i(t|t+k) \hat{p}_i(t|t+k) + \beta_5 \hat{p}_i^2(t|t+k) \end{aligned} \tag{4.31}$$

where $\hat{p}_i(t|t+k)$ is the k-step-ahead predicted PMV of room i, r is the productivity loss ration, and β_j $j \in \{0, 1, .., 5\}$ are the regression coefficients whose values are summarized in Table 4.2.

Computation of the productivity loss ratio in the above equation requires a prediction of PMV of the rooms over the predition horizon. PMV is a highly nonlinear function of

Table 4.2: Parameter Values of the PMV and Productivity Functions Obtained through Regression Analysis

Parameter	Value	Parameter	Value
α_0	-5.83	α_5	0
α_1	-0.20	β_0	
α_2	-0.005	β_0	
α_3	0.001	β_0	
α_4	0.0005		

several factors, but for the purposes of simulations in this thesis, PMV is expressed in terms of only temperature and humidity, while keeping all the other parameters constant. PMV corresponding to temperature values between 15^0C and 35^C, and humidity values between 0 and 100 were computed using the procedure described by Fagner. A regression analysis was then performed on the resulting data to obtain the quadratic relationship shown in Equation 4.32.

$$\begin{aligned} p_i(t|t+k) =& \alpha_0 + \alpha_1 \hat{T}_i(t|t+k) + \alpha_2 \hat{h}_i(t|t+k) + \alpha_3 \hat{T}_i^2(t|t+k) + \\ & \alpha_4 \hat{T}_i(t|t+k)\hat{h}_i(t|t+k) + \alpha_5 \hat{h}_i^2(t|t+k) \end{aligned} \tag{4.32}$$

where $\hat{h}_i$ is the predicted relative humidity of room i, and α_j $j \in \{0, 1, .., 5\}$ are parameters obtained through regression analysis whose values are summarized in Table 4.2. Since the relative humidity of the rooms were not modeled, the predictions $\hat{h}_i(t|t+k)$ $\forall k \in \{1, 2, ..., n_{pred}\}$ were assigned the last measured value $h_i(t)$. Rooms temperatures over the prediction horizon are computed using Equation 4.24.

4.4.3 Solution to the MPC Objective Function

Optimal values of the setpoints over the prediction horizon $\boldsymbol{U}_h^{mpc}$ are obtained by minimizing the objective function $\hat{\boldsymbol{J}}^{mpc}$ subject to the constraints 4.28 and 4.27 as show in

Equation 4.33

$$\boldsymbol{U}_h^{mpc} = \underset{\boldsymbol{U}_h^c}{argmin}\ \hat{\boldsymbol{J}}^{mpc} \tag{4.33}$$

where $\hat{\boldsymbol{J}}^{mpc}$ is the predicted cost of operation. $\hat{\boldsymbol{J}}^{mpc}$ is computed using Equation 4.34

$$\hat{\boldsymbol{J}}^{mpc} = \hat{\boldsymbol{J}}^{fan} + \hat{\boldsymbol{J}}^{he} + \hat{\boldsymbol{J}}^{comf} \tag{4.34}$$

The optimization problem is nonlinear and non convex. The interior-point method algorithm is used to minimize the constrained cost function. Interior point algorithm uses the gradient, and hessian of the cost function and the constraints in order to find the local minima. The numerical optimization solvers in MATLAB perform much faster when an analytical expression of the gradient and hessian is provided. A brief description of how the gradient, and hessian were computed is provided below.

4.4.3.1 Gradient of Objective Function

The gradient of the objective function is the sum of the gradients of the predicted fan cost ($\nabla\hat{\boldsymbol{J}}^{fan}$), gradient of predicted heat exchanger cost ($\nabla\hat{\boldsymbol{J}}_{he}$), and gradient of predicted comfort cost ($\nabla\hat{\boldsymbol{J}}_{co}$) as shown in Equation.

$$\nabla\hat{\boldsymbol{J}}^{mpc} = \nabla\hat{\boldsymbol{J}}^{fan} + \nabla\hat{\boldsymbol{J}}^{he} + \nabla\hat{\boldsymbol{J}}^{comf} \tag{4.35}$$

The gradient of each cost component can be further divided into gradient w.r.t flow setpoints $\boldsymbol{V}^{\boldsymbol{set}}$, the AHU discharge air temperature setpoint T_{ahu}^{set}, and the end static pressure setpoints over the prediction horizon $\boldsymbol{P}_{\boldsymbol{eds}}^{\boldsymbol{set}}$ as shown in Equation 4.36.

$$\begin{aligned}
\nabla\hat{\boldsymbol{J}}^{fan} &= \left[\frac{\partial\hat{\boldsymbol{J}}^{fan}}{\partial\boldsymbol{V}^{set}}, \frac{\partial\hat{\boldsymbol{J}}^{fan}}{\partial T_{ahu}^{set}}, \frac{\partial\hat{\boldsymbol{J}}^{fan}}{\partial\boldsymbol{P}_{\boldsymbol{eds}}^{\boldsymbol{set}}}\right]' \quad \nabla\hat{\boldsymbol{J}}^{he} = \left[\frac{\partial\hat{\boldsymbol{J}}^{he}}{\partial\boldsymbol{V}^{set}}, \frac{\partial\hat{\boldsymbol{J}}^{he}}{\partial T_{ahu}^{set}}, \frac{\partial\hat{\boldsymbol{J}}^{he}}{\partial\boldsymbol{P}_{\boldsymbol{eds}}^{\boldsymbol{set}}}\right]' \\
\nabla\hat{\boldsymbol{J}}^{co} &= \left[\frac{\partial\hat{\boldsymbol{J}}^{co}}{\partial\boldsymbol{V}^{set}}, \frac{\partial\hat{\boldsymbol{J}}^{co}}{\partial T_{ahu}^{set}}, \frac{\partial\hat{\boldsymbol{J}}^{co}}{\partial\boldsymbol{P}_{\boldsymbol{eds}}^{\boldsymbol{set}}}\right]'
\end{aligned} \tag{4.36}$$

$$
\begin{aligned}
\frac{\partial J_{cl}}{\partial V_m^{*t|j}} &= \eta_{he} \cdot r_{cl} \cdot t_s \cdot \rho_a \cdot T_{oa}^{t|j},\ j \in \{0, 1, ..., n_{pred}\} \\
\frac{\partial J_{cl}}{\partial T_{ahu}^{*t|0}} &= -\eta_{he} \cdot r_{cl} \cdot t_s \cdot \rho_a \cdot \sum_{j=1}^{n_{pred}} V_{ahu}^{t|j-1},\ j \in \{0, 1, ..., n_{pred}\} \\
\frac{\partial J_{cl}}{\partial P_{eds}^{*t|j}} &= 0,\ j \in \{0, 1, ..., n_{pred}\}
\end{aligned} \tag{4.37}
$$

$$
\begin{aligned}
\frac{\partial J_f}{\partial V_m^{*t|j}} &= r_f \cdot \frac{t_s}{3.6 \cdot 10^6} \cdot P_{eds}^{t|j},\ j \in \{0, 1, ..., n_{pred}\} \\
\frac{\partial J_f}{\partial T_{ahu}^{*t|0}} &= 0 \\
\frac{\partial J_f}{\partial P_{eds}^{*t|j}} &= r_f \cdot \frac{t_s}{3.6 \cdot 10^6} \cdot V_{ahu}^{t|j},\ j \in \{0, 1, ..., n_{pred}\}
\end{aligned} \tag{4.38}
$$

The gradient of the comfort cost can be obtained by first computing the gradient of the predicted room temperatures, which is derived in an iterative manner shown below.

- *Step 1:* Compute gradients at current timestep t using Equation 4.39.

$$
\begin{aligned}
\frac{\partial u_l^{ctr(t|0)}}{\partial V_l^{*t|0}} &= -(T_l^{t|0} - T_{ahu}^{*t|0});\ l \in \{1, 2, ...n_{sel}\} \\
\frac{\partial x^{t|1}}{\partial V_l^{*t|0}} &= B \frac{\partial \boldsymbol{u}_{ctr}^{t|0}}{\partial V_l^{*t|0}} \\
\frac{\partial \hat{T}^{t|1}}{\partial V_l^{*t|0}} &= C \frac{\partial x^{t|1}}{\partial V_l^{*t|0}}
\end{aligned} \tag{4.39}
$$

- *Step 2:* Using 4 nested for loops, with $i = 1 : n_{pred} - 1$, $j = 1 : i$, $l = 1 : n_{sel}$, and $m = 1 : n_{sel}$, iteratively compute the gradients using Equations 4.40, and 4.41.

$$\frac{\partial u_l^{ctr(t|i)}}{\partial V_m^{*t|j}} = \begin{cases} -(\hat{T}_l^{t|i} - T_{ahu}^{*t|0})\ if\ l = m\ \&\ i = j \\ else \\ -V_l^{*t|i} \dfrac{\partial \hat{T}_l^{t|i}}{\partial V_m^{*t|j}} \end{cases} \tag{4.40}$$

$$\begin{aligned} \frac{\partial x^{t|i+1}}{\partial V_m^{*t|j}} &= A\frac{\partial x^{t|i}}{\partial V_m^{*t|j}} + B\frac{\partial \boldsymbol{u}_{ctr}^{t|i}}{\partial V_m^{*t|j}} \\ \frac{\partial \hat{T}^{t|i+1}}{\partial V_m^{*t|j}} &= C\frac{\partial x^{t|i+1}}{\partial V_m^{*t|j}} \end{aligned} \tag{4.41}$$

- *Step 3:* A similar iterative procedure is adopted to find the gradient of the predicted temperatures w.r.t $T_{ahu}^{*}t|0$

  .

- *Step 4:* Equation 4.42 is used to compute the gradient w.r.t the comfort cost.

$$\begin{aligned} \frac{\partial J_{co}}{\partial V_m^{*t|j}} &= \sum_{i=1}^{n_{pred}} \sum_{l=1}^{n_{sel}} \frac{\partial J_{co}}{\partial pmv_l^{*t|i}} \cdot \frac{\partial pmv_l^{*t|i}}{\partial T_l^{*t|i}} \cdot \frac{\partial T_l^{*t|i}}{\partial V_m^{*t|j}} \\ \frac{\partial J_{co}}{\partial T_{ahu}^{*t|0}} &= \sum_{i=1}^{n_{pred}} \sum_{l=1}^{n_{sel}} \frac{\partial J_{co}}{\partial pmv_l^{*t|i}} \cdot \frac{\partial pmv_l^{*t|i}}{\partial T_l^{*t|i}} \cdot \frac{\partial T_l^{*t|i}}{\partial T_{ahu}^{*t|0}} \\ \frac{\partial J_{co}}{\partial P_{eds}^{*t|j}} &= 0 \end{aligned} \tag{4.42}$$

4.4.3.2 Hessian of Objective Function

The hessian of the cost function is the sum of the hessian of J_{co}, J_f, and J_c. The only non zero component of the hessian of the cooling cost is w.r.t the variables involving volume flow rate and the AHU discahrge air temperature computed as shown in Equa-

tion 4.43. Similarly the only nonzero component of the hessian of the fan cost is w.r.t the variables involving volume flow setpoint, and end static pressure shown in Equation 4.44.

$$\frac{\partial^2 J_c}{\partial V_l^{*t|i} \partial T_{ahu}^{*t|0}} = -r_c \tag{4.43}$$

$$\frac{\partial^2 J_f}{\partial V_l^{*t|i} \partial T_{ahu}^{*t|0}} = r_f \tag{4.44}$$

The Hessian of the comfort cost w.r.t two flow setpoint variables is computed iteratively using the following algorithm. A similar algorithm can be used to find the hessian involving $T_{ahu}^{*t|0}$. The hessian of the comfort cost involving P_{eds}^{*} is 0.

- *Step 1:* Using 6 nested for loops, with $i = 1 : n_{pred} - 1$, $j = 1 : i$,$k = 1 : j$, $l = 1 : n_{sel}$, $m = 1 : n_{sel}$, and $n = 1 : n_{sel}$, iteratively compute the Hessians using Equations 4.45, and 4.46.

$$\frac{\partial^2 u_l^{ctr(t|i)}}{\partial V_m^{*t|j} \partial V_n^{*t|k}} = \begin{cases} -\dfrac{\partial \hat{T}_l^{t|i}}{\partial V_n^{*t|k}} \; if \; l = m \; \& \; i = j \\ else \\ -V_l^{*t|i} \dfrac{\partial^2 \hat{T}_l^{t|i}}{\partial V_m^{*t|j} \cdot \partial V_n^{*t|k}} \end{cases} \tag{4.45}$$

$$\begin{aligned} \frac{\partial^2 x^{t|i+1}}{\partial V_m^{*t|j} \partial V_n^{*t|k}} &= A \frac{\partial^2 x^{t|i}}{\partial V_m^{*t|j} \cdot \partial V_n^{*t|k}} + B \frac{\partial^2 \boldsymbol{u}_{ctr}^{t|i}}{\partial V_m^{*t|j} \cdot \partial V_n^{*t|k}} \\ \frac{\partial^2 \hat{T}^{t|i+1}}{\partial V_m^{*t|j} \cdot \partial V_n^{*t|k}} &= C \frac{\partial^2 x^{t|i+1}}{\partial V_m^{*t|j} \cdot \partial V_n^{*t|k}} \end{aligned} \tag{4.46}$$

- *Step 3:* The hessian of the comfort cost w.r.t two flow setpoint variables is given by Equation 4.47.

$$\frac{\partial^2 J_{co}}{\partial V_m^{*t|j} \partial V_n^{*t|k}} = \sum_{i=1}^{n_{pred}} \sum_{l=1}^{n_{sel}} \frac{\partial^2 J_{co}}{\partial (pmv_l^{*t|i})^2} \cdot \left(\frac{\partial (pmv_l^{*t|i})}{\partial \hat{T}_l^{t|i}} \right)^2 \cdot \frac{\partial \hat{T}_l^{t|i}}{\partial V_m^{*t|j}} \cdot \frac{\partial \hat{T}_l^{t|i}}{\partial V_n^{*t|k}} + \frac{\partial J_{co}}{\partial pmv_l^{*t|i}} \cdot \frac{\partial (pmv_l^{*t|i})}{\partial \hat{T}_l^{t|i}} \cdot \frac{\partial^2 \hat{T}_l^{t|i}}{\partial V_m^{*t|j} \partial V_n^{*t|k}} \tag{4.47}$$

4.5 MPC Operation

In order to compare the performance of centralized MPC and the traditional PI control strategy several simulations are run with varying operating conditions. The simulations can be categorized into two sets depending on the parameter that is varied to enforce the changing operating conditions. The first set of simulations correspond to operating conditions that are varied by changing the cost parameters r_{fan}, r_{he}, and r_{sal}^{yr}. The second set of simulations correspond to operating conditions that are varied by changing the outside air temperature and humidity. For each operating conditions, three simulations are run corresponding to centralized MPC, PI control with setpoint corresponding to the upper bound for comfort, and PI control with temperature setpoints corresponding to the lower bound for comfort. The results of the comparison are provided below.

4.5.1 Changing Cost Parameters

The cost parameters of the objective function are varied to generate three operating conditions to simulate high cooling costs, high fan costs, and real-world estimation of costs. The cost parameters corresponding to each operationg condition are shown in Ta-

ble 4.3.

Table 4.3: Cost Parameters

Operation	r_{clng}	r_{fan}	r_{sal}^{yr}
High Cooling Cost	$1.101e-6$	$1.1078e-5$	200
High Fan Cost	$1.101e-4$	$1.1078e-7$	200
Real World Cost	$1.101e-6$	$1.1078e-7$	$20,000$

The first operating condition corresponds to high cooling costs. The result of applying centralized MPC with high cooling cost is shown with the help of Figure 4.11 which shows the simulated temperature of room 1. For comparison purposes, the simulated temperature of room 1 is shown with two PI type controls (high temperature setpoint PI, low temperature setpoint PI) applied to the simulation system. The figure also shows the simulated temperature of room 1 when no control is applied. The end static pressure setpoint P_{ed}, and the volume flow rates through each room $\boldsymbol{V}^{set}$ are set to the lowest permissible values, and the discharge air temperature T_{ahu}^{set} is set to the highest permissible value. The end static pressure setpoint and discharge air temperature setpoint computed by the three controllers are shown in Figure The same process is repeated with high fan cost and real world cost parameters. The results corresponding to the two operating conditions are shown in Figures. A summary of the cost achieved by the controllers for each operating condition is summarized in Table . The results show that for each type of controller, MPC outperforms the PI type controllers.

As is evident from the above results MPC performs component level optimization by seeking to find the combination of flow rates, end static pressure and discharge air temperature setpoints that would result in the least value of the objective function. Furthermore the summary of the results in Table 4.4 show that MPC performs better than the PI type

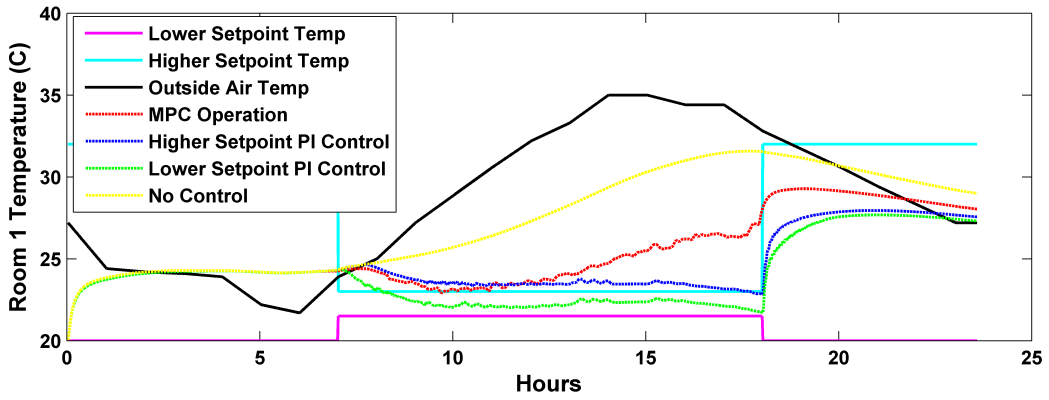


Figure 4.11: Simulated Temperature of Room 1 Under High Cooling Cost Conditions with Different Control Methodologies

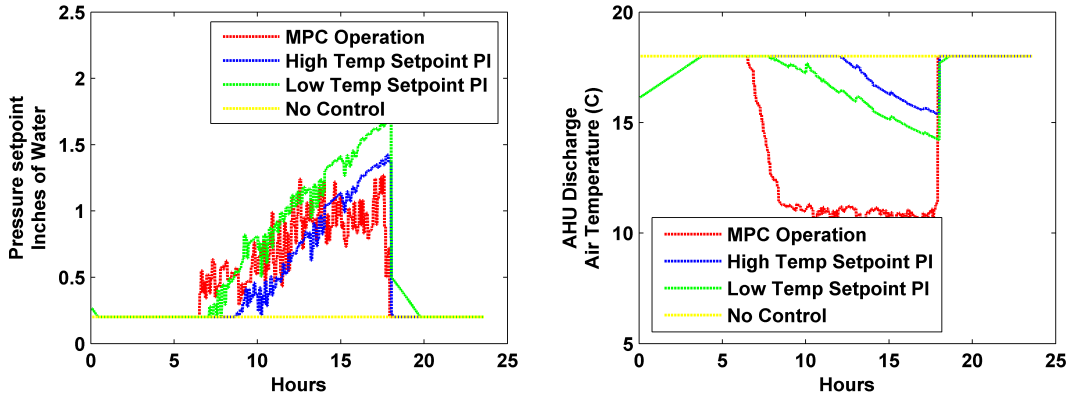


Figure 4.12: Simulated End Static Pressure and Discharge Air Temperature Under High Cooling Cost Conditions with Different Control Methodologies

controllers under a wide range of conditions.

4.5.2 Changing Outside Air Conditions

In order to compare the performance of MPC with PI-type controllers under varying weather conditions, three sets of simulations are run. The operating condition corresponding to each set of simulation is summarized in Table. The operating conditions are varied by varying the day of the simulation. The first condition corresponds to the weather con-

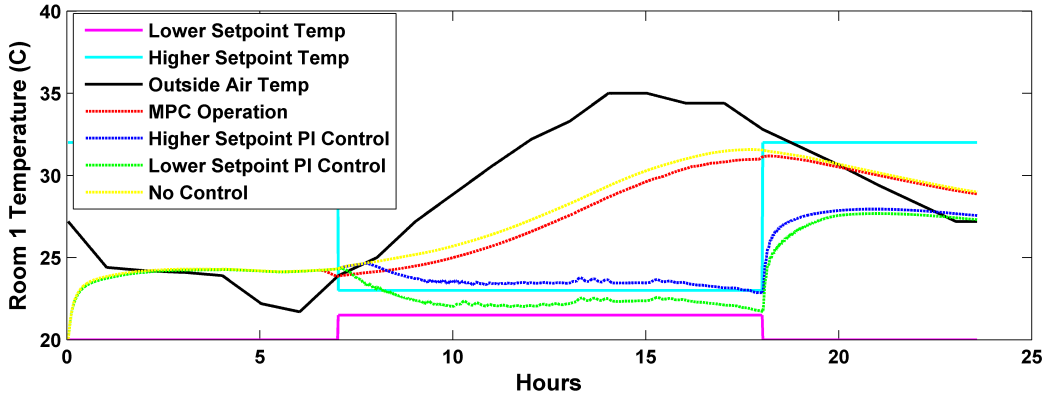


Figure 4.13: Simulated Temperature of Room 1 Under High Fan Cost Conditions with Different Control Methodologies

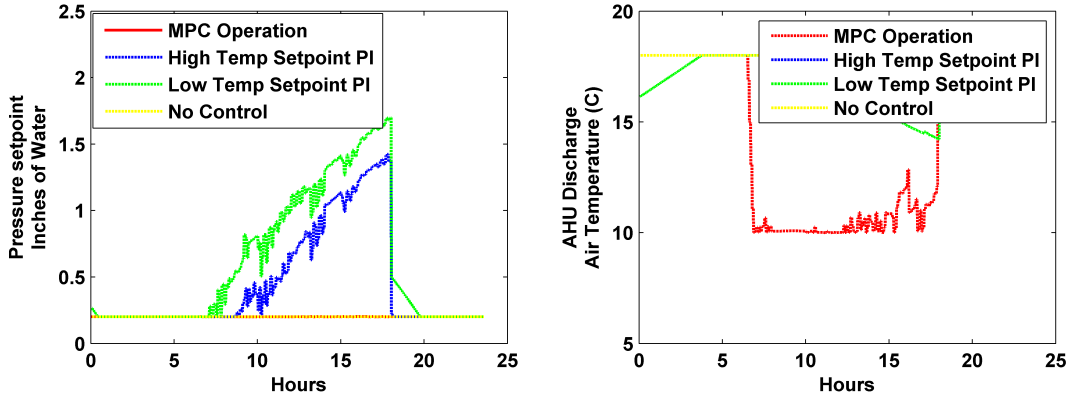


Figure 4.14: Simulated End Static Pressure and Discharge Air Temperature Under High Fan Cost Conditions with Different Control Methodologies

ditions on April 1st of the EnergyPlus weather file of College Station, TX. The second and third operating conditions correspond to the days June 1st, and August 1st respectively. The results of the comparison are summarized in Table

4.5.3 Effect of Prediction Horizon

In this section, the effect of the prediction horizon on the performance of MPC is studied. Three simulations are run corresponding to a prediction horizon of 6, 12, and 18

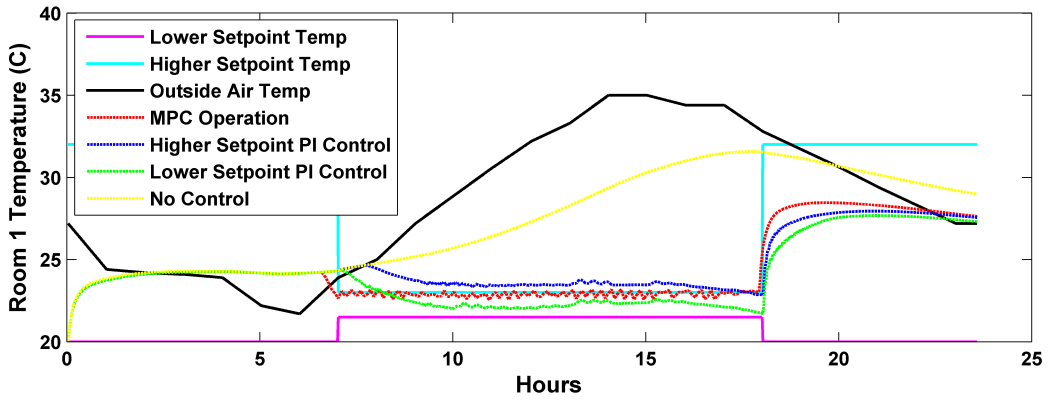


Figure 4.15: Simulated Temperature of Room 1 Under Real-World Cost Conditions with Different Control Methodologies

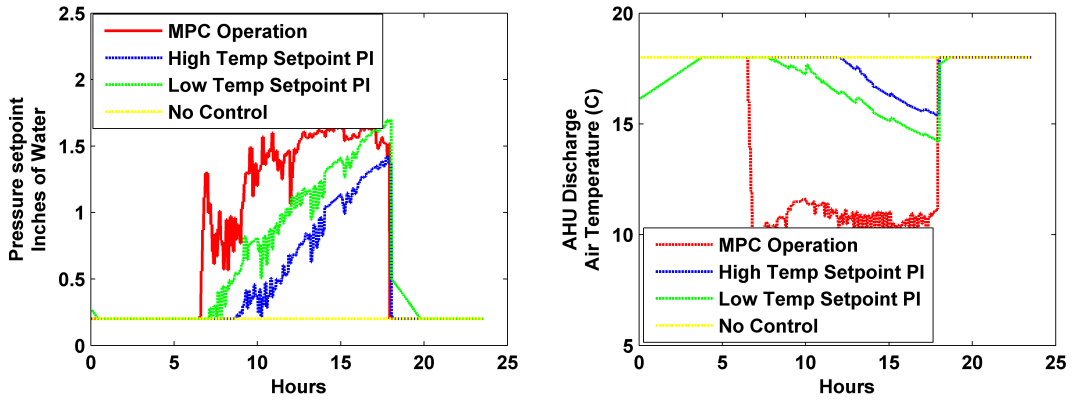


Figure 4.16: Simulated End Static Pressure and Discharge Air Temperature Under Real-World Cost Conditions with Different Control Methodologies

time steps, respectively. In units of time, the prediction horizon of the three simulations steps are 0.5 hours, 1 hour, and 1.5 hours, respectively. A comparison of the cost achieved with real world cost parameters on June 1st is shown in Table.

Simulations are run with estimates real world objective function parameters, and outside air conditions corresponding to June 1st of the EnergyPlus weather file of College Station. Table 4.6 shows the summary of the costs achieved by applying MPC with vary-

Table 4.4: Summary of the Costs Achieved with Varying Objective Function Parameter Values

Real World Cost Conditions			
Cost/Control	PI high setpoint	PI low setpoint	MPC
Comfort Cost ($)	103.18	24.44	1.33
AHU fan and HE cost ($)	0.44	0.68	0.46
Total Cost($)	103.63	25.12	1.79
High Cooling Cost Conditions			
Comfort Cost ($)	1.03	0.24	0.84
AHU fan and HE cost ($)	7.16	9.23	4.14
Total Cost($)	9.47	8.19	4.98
High Fan Cost Conditions			
Comfort Cost ($)	1.03	0.24	2.49
AHU fan and HE cost ($)	37.89	59.92	1.77
Total Cost($)	38.92	60.17	4.26

Table 4.5: Summary of the Costs Achieved Under Varying Outside Air Conditions

April 1st			
Cost/Control	PI high setpoint	PI low setpoint	MPC
Comfort Cost ($)	38.6	16.15	15.77
AHU fan and HE cost ($)	0.03	0.08	0.11
Total Cost($)	38.63	16.23	15.88
June 1st			
Comfort Cost ($)	103.18	24.44	1.33
AHU fan and HE cost ($)	045	0.68	0.46
Total Cost($)	103.63	25.13	1.79
August 1st			
Comfort Cost ($)	96.33	22.94	1.40
AHU fan and HE cost ($)	0.47	0.71	0.45
Total Cost($)	96.80	23.65	1.85

ing lengths of prediction horizon.

Table 4.6: Summary of the Costs Achieved with Different Prediction Horizons

April 1st			
Cost/Prediction horizon	0.5 hrs	1 hr	1.5 hr
Comfort Cost ($)	1.33	0.90	0.97
AHU fan and HE cost ($)	0.46	0.45	0.48
Total Cost($)	1.80	1.35	1.46

4.6 Conclusion

One of the significant bottlenecks to the application of MPC is the development of reliable and easy to compute models. This chapter demonstrated that the black-box modeling algorithm proposed in Chapter 3 which can be developed and automated by just using data from BEMS has the potential to be used to apply MPC on real building systems. A method was presented whereby, the models can be used to implement MPC on EDL components of the building HVAC systems. Results from high-fidelity simulation models showed that MPC using linear black-box models can outperform the traditional PI-type controllers under varying operating conditions.

5. COMPARISON OF CENTRALIZED AND DISTRIBUTED MPC

In Chapter 4 a methodology was proposed whereby models developed using black-box identification methods were used to implement MPC using a centralized architecture. The proposed methodology was verified using high fidelity simulation models of the HVAC components of the energy distribution level. A comparison of the simulation results showed that MPC consumed 10% less energy than a PI-type control which is widely used by BMSs. The reduction in energy usage was achieved without any adverse impact on occupancy comfort.

Although implementing MPC resulted in lesser energy consumption in simulations, the centralized nature of the control architecture poses a few challenges for real-world application. Building HVAC components comprise hundreds of components. Implementing centralized MPC requires a single optimizer that contains models of the subsystem dynamics and their interactions with each other. The optimizer minimizes a centralized objective function to compute setpoints corresponding to all the subsystems being regulated. As the number of components increase, the computational burden required to calculate optimal setpoints may make the implementation infeasible.

In addition to the computational burden, centralized architecture reduces the flexibility and modularity of the HVAC operation. Several changes to the system dynamics may occur over the course of the HVAC operation. For example, components may be modified or replaced, operating conditions may change due to changes in setpoints, climatic conditions etc. The centralized model and objective function has to be updated every time such a change occurs. In order to make the control approach more modular and flexible, a methodology for implementing MPC on building HVAC systems using a distributed control architecture is proposed in this chapter. In the distributed architecture, each sub-

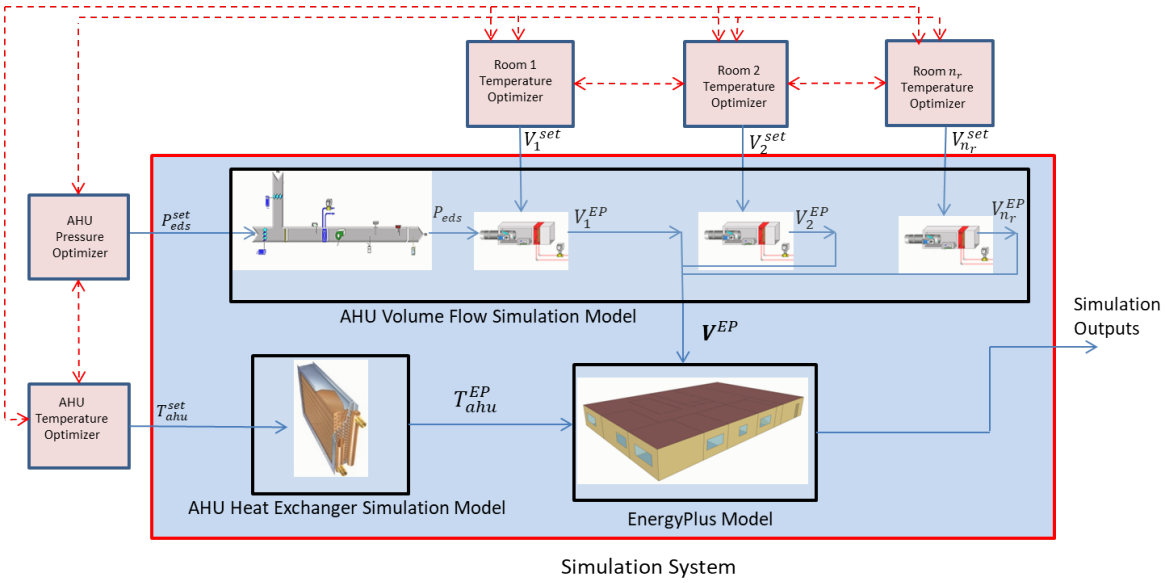


Figure 5.1: A Schematic Showing the Application of MPC with a Distributed Architecture

system is assigned its own local MPC optimizer. The optimizer computes the setpoints of only the subsystem it is assigned. For example the MPC optimizer for room 1 only computes the volume flow rate setpoint V_i^{set} for the VAV box serving room 1. Figure 5.1 is used to illustrate the application of MPC using a distributive architecture on the simulation system described in Chapter 4. In the figure the red dotted line is used to indicate the communication between the optimizers.

As opposed to the optimizer of the centralized MPC, each DMPC optimizer contains limited knowledge of the entire system. The optimizer contains knowledge of its own subsystem and a portion of the knowledge of the rest of the system which it receives via communication with other DMPC optimizers. Depending on the objective function minimized, and the amount of information that is exchanged between the optimizers, there are several different algorithms available in literature to apply MPC using a distributed control architecture.

In this chapter, the algorithm proposed by Elliot in [61] called the Neighbor Com-

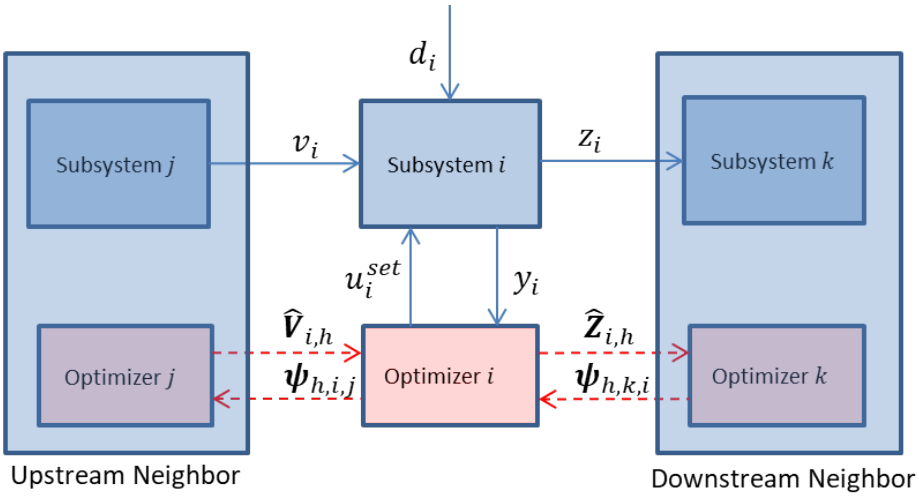


Figure 5.2: A Schematic Showing the Inputs and Outputs of the Subsystem i and the Data Communicated to and by Optimizer i.

munication Optimal Control (NC Opt) algorithm is adopted to compute the subsystem setpoints. Elliot showed that by judiciously selecting the objective function of each optimizer, a global optimal solution can be achieved by each optimizer communicating with only a limited number of other optimizers. The NC-Opt algorithm is extended in this chapter to include dynamic systems and is referred to as the NC DMPC algorithms. This chapter provides a methodology, whereby, the NC DMPC algorithm in conjunction with the automated black-box modeling algorithm can be used to apply MPC on building HVAC systems. The subsequent sections in this chapter provide a description of the procedure used to compute the setpoints of each subsystem, and the simulation results of applying the methodology on the building HVAC simulation system in Figure 5.1.

5.1 NC DMPC Algorithm Preliminaries and Procedure

In the NC DMPC algorithm, each optimizer computes the setpoints corresponding to the subsystem it is assigned to. Before detailing the procedure to compute the setpoints, the terms and definitions used by algorithm is presented with the help of Figure 5.2. The

figure shows a schematic representation of a subsystem i being regulated by its optimizer. The blue arrows are used to the physical interactions between the subsystems and their optimizers, while the red arrows correspond to the exchange of data between the optimizers.

5.1.1 NC DMPC Algorithm Preliminaries

There are several factors which influence the dynamics of any HVAC subsystem. While modeling the dynamics, however, only those factors which are measured by the BMS sensors are taken into consideration. The inputs considered for modeling the dynamics can be classified into three categories, exogenous disturbance inputs, inputs from upstream neighbors, and control setpoints computed by the DMPC optimizer. For example, for subsystem i in Figure 5.2, d_i, v_i, and u_i^{set} are the exogenous disturbance inputs, inputs from upstream neighbors, and control setpoint, respectively. The models for each subsystem are developed by taking into consideration the aforementioned categories of inputs.

The term neighboring in the preceding paragraph does not correspond to physical proximity of the subsystems. Instead two subsystems are said to be neighbors if they have interconnecting dynamics. The output of an upstream neighbor acts as an input to a downstream neighbor. For instance taking the example of subsystem i again in Figure 5.2, subsystem j is an upstream neighbor, and subsystem k is a downstream neighbor to i.

The outputs of the subsystem that are tracked by the optimizer are also classified into two categories. If the output of a subsystem acts as an input to its downstream neighbor, then it is referred to as a downstream output. On the other hand if the output is used in the computation of the cost of operation of the subsystem, then it is referred to as the component output. An output can simultaneously be downstream and component output. Referring to the Figure 5.2 once again, the downstream output and the component output of subsystem i are z_i and y_i, respectively.

As was discussed in Chapter 4, MPC computes the optimal setpoints not just for the current time instant (say t) but for a future time period known as the prediction horizon t_{pred}. The computation of the optimal setpoints over the prediction horizon, requires predictions of the downstream and component outputs over the same horizon. In this chapter, the predictions of these outputs for each type of subsystem of the energy distribution level are made using the system identification procedure described in Chapter 3. A brief description of the models is provided again in the next section for ease of reference. The predictions of the outputs over the horizon using the models can be expressed as shown in Equation 5.1

$$\begin{aligned} \hat{\boldsymbol{y}}_{h,i} &= f_{h,i}^{comp}(\hat{\boldsymbol{d}}_{h,i}, \hat{\boldsymbol{v}}_{h,i}, \boldsymbol{u}_{i,h}^{set}) \\ \hat{\boldsymbol{z}}_{h,i} &= f_{h,i}^{d}(\hat{\boldsymbol{d}}_{h,i}, \hat{\boldsymbol{v}}_{h,i}, \boldsymbol{u}_{i,h}^{set}) \end{aligned} \tag{5.1}$$

where $f_{h,i}^{comp}$ and $f_{h,i}^{d}$ are the identified models corresponding to the component and downstream outputs, respectively. The vectors $\hat{\boldsymbol{d}}_{h,i}$, $\hat{\boldsymbol{v}}_{h,i}$, and $\boldsymbol{u}_{i,h}^{set}$ refer to the estimated values over the prediction horizon of the three types of inputs to subsystem i. The vectors $\hat{\boldsymbol{y}}_{h,i}$ and $\hat{\boldsymbol{z}}_{h,i}$ refer to the predictions of the component and downstream outputs, respectively over the prediciton horizon. The $\hat{}$ symbol over a variable is used to indicate that their values are estimated either using a forecast, or predicted with the help of a model. The subsript h is assigned to a vector when it contains elements corresponding to the prediction horizon.

As was stated earlier, the optimizer does not contain all the information it requires to reach the global optimal solution. In order to perform the optimization, the DMPC optimizer receives two pieces of information from neighboring subsystems. The first piece of information is the output predictions of the upstream neighbors. The output predictions from the upstream neighbors are the upstream inputs of the subsystem under consideration

as shown below for subsystem i in Figure 5.2.

$$\hat{\boldsymbol{z}}_{h,j} = \hat{\boldsymbol{v}}_{h,i} \tag{5.2}$$

The second piece of information that is required by the optimizer are the sensitivity vector. The sensitvity vector is denoted by the symbol $\boldsymbol{\psi}_{h,k,i}$ where the subscript k refers to the downstream optimizer sending the information, and i refers to the optimizer receiving the information. The sensitivity vector comprises the sensitivity values over the prediction horizon, i.e., $\boldsymbol{\psi}_{h,k,i} = [\psi_{k,i}(t|t+1), \psi_{k,i}(t|t+2), ..., \psi_{k,i}(t|t+n_{pred})]'$, where n_{pred} is the number of control time steps in the prediction horizon. The expression $t|t+r$ is used to indicate an r-step-ahead prediction made at time instant t. Sensitivity values are a measure of how the downstream output impacts the operational cost of the downstream neighbor. The method to compute the sensitivity values is provided in the next section.

5.1.2 Construction of the NC DMPC Objective Function

The setpoints computed by the optimizer i over the prediction horizon are denoted by the vector $\boldsymbol{u}_{h,i}^{dmpc}(t) = [u_i^{dmpc}(t), u_i^{dmpc}(t+1), ..., u_i^{dmpc}(t+n_{pred}-1)]'$. The setpoint vector $\boldsymbol{u}_{h,i}^{dmpc}(t)$ is computed by minimizing an objective function $\hat{\boldsymbol{J}}_{h,i}^{dmpc}$ as shown in Equation5.3

$$\boldsymbol{u}_{h,i}^{dmpc} = \underset{\boldsymbol{u}_{h,i}^{set}}{argmin}\ \hat{\boldsymbol{J}}_{h,i}^{dmpc} \tag{5.3}$$

where $\boldsymbol{u}_{h,i}^{set}(t) = [u_i^{set}(t), u_i^{set}(t+1), ..., u_i^{set}(t+n_{pred}-1)]'$ is the vector of control input setpoints. The objective function $\hat{\boldsymbol{J}}_{h,i}^{dmpc}$ is composed of two parts as shown in Equation 5.4.

$$\hat{\boldsymbol{J}}_{h,i}^{dmpc} = \hat{\boldsymbol{J}}_{h,i}^{comp} + \hat{\boldsymbol{J}}_{h,i}^{sen} \tag{5.4}$$

The first part $\hat{\boldsymbol{J}}_{h,i}^{comp}$ referred to as component cost, is an estimate of the cost of oper-

ation of the subsystem over the prediction horizon. For example, for the AHU Pressure subsystem the component cost is the cost of energy expended by the AHU fan. The component cost is measured as a function of the component output as shown in Equation 5.5.

$$\hat{\boldsymbol{J}}_{h,i}^{comp}(t) = f_{h,i}^{c}(\hat{\boldsymbol{y}}_{h,i}(t)) \tag{5.5}$$

The second part of the objective function $\hat{\boldsymbol{J}}_{h,i}^{dmpc}$, referred to as the sensitivity cost, is a measure of how the downstream output $\hat{\boldsymbol{z}}_{h,i}$ affects the operational costs of the downstream subsystem over the prediction horizon. The sensitivity cost is constructed with the help of the predictions of the downstream output, and the sensitivity vectors sent by the downstream optimizers as shown in Equation 5.6.

$$\hat{\boldsymbol{J}}_{h,i}^{sen}(t) = \sum_{k=1}^{n_i^d} \boldsymbol{\psi}_{h,k,i}(t)' \cdot \hat{\boldsymbol{z}}_{h,i}(t) \tag{5.6}$$

In the equation above n_i^d is the number of downstream neighbors of subsystem i. The r^{th} element of the sensitivity vector $\boldsymbol{\psi}_{h,k,i}(t)$ is computed by taking the partial derivative of the cost function of subsystem k with respect to the r-step-ahead predicted value of the downstream output of i as shown in Equation 5.7.

$$\psi_{k,i}(t|t+r) = \frac{\partial \hat{\boldsymbol{J}}_{h,k}^{dmpc}(t)}{\partial \hat{z}_i(t|t+r)} \tag{5.7}$$

A sumamry of the variables and functions associated with the NC DMPC procedure for subsystem i is provided in Table 5.1.

Table 5.1: Summary of the Variables and Functions of the NCDMPC Procedure for Subsystem i

Inputs	**Description**
d_i	exogenous inputs
v_i	upstream inputs
u_i^{set}	control setpoint
Outputs	**Description**
y_i	component output
z_i	downstream output
Data received by optimizer i	**Description**
$\hat{\boldsymbol{v}}_{h,i}$	outputs from upstream neighbors
$\boldsymbol{\psi}_{h,k,i}$	sensitivities from downstream neighbors where $k \in \{1, 2, ..., n_i^d\}$
Data sent by optimizer i	**Description**
$\hat{\boldsymbol{z}}_{h,i}$	Predictions of downstream output
$\boldsymbol{\psi}_{h,i,j}$	sensitivities to upstream neighbors
Equations	**Description**
$\hat{\boldsymbol{y}}_{h,i} = f_{h,i}^{comp}(\hat{\boldsymbol{d}}_{h,i}, \hat{\boldsymbol{v}}_{h,i}, \boldsymbol{u}_{i,h}^{set})$	model of component output
$\hat{\boldsymbol{z}}_{h,i} = f_{h,i}^{d}(\hat{\boldsymbol{d}}_{h,i}, \hat{\boldsymbol{v}}_{h,i}, \boldsymbol{u}_{i,h}^{set})$	model of downstream output
$\boldsymbol{u}_{h,i}^{dmpc} = \underset{\boldsymbol{u}_{h,i}^{set}}{argmin}\ \hat{\boldsymbol{J}}_{h,i}^{dmpc}$	optimal inputs over prediction horizon
$\hat{\boldsymbol{J}}_{h,i}^{dmpc} = \hat{\boldsymbol{J}}_{h,i}^{comp} + \hat{\boldsymbol{J}}_{h,i}^{sen}$	parts of the objective function
$\hat{\boldsymbol{J}}_{h,i}^{comp}(t) = f_{h,i}^{c}(\hat{\boldsymbol{y}}_{h,i}(t))$	component cost
$\hat{\boldsymbol{J}}_{h,i}^{sen}(t) = \sum_{k=1}^{n_i^d} \boldsymbol{\psi}_{h,k,i}(t)' \cdot \hat{\boldsymbol{z}}_{h,i}(t)$	sensitivity cost
$\psi_{i,j}(t\|t+r) = \frac{\partial \hat{\boldsymbol{J}}_{h,i}^{dmpc}(t)}{\partial \hat{z}_j(t\|t+r)}$	r-step-ahead element of sensitivity vector sent to subsystem j

5.1.3 NC DMPC Algorithm Pseudo-Code

As was stated in the previous section the DMPC optimizers do not have all the information to carry out the optimization procedure by themselves. Information is exchanged between neighboring optimizers. For example, optimier i receives predictions of upstream inputs $\hat{\boldsymbol{v}}_{h,i}$ and sensitivities from downstream neighbors $\boldsymbol{\psi}_{h,k,i}$ from downstream neighbors where $k \in \{1, 2, ..., n_i^d\}$. However, the upstream and downstream optimizers themselves require data from their respective neighbors to compute the predictions and sensitivity values. Hence, in the NC DMPC procedure information is exchanged between the optimizers in an iterative manner at each control time step. These iterations are referred to as communication iterations. In this chapter, the communication iteration number is referred to by placing it in parenthesis and as a superscript to the variable under consideration.

For instance, consider again the example of subsystem i in Figure 5.2. At control time instant t and communication iteration number r, optimizer i receives the each communication iteration, the optimizer i receives the predictions $\hat{\boldsymbol{v}}_{h,i}^{(r-1)}(t)$ from the upstream neighbor j, and the sensitvity vector $\boldsymbol{\psi}_{h,k,i}^{(r-1)}(t)$ from the downstream neighbor k. The optimizer i then computes the optimal setpoint vector $\boldsymbol{u}_i^{dmpc(r)}(t)$ using Equation 5.3. After computing $\boldsymbol{u}_i^{dmpc(r)}(t)$ optimizer i then computes the downstream output predictions $\hat{\boldsymbol{z}}_{h,i}^{(r)}(t)$ and the senstivity vector $\boldsymbol{\psi}_{h,i,k}(t)^{(r)}$ which it passes on to the downstream and upstream neighbors, respectively. The process of computing the optimal setpoints and communicating predictions and sensitivities continues until the maximum allowable number of communication iterations n_{com}^{max}, or when there is no significant change in the value of the computed setpoints between successive iterations. A summary of the algorithm to compute the setpoints for subsystem i is provided below as a pseudo code.

- *Step 0:* Initialize the communication iteration number $r = 0$, and the prediction vectors of the component and downstream outputs, and the sensitivity vectors to be

passed on to upstream systems $\hat{\boldsymbol{y}}_{h,i}^{(0)}(t)$,$\hat{\boldsymbol{z}}_{h,i}^{(0)}(t)$, $\boldsymbol{\psi}_{h,i,j}^{(0)}(t)$ $j \in \{1, 2, ..., n_i^u\}$, where n_i^u is the number of upstream neighbors. The vectors are generally initialized using values from previous control time step.

- *Step 1:* Optimizer i communicates $\hat{\boldsymbol{z}}_{h,i}^{(r)}(t)$ to its downstream neighbors, and $\boldsymbol{\psi}_{h,i,j}(t)$ $j \in \{1, 2, ..., n_i^u\}$ to its upstream neighbors. At the same time the optimizer i receives $\hat{\boldsymbol{v}}_{h,i}^{(r)}(t)$ from upstream neighbors, and $\boldsymbol{\psi}_{h,i,k}^{(r)}(t)$ $k \in \{1, 2, ..., n_i^d\}$ from downstream neighbors.

- *Step 2:* Increment the iteration number $r = r + 1$. The communication data from Step 1 is used to compute $\boldsymbol{u}_{h,i}^{dmpc(r)}$ using Equation 5.3.

- *Step 3:* Optimizer i computes the component and downstream outputs $\hat{\boldsymbol{y}}_{h,i}^{(r)}(t)$ and $\hat{\boldsymbol{z}}_{h,i}^{(r)}(t)$ using Equation 5.1, and the sensitivities to the upstream neighbors $\boldsymbol{\psi}_{h,i,j}^{(r)}(t)$ using Equation 5.7.

- *Step 4:* If $\boldsymbol{u}_{h,i}^{dmpc(r)} - \boldsymbol{u}_{h,i}^{dmpc(r-1)} < \boldsymbol{u}_\delta$ or $r > n_{com}^{max}$ stop iterations, else repeat steps 1 through 4. If the change in the value of $\boldsymbol{u}_{h,i}^{dmpc(r)}$ between successive iterations is less than the threshold value or the maximum number of iterations is reached then the iterations are stopped.

- *Step 5:* Apply the first element of the vector computed in Step 3 $u_i^{dmpc(r)}(t)$ to the subsystem i.

5.2 Distributed Model Predictive Control

Whereas centralized MPC contains a single optimizer that computes the setpoints of all the subsystems, in the NC DMPC approach there are several local optimizers each serving their own subsystem. Each optimizer seeks to compute global optimal setpoints of the subsystem it regulates. The term global optimal is used to refer to the solution that

minimizes the centralized cost function designed in Chapter 4 in Equation. As was stated in the previous section, by judiciously designing the objective function of each optimizer, the NC DMPC method approaches the global optimal solution.

The NC DMPC approach in this chapter is applied in simulation to the components corresponding to the energy distribution level. The trhee subsystems corresponding to the energy distribution level are the AHU pressure subsystem, AHU temperature subsystem, and the room temperature subsystem. The following subsections provide a description of how the NC DMPC algorithm is used to compute optimal setpoints of the aforementioned subsystems.

5.2.1 Optimal Setpoints for the AHU Pressure Subsystem

The AHU Pressure subsystem comprises the AHU fan which creates a pressure differential called the end static pressure P_{eds} in the AHU. The pressure differential causes a mixture of the outside and return air to flow through the AHU heat exchanger and be delivered to the VAV boxes of each room. P_{eds} is actuated to a setpoint value P_{eds}^{set} with a help of a local PI controller that regulates the fan speed. Greater the value of P_{eds}^{set}, greater is the fan speed required. The objective of the pressure optimizer is to determine the value of P_{eds}^{set} at each control time step, so as to ahieve a global optimal solution using the NC DMPC algorithm.

Computation of optimal values of P_{eds}^{set} using the NC DMPC procedure requires a model of the outputs of the pressure subsystem. The end static pressure P_{eds} and the energy expended by the fan E_{fan} are the two outputs of the pressure subsystem that are tracked. P_{eds} acts as a downstream output as it affects the amount of conditioned air that is received by the room temperature subsystems. A model of the dynamics of the fan requires a knoweldge of the construction of the AHU, the type of fan used, the dampers in each room etc. The pressure dynamcis, however, are much faster than the room tempera-

ture dynamics. Hence, in this chapter, while modeling P_{eds} the pressure dynamics are not included. An assumption is made that in a single control time step the AHU fan is able to actuate P_{eds} to its setpoint value as shown in Equation 5.8.

$$\hat{P}_{eds}(t|t+1) = P_{eds}^{set}(t) \tag{5.8}$$

The output E_{fan} is selected as the component output, since the cost of operation of the pressure subsystem is determined by the energy expended by the fan. The energy expended by the fan is obtained from fan laws which state that the fan power is equal to the product of the volume flow rate, and the pressure differential across the fan. The one-step-ahead estimation made at time instant t of the fan energy expended is thus obtained using Equation 5.9

$$\hat{E}_{eds}(t|t+1) = \eta_{fan} \cdot \hat{P}_{eds}(t|t+1) \cdot \hat{V}_{ahu}(t|t+1) \tag{5.9}$$

where η_{fan} is the efficiency of the fan motor, and $\hat{V}_{ahu}$ is the estimate of the volume flow rate flowing through the AHU fan. $\hat{V}_{ahu}$ is an upstream input obtained by summing the volume flow rates through each of the VAV boxes being served by the AHU, as shown in Equation 5.10

$$\hat{V}_{ahu}(t|t+1) = \sum_{i=1}^{n_r} \hat{V}_i(t|t+1) \tag{5.10}$$

where n_r is the number of rooms.

The pressure optimizer computes the optimal values for P_{eds}^{set} over the prediction horizon by minimizing an objective function $\hat{\boldsymbol{J}}_{h,eds}^{dmpc}$ as shown in Equation 5.11.

$$\boldsymbol{P}_{h,eds}^{dmpc} = \underset{\boldsymbol{P}_{h,eds}^{set}}{argmin}\ \hat{\boldsymbol{J}}_{h,eds}^{dmpc} \tag{5.11}$$

The AHU Pressure subsystem objective function is made up of two parts, componet cost $\hat{\boldsymbol{J}}^{comp}_{h,eds}$ and sensitivity cost $\hat{\boldsymbol{J}}^{sen}_{h,eds}$ as shown in Equation 5.12.

$$\hat{\boldsymbol{J}}^{dmpc}_{h,eds} = \hat{\boldsymbol{J}}^{comp}_{h,eds} + \hat{\boldsymbol{J}}^{sen}_{h,eds} \tag{5.12}$$

The component cost $\hat{\boldsymbol{J}}^{comp}_{h,eds}$ is equal to the cost of the energy expended by the AHU fan over the prediction horizon and is expressed by using Equation 5.13.

$$\hat{\boldsymbol{J}}^{comp}_{h,eds} = r_{fan} \cdot \sum_{k=1}^{n_{pred}} \hat{E}_{fan}(t|t+k) \tag{5.13}$$

The sensitivity cost $\hat{\boldsymbol{J}}^{sen}_{h,eds}$ is constructed with the help of the sensitivity values $\boldsymbol{\psi}_{h,i,ahu}$ received from the downstream neighbors $i \in \{1, 2, ..., n^d_{ahu}\}$ as shown in Equation 5.14.

$$\hat{\boldsymbol{J}}^{sen}_{h,eds} = \sum_{i=1}^{n_{sel}} \boldsymbol{\psi}_{h,i,ahu}(t) \cdot \hat{\boldsymbol{P}} \tag{5.14}$$

For the pressure subsystem, the downstream neighbors are all the rooms served by the AHU, i.e., $n^d_{ahu} = n_r$. The sensitivity vector $\boldsymbol{\psi}_{h,i,eds}$ contains the sensitivity values of the objective function of subsystem i ($\hat{\boldsymbol{J}}^{dmpc}_{h,i}$) to the estimated values of the downstream output $\hat{\boldsymbol{P}}_{h,eds}$. The r-step-ahead sensitivity value corresponding to the vector $\boldsymbol{\psi}_{h,i,eds}$ is as shown in Equation 5.15.

$$\psi_{i,eds}(t|t+r) = \frac{\partial \hat{\boldsymbol{J}}^{dmpc}_{h,i}}{\partial \hat{P}_{eds}(t|t+r)} \tag{5.15}$$

A detailed discussion of how the sensitiviy vector $\boldsymbol{\psi}_{h,i,eds}$ is computed is provided in Section 5.3. Table 5.2 provides a summary of inputs, outputs, and equations used in the computation of the optimal setpoint vector corresponding to the AHU Pressure subsystem. With the help of these variables and functions the NC DMPC algorithm described in

the previous section is applied to compute the optimal setpoint vector $\boldsymbol{P}_{h,eds}^{set}$.

Table 5.2: Summary of the Variables and Functions of the NCDMPC Procedure for AHU Pressure Subsystem

Inputs	**Description**
$d_i = []$	no exogenous inputs
$v_i = [V_1, V_2, ..., V_{n_r}]$	volume flow rates from all the rooms
$u_i^{set} = P_{eds}^{set}$	end static pressure setpoint
Outputs	**Description**
$y_i = E_{fan}$	energy expended by fan
$z_i = P_{eds}$	end static pressure
Data received by optimizer	**Description**
$\hat{\boldsymbol{V}}_{h,i}$, $i \in \{1, 2, ..., n_r\}$	flow rates from all the rooms
$\boldsymbol{\psi}_{h,i,eds}$, $i \in \{1, 2, ..., n_r\}$	sensitivity vectors from all the rooms
Data sent by optimizer i	**Description**
$\hat{\boldsymbol{P}}_{h,eds}$	predictions of end static pressure
$\boldsymbol{\psi}_{h,eds,i}$, $i \in \{1, 2, ..., n_r\}$	sensitivities to all the rooms
Equations	**Description**
$\hat{P}_{eds}(t\|t+1) = P_{eds}^{set}(t)$	model of end static pressure
$\hat{E}_{eds}(t\|t+1) = \eta_{fan} \cdot \hat{P}_{eds}(t\|t+1) \cdot \hat{V}_{ahu}(t\|t+1)$	model of energy expended by fan
$\boldsymbol{P}_{h,eds}^{dmpc} = \underset{\boldsymbol{P}_{h,eds}^{set}}{argmin}\ \hat{\boldsymbol{J}}_{h,eds}^{dmpc}$	optimal values of end static pressure
$\hat{\boldsymbol{J}}_{h,eds}^{sen} = \sum_{i=1}^{n_{sel}} \boldsymbol{\psi}_{h,i,ahu}(t) \cdot \hat{\boldsymbol{P}}$	objective function parts
$\hat{\boldsymbol{J}}_{h,eds}^{comp} = r_{fan} \cdot \sum_{k=1}^{n_{pred}} \hat{E}_{fan}(t\|t+k)$	component cost
$\hat{\boldsymbol{J}}_{h,eds}^{sen} = \sum_{i=1}^{n_{sel}} \boldsymbol{\psi}_{h,i,ahu}(t) \cdot \hat{\boldsymbol{P}}$	sensitivity cost
$\psi_{i,eds}(t\|t+r) = \frac{\partial \hat{\boldsymbol{J}}_{h,i}^{dmpc}}{\partial \hat{P}_{eds}(t\|t+r)}$	r-step-ahead element of sensitivity vector received from subsystem i

5.2.2 Optimal Setpoints for the AHU Temperature Subsystem

The AHU temperature subsystem comprises a heat exchanger which conditions the air flowing through the AHU before being delivered to the rooms. The discharge air temperature T_{ahu} is actuated to a setpoint value T_{ahu}^{set} with the help of a valve that regulates the amount of CHW flowing through the heat exchanger. The objective of the temperature optimizer is to determine the value T_{ahu}^{set} at each control time step to achieve a global optimal solution.

Proceeding in the similar fashion as that of the pressure subsystem, models of the outputs of the AHU temperature subsystem are identified. The outputs of the AHU temperature subsystem that are tracked in this chapter are the discharge air temperature T_{ahu}, and the cooling energy corresponding to the CHW usage of the heat exchanger E_{he}. The temperature T_{ahu} is selected as the downstream output since it affects the amount of cooling received by each room. Similar to the procedure adopted while modeling the AHU end static pressure, the dynamics of the discharge air temperature are ignored since they are much faster than the room temperature dynamics. The discharge air temperature is assumed to have reached its setpoint value T_{ahu}^{set} by the next control time step as shown in Equation 5.16.

$$\hat{T}_{ahu}(t|t+1) = T_{ahu}^{set}(t) \tag{5.16}$$

The output E_{he} is selected as the component output, since the cost of CHW used for cooling the air through the AHU determines the cost of operation of the subsystem. The cooling energy obtained from CHW is estimated by using the principle of energy conservation as shown in Equation 5.17

$$\hat{E}_{he}(t|t+1) = \eta_{he} \cdot \rho_{air} \cdot \hat{V}_{ahu}(t|t+1) \cdot (T_{oa}(t+1) - \hat{T}_{ahu}(t|t+1)) \tag{5.17}$$

where η_{he} is the efficiency of the heat exchanger, ρ_{air} is the density of air, and T_{oa} is the temperature of air entering the AHU.

The optimal setpoint $\boldsymbol{T}_{h,ahu}^{dmpc}$ over the prediction horizon is computed by minimizing the objective function corresponding to the temperature subsystem $\hat{\boldsymbol{J}}_{h,ahu}^{dmpc}$ as shown in Equation 5.18.

$$\boldsymbol{T}_{h,ahu}^{dmpc} = \underset{\boldsymbol{T}_{h,ahu}^{set}}{argmin}\ \hat{\boldsymbol{J}}_{h,ahu}^{dmpc} \tag{5.18}$$

The objective funciton $\hat{\boldsymbol{J}}_{h,ahu}^{comp}$ comprises of two parts, the component cost $\hat{\boldsymbol{J}}_{h,ahu}^{comp}$, and the sensitivity cost $\hat{\boldsymbol{J}}_{h,ahu}^{sen}$ as shown in Equation 5.19.

$$\hat{\boldsymbol{J}}_{h,ahu}^{dmpc} = \hat{\boldsymbol{J}}_{h,ahu}^{comp} + \hat{\boldsymbol{J}}_{h,ahu}^{sen} \tag{5.19}$$

The component cost $\hat{\boldsymbol{J}}_{h,ahu}^{comp}$ is obtained by using the estimate of the cooling energy in Equation 5.17 as shown below

$$\hat{\boldsymbol{J}}_{h,ahu}^{comp} = \sum_{k=1}^{n_{pred}} r_{chw} \cdot \hat{E}_{he}(t|t+k) \tag{5.20}$$

where r_{chw} is the unit cost of chilled water usage.

As is the case for the pressure subsystem, the downstream neighbors of the temperature subsystem are each of the rooms being served by the AHU. The sensitivity cost is constructed from the sensitivity vectors $\boldsymbol{\psi}_{h,i,ahu}$ passed on from the downstream subsystem optimizers $i \in \{1, 2, ..., n_r\}$ as shown in Equation 5.21.

$$\hat{\boldsymbol{J}}_{h,ahu}^{sen} = \sum_{i=1}^{n_{sel}} \boldsymbol{\psi}_{h,i,ahu} \cdot \hat{\boldsymbol{T}}_{h,ahu} \tag{5.21}$$

A discussion of how the sensitivities of $\boldsymbol{\psi}_{h,i,eds}$ is computed is provided in the Section 5.3. A summary of all the equations used in the computation of $\hat{\boldsymbol{T}}_{h,ahu}^{dmpc}$ is provided in Table.

Table 5.3: Summary of the Variables and Functions of the NCDMPC Procedure for AHU Temperature Subsystem

Inputs	**Description**
$d_i = []$	no exogenous inputs
$v_i = [V_1, V_2, ..., V_{n_r}]$	volume flow rates from all the rooms
$u_i^{set} = T_{ahu}^{set}$	end static pressure setpoint
Outputs	**Description**
$y_i = E_{he}$	cooling energy of heat exchanger
$z_i = T_{ahu}$	discharge air temperature
Data received by optimizer	**Description**
$\hat{\boldsymbol{V}}_{h,i}$, $i \in \{1, 2, ..., n_r\}$	flow rates from all the rooms
$\boldsymbol{\psi}_{h,i,ahu}$, $i \in \{1, 2, ..., n_r\}$	sensitivity vectors from all the rooms
Data sent by optimizer	**Description**
$\hat{\boldsymbol{T}}_{h,ahu}$	predictions of discharge air temperature
$\boldsymbol{\psi}_{h,ahu,i}$, $i \in \{1, 2, ..., n_r\}$	sensitivities to all the rooms
Equations	**Description**
$\hat{T}_{ahu}(t\|t+1) = T_{ahu}^{set}(t)$	model of discharge air temperature
$\hat{E}_{he}(t\|t+1) = \eta_{he} \cdot \rho_{air} \cdot$ $\hat{V}_{ahu}(t\|t+1) \cdot (T_{oa}(t+1) - \hat{T}_{ahu}(t\|t+1))$	model of cooling energy
$\boldsymbol{T}_{h,ahu}^{dmpc} = \underset{\boldsymbol{T}_{h,ahu}^{set}}{argmin}\ \hat{\boldsymbol{J}}_{h,ahu}^{dmpc}$	optimal values of discharge air temperature setpoints
$\hat{\boldsymbol{J}}_{h,ahu}^{dmpc} = \hat{\boldsymbol{J}}_{h,ahu}^{comp} + \hat{\boldsymbol{J}}_{h,ahu}^{sen}$	objective function parts
$\hat{\boldsymbol{J}}_{h,ahu}^{comp} = \sum_{k=1}^{n_{pred}} r_{chw} \cdot \hat{E}_{he}(t\|t+k)$	component cost
$\hat{\boldsymbol{J}}_{h,ahu}^{sen} = \sum_{i=1}^{n_{sel}} \boldsymbol{\psi}_{h,i,ahu} \cdot \hat{\boldsymbol{T}}_{h,ahu}$	sensitivity cost
$\psi_{i,ahu}(t\|t+r) = \frac{\partial \hat{\boldsymbol{J}}_{h,i}^{dmpc}}{\partial \hat{T}_{ahu}(t\|t+r)}$	r-step-ahead element of sensitivity vector received from subsystem i

5.2.3 Optimal Setpoints for Room Temperature Subsystem

Computation of the optimal setpoints for the room temperature subsystems first requires an objective way of measuring thermal comfort. A brief description of the method

used to measure thermal comfort is first provided.

The principal objective of the HVAC system is to provide adequate thermal comfort to the occupants. There are several factors which determine thermal comfort such as humidity, temperature, clothing, metabolism etc. For the sake of simplicity, however, in this paper, thermal comfort of the occupants is measured as a function of only the ambient room temperature and humidity with other factors assumed to remain constant. The temperature and humidity of the room are converted into a single numerical value called Predicted Mean Vote (PMV) which serves as a measure for thermal comfort. The scale ranges from -3 for with the negative value indicating cold ambient conditions to +3 with the positive value indicating warm ambient conditions.

The value of PMV is measured as a deviation from a thermally neutral temperature (temperature at which occupant neither feels warm or cold). The thermally neutral temperature can be altered based on the clothing level of the occupant. In this paper, an upper (23^C) and a lower (21.5^C) thermally neutral temperatures are considered. An assumption is made that within the temperature range of the upper and lower thermally neutral temperatures, the occupants can maintain their desired level of comfort simply be adjusting their level of clothing. For temperatures above 23^0C the Equation 5.22 is used, and for temperatures below 21.5^0C the Equation 5.23 is used to calculate the value of the PMV of the room.

$$r_i^{pmv} = a_0 + a_1 T_i + a_2 R_i + a_3 T_i^2 + a_4 T_i R_i \tag{5.22}$$

$$r_i^{pmv} = b_0 + b_1 T_i + b_2 R_i + b_3 T_i^2 + b_4 T_i R_i \tag{5.23}$$

In the equations above r_i^{pmv} refers to the PMV value of room i, and R_i is the relative humidity in percentage. The values of the coefficients are provided in Table 5.4. A detalied description of the process used to compute PMV values is provided in Chapter 4.

In order to have consistent units with the cost of operation of the AHU pressure and

Table 5.4: Coefficients used to Compute PMV Values

Coefficient	Value	Coefficient	Value
a_0	-4.42	b_0	-5.53
a_1	0.16	b_1	0.19
a_2	-4e-3	b_2	-4.5e-3
a_3	9e-4	b_3	1.1e-3
a_4	4.2e-4	b_4	4.7e-4

temperature subsystem ($), a relationship between thermal comfort and loss in productivity is sought. An occupant's productivity in a work environment is linked to their thermal comfort. A relationship between thermal comfort scale PMV and an occupant's productivity loss in an office setting involving thinking tasks was proposed in . In Chapter 4, a quadratic fit between PMV and loss of productivity r_{pl} was obtained from the relationship proposed in []. The quadratic fit is reproduced in Equation 5.24.

$$r_i^{pl} = 8.53 + 37.97 r_i^{pmv} - 12.65 (r_i^{pmv})^2 \tag{5.24}$$

The productivity loss is multiplied with the occupant's average salary per control time step to measure thermal comfort in dollars.

The measure of thermal comfort described above is used in the construction of the objective function of the room temperature subsystem. The objective function of the room temperature subsystem is then used in the computation of the optimal flow rate setpoints V_i^{set} where $i \in 1, 2, ..., n_r$. The procedure to compute the setpoints using the objective function follows the same process as that of the AHU temperature and pressure subsystems. The method is described below.

The first task in computing the optimal setpoints is to define the outputs and inputs to the system. The outputs of the room temperature subsystems i that are tracked are

the temperature T_i and the volume flow rate through the VAV V_i. The temperature T_i is selected as the component output since its used in the measurement of thermal comfort. Although, the measurement of thermal comfort also requires the relative humidity of the room R_i, it is treated as an unmodeled disturbance in this chapter for the sake of simplicity. The downstream outputs are selected to be the volume flow rate V_i and the temperature T_i (acts both as a component output and donwstream output). The volume flow rate V_i is one of the factors that determine the fan energy and cooling energy requirement of the AHU temperature and pressure subsystems. The temperature T_i may act as an input affecting the dynamics of the neighboring room temperatures. Hence V_i and T_i are selected as the downstream outputs of subsystem i.

A model of the room temperature T_i is developed by using the system identification techniques described in Chapter 3. The system identification procedure identifies state space matrices that define the relationship between the output T_i and the inputs u_i^{sel}. In order to determine the state space matrices, a list of possible inputs u_i^{pos} that potentially impact the dynamics of the temperature of room i is first selected. The set of possible inputs is a vector containing the outside air temperature T_{oa}, the control input $u_i^c = V_i(T_i - T_{ahu})$, surrounding room temperatures T_j where $j \in N_i^{sur}$ and the outside ari relative humidity R_{oa} as shown in Table 5.5. The set N_i^{sur} contains the indices of all the neighboring room temperatures of i.

Table 5.5: List of Possible Inputs for Modeling T_i

Input	Description
T_{oa}	outside air temperature
$u_i^c = V_i(T_i - T_{ahu})$	control input
T_j where $j \in N_i^{sur}$	surrounding room temperatures
R_{oa}	outside air realtive humidity

The algorithm then selects a list of significant inputs u_i^{sel} from u_i^{pos}. The inputs T_{ao}, and u_i^c are always considered as significant inputs, hence, the input set u_i^{sel} takes the form $[T_{oa}, u_i^c, \boldsymbol{u}_i^{rem}]$, where $\boldsymbol{u}_i^{rem}$ is a set of the remaining significant inputs. The state space matrices that define the relationship between T_i and u_i^{se} are then identified and expressed as shown in Equation 5.25

$$\begin{aligned} \dot{x}(k+1) =& A_i x(k) + B_i u_i^{sel}(k) + K_i e(k) \\ \hat{T}_i(k) =& C_i x(k) \end{aligned} \tag{5.25}$$

where x is a vector containing the states, and A_i, B_i, C_i, and K_i are the identified state space matrices.

The second output that is modeled is the flow rate through the VAV box V_i. The model of the flow through the VAV box adopts the same procedure as that of the T_{ahu} and P_{eds}, i.e, the dynamics of the flow rate are not included in the model as they are much faster than the room temperature dynamics. An assumption is made that V_i is able to track its setpoint value V_i^set by the next control time step as shown in Equation 5.26.

$$\hat{V}_{i,t|t+1} = V_i^{set}(t) \tag{5.26}$$

The optimal flow setpoints $\boldsymbol{V}_{h,i}^{dmpc}$ over the prediction horizon is computed by minimizing the objective function $\hat{\boldsymbol{J}}_{h,i}^{dmpc}$ as shown in Equation 5.27.

$$\boldsymbol{V}_{h,i}^{dmpc} = \underset{\boldsymbol{V}_{h,i}^{set}}{argmin}\ \hat{\boldsymbol{J}}_{h,i}^{dmpc} \tag{5.27}$$

The objective function $\hat{\boldsymbol{J}}_{h,i}^{dmpc}$ comprises of two parts, the component cost $\hat{\boldsymbol{J}}_{h,i}^{comp}$ and the

sensitivity cost $\hat{\boldsymbol{J}}_{h,i}^{sen}$ as shown in Equation 5.28.

$$\hat{\boldsymbol{J}}_{h,i}^{dmpc} = \hat{\boldsymbol{J}}_{h,i}^{comp} + \hat{\boldsymbol{J}}_{h,i}^{sen} \tag{5.28}$$

The productivity loss ratio described earlier in the section is used in the construction of the component cost. First, the state space relationship defined in Equation5.25 are used to predict the temperatures over the prediction horizon $\hat{\boldsymbol{T}}_{h,i}$. Using the predicted temperatures and the last measured value of the room relative humidity $R_i(t)$, the PMV values of the room over the prediction horizon $\hat{\boldsymbol{r}}_{h,i}^{pmv}$ are computed using either Equation 5.23 or Equation 5.22 as the case may be. The PMV values are then used to compute the productivity loss ratio over the prediction horizon $\boldsymbol{r}_{h,i}^{pl}$ using Equation 5.24. Finally, the component cost is computed using $\boldsymbol{r}_{h,i}^{pl}$ as shown in Equation 5.29

$$\hat{\boldsymbol{J}}_{h,i}^{comp}(t) = \sum_{k=1}^{n_{pred}} r_i^{sal} \cdot r_{i,t|t+k}^{pl} \tag{5.29}$$

where r_i^{sal} is the occupant's average salary per duration of the control time step.

The second part of the objective function is the sensitivity cost $\hat{\boldsymbol{J}}_{h,i}^{sen}$. The sensitivity cost is constructed with the help of the sensitivity values associated with the downstream outputs. As stated earlier the room temperature subsystem, has two downstream outputs, V_i and T_i. The downstream subsystems corresponding to the output V_i are the AHU temperature and pressure subsystems. Sensitivity vectors $\boldsymbol{\psi}_{h,eds,i}$ and $\boldsymbol{\psi}_{h,ahu,i}$ are sent by the AHU pressure and temperatuer optimizers, respectively.

The downstream subsystems corresponding to the output T_i are the neighboring room temperatures which have T_i as one of their significant input. The neighbor with T_i as a significant input (room j say) sends the sensitivity vector $\boldsymbol{\psi}_{h,j,i}$ to the optimizer i. A discussion of how all sensitivity vectors are computed is provided in the next section. Con-

struction of the sensitivity cost, first requires the identification of the downstream outputs of the subsystem. There are two downstream outputs of the room temperature subsystem that are tracked, the volume flow rate V_i and the temperature T_i. The downstream neighbors corresponding to the output V_i are the AHU pressure and temperature subsystem. The total flow rate of the conditioned air through the AHU is the sum of the flow rates through the VAV box of each room. Hence a change in the value of V_i affects the total fan and cooling energy of the pressure and temperature subsystems, respectively.

The estimated values of the flow rates over the prediction horizon $\hat{\boldsymbol{V}}_{h,i}$ are sent to the downstream neighbors' optimizers. $\hat{\boldsymbol{V}}_{h,i}$ is used to compute the sensitivities to the cost function $\boldsymbol{\psi}_{h,eds,i}$ and $\boldsymbol{\psi}_{h,ahu,i}$ corresponding to the AHU pressure and temperature subsystem, respectively. A discussion of how to calculate the aforementioned sensitivities is provided in the next section. The sensitivity vectors are used to construct the sensitivity cost as shown below.

$$\hat{\boldsymbol{J}}_{h,i}^{sen} = \boldsymbol{\psi}_{h,eds,i} \cdot \hat{\boldsymbol{V}}_{h,i} + \boldsymbol{\psi}_{h,ahu,i} \cdot \hat{\boldsymbol{V}}_{h,i} + \sum_{j \in N_i^{sur}} \boldsymbol{\psi}_{h,j,i} \cdot \hat{\boldsymbol{T}}_{h,i} \tag{5.30}$$

The objective function corresponding to the room temperature subsystem in Equation 5.28 can be expressed with the help of the component cost and the sensitivity cost in Equations 5.29 and 5.30. The NCDMPC algorithm in the previous section is then used in the computaiton of $\boldsymbol{V}_i^{set}$ for all $i \in \{1, 2, ..., n_r\}$. A summary of the inputs, outputs and equations associated with the computation of $\boldsymbol{V}_i^{set}$ is provided in Table 5.6.

5.3 Sensitivity to Cost Computation

The construction of the sensitivity cost of each subsystem in the previous section requires sensitivity vectors from downstream neighbors. This section provides a description of how the sensitivity vector associated with each type of subsystem is computed.

Table 5.6: Summary of the Variables and Functions of the NCDMPC Procedure for Room i Temperature Subsystem

Inputs	**Description**
$d_i = [T_{oa}, R_{oa}]$	exogenous inputs
$v_i = [P_{eds}, T_{ahu}]$	upstream inputs
$u_i^{set} = V_i^{set}$	volume flow rate setpoint
Outputs	**Description**
$y_i = T_i$	component output
$z_i = [V_i, T_i]$	downstream outputs
Data received by optimizer	**Description**
$\hat{\boldsymbol{P}}_{h,eds}$, $\hat{\boldsymbol{T}}_{h,ahu}$, $\hat{\boldsymbol{T}}_j$ $j \in N_i^{sur}$	predictions from upstream
$\boldsymbol{\psi}_{h,ahu,i}$, $\boldsymbol{\psi}_{h,eds,i}$, $\boldsymbol{\psi}_{h,eds,i}$	sensitivityfrom downstream
Data sent by optimizer	**Description**
$\hat{\boldsymbol{T}}_{h,i}$ $\hat{\boldsymbol{V}}_{h,i}$	predictions of downstream outputs
$\boldsymbol{\psi}_{h,i,j}$ $j \in N_i^{sur}$, $\boldsymbol{\psi}_{h,i,eds}$, $\boldsymbol{\psi}_{h,i,ahu}$	sensitivities to upstream inputs
Equations	**Description**
$\dot{x}(k+1) = A_i x(k) + B_i u_i^{sel}(k) + K_i e(k)$	state space model
$\hat{T}_i(k) = C_i x(k)$	
$\hat{V}_{i,t\|t+1} = V_i^{set}(t)$	volume flow rate model
$\boldsymbol{V}_{h,i}^{dmpc} = \underset{\boldsymbol{V}_{h,i}^{set}}{argmin}\ \hat{\boldsymbol{J}}_{h,i}^{dmpc}$	optimal volume flow rate setpoints
$\hat{\boldsymbol{J}}_{h,i}^{dmpc} = \hat{\boldsymbol{J}}_{h,i}^{comp} + \hat{\boldsymbol{J}}_{h,i}^{sen}$	objective function parts
$\hat{\boldsymbol{J}}_{h,i}^{comp}(t) = \sum_{k=1}^{n_{pred}} r_i^{sal} \cdot r_{i,t\|t+k}^{pl}$	component cost
$\hat{\boldsymbol{J}}_{h,i}^{sen} = \boldsymbol{\psi}_{h,eds,i} \cdot \hat{\boldsymbol{V}}_{h,i} +$ $\boldsymbol{\psi}_{h,ahu,i} \cdot \hat{\boldsymbol{V}}_{h,i} + \sum_{j \in N_i^{sur}} \boldsymbol{\psi}_{h,j,i} \cdot \hat{\boldsymbol{T}}_{h,i}$	sensitivity cost
$\psi_{eds,i}(t\|t+r) = \frac{\partial \hat{\boldsymbol{J}}_{h,eds}^{dmpc}}{\partial \hat{V}_i(t\|t+r)}$	sensitivity from AHU pressure optimizer
$\psi_{ahu,i}(t\|t+r) = \frac{\partial \hat{\boldsymbol{J}}_{h,ahu}^{dmpc}}{\partial \hat{V}_i(t\|t+r)}$	sensitivity from AHU temp optimizer
$\psi_{j,i}(t\|t+r) = \frac{\partial \hat{\boldsymbol{J}}_{h,j}^{dmpc}}{\partial \hat{T}_i(t\|t+r)}$	sensitivity from optimizer j

5.3.1 Sensitivity to Downstream Output of AHU Pressure Subsystem

The downstream output corresponding to the AHU pressure subsystem is the end static pressure P_{eds}, and the downstream neighbors of the pressure subsystem are all the rooms being served by the AHU. Each room temperature optimizer computes the sensitvity vector $\boldsymbol{\psi}_{h,i,eds}$ which the sensitivity of its objective function $\hat{\boldsymbol{J}}_{h,i}^{dmpc}$ to the predictions of end static pressure $\hat{\boldsymbol{P}}_{h,eds}$.

The sensitivity vector computed at control time step t, $\boldsymbol{\psi}_{h,i,eds}(t)$, is composed of sensitivity values corresponding to each future time step of the prediction horizon as shown in Equation 5.31.

$$\boldsymbol{\psi}_{h,i,eds}(t) = [\psi_{i,ahu}(t|t+1), \psi_{i,ahu}(t|t+2), ..., \psi_{i,ahu}(t|t+n_{pred})] \tag{5.31}$$

The computation of the element $\psi_{i,eds}(t|t+k)$ of the sensitivity vector requires a knowledge of how $\hat{P}_{eds}(t|t+k)$ impacts the component cost function of the downstream neighbor $\hat{\boldsymbol{J}}_{h,i}^{comp}(t)$ as shown in Equation 5.32.

$$\psi_{i,eds}(t|t+k) = \frac{\partial \hat{\boldsymbol{J}}_{h,i}^{comp}(t)}{\partial \hat{P}_{eds}(t|t+k)} \tag{5.32}$$

However, since a direct relationship between $\boldsymbol{P}_{h,eds}$ and $\hat{\boldsymbol{J}}_{h,i}^{comp}$ has not been directly modeled, the partial derivative in the equation above is computed through a series of intermediary variables as shown in Equation 5.33.

$$\begin{aligned}\psi_{i,eds}(t|t+k) = \frac{\partial \hat{\boldsymbol{J}}_{h,i}^{comp}(t)}{\partial \hat{P}_{eds}(t|t+k)} = \sum_{j=1}^{n_{pred}} \frac{\partial \hat{\boldsymbol{J}}_{h,i}^{comp}(t)}{\partial r_i^{pmv}(t|t+j)} \cdot \frac{\partial r_i^{pmv}(t|t+j)}{\partial \hat{T}_i(t|t+j)} \cdot \\ \frac{\partial \hat{T}_i(t|t+j)}{\partial \hat{V}_i(t|t+k)} \cdot \frac{\partial \hat{V}_i(t|t+k)}{\partial \hat{P}_{eds}(t|t+k)}\end{aligned} \tag{5.33}$$

The method to compute each of the partial derivatives in the equation above is described below.

The first partial $\frac{\partial \hat{\boldsymbol{J}}_{h,i}^{comp}(t)}{\partial r_i^{pmv}(t|t+j)}$ derivative corresponds to the relationship between the component cost and the k-step-ahead PMV value . This relationship is obtained from Equations 5.29 and 5.24 and is expressed below.

$$\frac{\partial \hat{\boldsymbol{J}}_{h,i}^{comp}(t)}{\partial r_i^{pmv}(t|t+j)} = r_i^{sal} \cdot (c_1 + 2c_2 \cdot r_i^{pmv}(t|t+j)) \tag{5.34}$$

The second partial derivative $\frac{\partial V_{i,t|t+j}^{pmv}}{\partial \hat{T}_{i,t|t+j}}$ corresponds to the relationship between the j-step-ahead predicted temperature and the corresponding PMV value. The relationship is obtained by using either Equation5.22 or 5.23 depending on whehter the predicted temperature is greater than 23^0C or less than 21.5^0C. Where the predicted temperature is between 21.5^0C and 23^0C there is no change in the PMV value, hence the derivative is 0. The partial derivative is computed as shown below.

$$\frac{\partial r_i^{pmv}(t|t+j)}{\partial \hat{T}_i(t|t+j)} = \begin{cases} a_1 + 2a_3 \cdot \hat{T}_i(t|t+j) + a_4 \cdot R_i(t) & \hat{T}_i(t|t+j) \geq 23^0C \\ b_1 + 2b_3 \cdot \hat{T}_i(t|t+j) + b_4 \cdot R_i(t) & \hat{T}_i(t|t+j) \leq 21.5^0C \\ 0 & 21.5 < \hat{T}_i(t|t+j) < 23 \end{cases} \tag{5.35}$$

The third partial derivative $\frac{\partial \hat{T}_{i,t|t+j}}{\partial \hat{V}_{i,t|t+k}}$ corresponds to the relationship between the j-step ahead predicted temperature and the $k-1$-step-ahead flow rate setpoint. In order to find the the relationship, consider the state space model of room temperature in Equation 5.25. The state space equation is rewritten with the input set u_i^{se} expanded into its constituent

elements in Equation 5.36.

$$\begin{aligned} x(k+1) =& A_i x(k) + B_i^{oa} T_{oa} + B_i^c V_i^{set}(k) \cdot (T_i(k) - T_{ahu}(k)) + B_i^{rem} \boldsymbol{u}_i^{rem} + K_i e \\ T_i(k) =& C_i x(k) \end{aligned} \tag{5.36}$$

In the equation above, the matrices B_i^{oa}, B_i^c, and B_i^{ref} are the column vectors of B_i corresponding to the inputs T_{oa}, $V_i^{set}(k)(T_i(k) - T_{ahu}(k))$ and $\boldsymbol{u}_i^{rem}$, respectively. The control input of the state space is bilinear in nature, i.e. the control input contains terms which involve the product of two variables $V_i^{set}(k)$ and $T_i(k)$. In addition, the control input is dependent on the output of the previous time step. These factors make it difficult to find a simple algegraic relationship between the j-step-ahead prediction of temeprature and the $k-1^{th}$ flow rate setpoint $V_i^{set}(t+k-1)$. Instead of determining the algebraic relationship between the two variables, the partial derivatives $\frac{\partial \hat{T}_i(t|t+j)}{\partial \hat{V}_i(t|t+k)}$ $\forall j \in \{1, 2, ..., n_{pred}\}$ and $k \in \{1, 2, ..., n_{pred}\}$ is obtained through an iterative procedure.The procedure is detailed with the help of the following algorithm.

- *Step 0* Initialize iteration variables $j = 1$ and $k = 1$.
- *Step 1* Compute the partial derivative shown below using the last measured values of room and AHU discharge air temperature ($T_i(t)$ and $T_{ahu}(t)$).

$$\begin{aligned} \frac{\partial x_i(t+j)}{\partial V_i^{set}(t+k-1)} = \frac{\partial x_i(t+1)}{\partial V_i^{set}(t)} &= -B_i^u \cdot (T_i(t) - T_{ahu}(t) \\ \frac{\partial \hat{T}_i(t|t+1)}{\partial V_i^{set}(t)} &= C \frac{\partial x_i(t+1)}{\partial V_i^{set}(t)} \end{aligned} \tag{5.37}$$

- *Step 2* Increment the value of j by 1. $j = j + 1$.
- *Step 3* Compute the partial derivative of the j-step-ahead temperature with respect

to the $k-1$ step ahead flow setpoint using the equations below.

$$\frac{\partial x_i(t+j)}{\partial V_i^{set}(t+k-1)} = \begin{cases} A_i \frac{\partial x_i(t+j-1)}{\partial V_i^{set}(t+k-1)} - B_i^c V_i^{set}(t+j-1) \frac{\partial \hat{T}_i(t+j-1)}{\partial V_i^{set}(t+k-1)} & j > k \\ -B_i^c(\hat{T}_i(t|t+j-1) - \hat{T}_{ahu}(t|t+j-1)) & j == k \\ 0 & j < k \end{cases} \tag{5.38}$$

$$\frac{\partial \hat{T}_i(t|(t+j))}{\partial V_i^{set}(t+k-1)} = C \frac{\partial x_i(t+j)}{\partial V_i^{set}(t+k-1)} \tag{5.39}$$

- *Step 4* Depending on the value of j and k perform one of the following steps.

$$\begin{cases} \text{Repeat iterations (Steps 2 to 4)} & j < n_{pred} \;\&\&\; k < n_{pred} \\ \Big[k = k+1, \text{Set } j = k-1 & \\ \quad \text{Repeat iterations (Steps 2 to 4)} \Big] & j == n_{pred} \;\&\&\; k < n_{pred} \\ \text{stop iterations} & j == n_{pred} \;\&\&\; k == n_{pred} \end{cases} \tag{5.40}$$

In order to compute the fourth partial derivative $\frac{\partial V_i^{set}(t+k-1)}{\partial \hat{P}_{eds}(t|t+j)}$ consider the simulation model of the VAV box derived in Chapter 3 and reproduced below in Equation 5.41

$$\hat{V}_i(t|t+1) = \Big[a \cdot D_i(t|t+1) + b \cdot D_i^2(t|t+1) \Big] \sqrt{\hat{P}_{eds}(t|t+1)} \tag{5.41}$$

where a and b are parameters obtained through regression analysis, and D_i is the damper position of the VAV box i. The maximum flow rate through the VAV box occurs when the damper position is fully open, i.e. $D_i = 1$. The maximum possiple flow rate at $t+1$ for end static pressure value of $P_{eds}(t|t+1)$ is given by $V_i^{fd}(t+1)$ and is expressed as shown in Equation 5.42

$$\hat{V}_i^{fd}(t|t+1) = (a+b)\sqrt{\hat{P}_{eds}(t|t+1)} \tag{5.42}$$

When a flow setpoint value $V_i^{set}(t)$ is provided to the VAV box, a local PI controller regulates the damper position D_i to actuate the flow rate $\hat{V}_i(t|t+1)$ to equal to $V_i^{set}(t)$ as long as $V_i^{set}(t)$ is less than $\hat{V}_i^{fd}(t|t+1)$. Hence under all cases where $V_i^{set}(t)$ is less than $\hat{V}_i^{fd}(t|t+1)$ the desired flow rate can be achieved by the loacl controller simply by regulating the damper position. Hence a small change in the value of $\hat{P}_{eds}(t|t+1)$ has no impact on the room temperature dynamics when the damper is not fully open.

In the limiting case, however, i.e. when the dampers are fully open the following equation holds.

$$V_i^{set}(t) = \hat{V}_i^{fd}(t|t+1) = \hat{V}_i(t|t+1) = (a+b)\sqrt{\hat{P}_{eds}(t|t+1)} \tag{5.43}$$

The local controller cannnot increase the damper position to meet an increase in the flow set point value. A change in $\hat{P}_{eds}(t|t+1)$ thus affects the flow rate through the room. Thus the partial derivative $\frac{\partial V_i^{set}(t+k-1)}{\partial \hat{P}_{eds}(t|t+j)}$ has a nonzero value only during the limiting case. By using Equations and , the fourth partial derivative can be computed as follows.

$$\frac{\partial V_i^{set}(t+k-1)}{\partial \hat{P}_{eds}(t|t+k)} = \begin{cases} \frac{a+b}{2\sqrt{\hat{P}_{eds}(t|t+k)}} & \hat{D}_i(t|t+k) > 0.95 \\ 0 & \hat{D}_i(t|t+k) < 0.95 \end{cases} \tag{5.44}$$

5.3.2 Sensitivity to Downstream Outputs of AHU Temp Subsystem

The downstream output corresponding to the AHU temperature subsystem is the discharge air temperature T_{ahu}, and the downstream neighbors are all the rooms being served by the AHU. Each room temperature optimizer computes $\boldsymbol{\psi}_{h,i,ahu}$ which is sent to the AHU temperature optimizer.

The sensitivity value $\boldsymbol{\psi}_{h,ahu,i}$ is the partial derivative of the objective function of room i,$\hat{\boldsymbol{J}}_{h,i}^{dmpc}$, with respect to the predicted AHU discharge air temperature $\hat{T}_{ahu}$. In this paper

a single setpoint for the ahu discharge air temperature for the entire prediction horizon is selected to reduce the computational burden. As as result, the sensitivity vector $\boldsymbol{\psi}_{h,ahu,i}$ consists of a single element as shown in Equation 5.45.

$$\psi_{i,ahu}(t) = \frac{\partial \hat{\boldsymbol{J}}_{h,i}^{comp}(t)}{\partial \hat{T}_{ahu}(t+1)} \tag{5.45}$$

Following the same procedure that is used to compute $\boldsymbol{\psi}_{ahu,i}(t)$, the partial derivative $\frac{\partial \hat{\boldsymbol{J}}_{h,i}^{comp}(t)}{\partial T_{ahu}^{set}(t)}$ is computed with the help of intermediary variables as shown in Equation 5.46.

$$\psi_{i,ahu}(t) = \frac{\partial \hat{\boldsymbol{J}}_{h,i}^{comp}(t)}{\partial \hat{T}_{ahu}(t|t+k)} = \sum_{j=1}^{n_{pred}} \frac{\partial \hat{\boldsymbol{J}}_{h,i}^{comp}(t)}{\partial r_i^{pmv}(t|t+j)} \cdot \frac{\partial r_i^{pmv}(t|t+j)}{\partial \hat{T}_i(t|t+j)} \cdot \frac{\partial \hat{T}_i(t|t+j)}{\partial \hat{T}_{ahu}(t|t+k)} \tag{5.46}$$

Since the setpoint value T_{ahu}^{set} is assumed to be constant throughout the prediction horizon, the predicted ahu discharge air temperature $\hat{T}_{ahu}(t|t+k)$ is the same as T_{ahu}^{set} $\forall k \in \{1, 2, ..., n_{pred}\}$.

The first two partial derivatives in Equation 5.46 are the same as those described in Section 5.3.1. The state space system expressed in Equation 5.36 shows the relationship between the room temperature at time t and the AHU discharge air temperature at time $t-1$. The relationship is used in an iterative manner to compute $\frac{\partial \hat{V}_i(t|t+j)}{\partial \hat{T}_{ahu}(t|t+k)}$. The value of j and k are first initialized to 1. The iteration variable j is then incremented by 1 until it reaches the value n_{pred}. For each value of j the partial derivative $\frac{\partial \hat{T}_i(t|t+j)}{\partial T_{ahu}^{set}(t+k-1)}$ is computed using Equations 5.47 and 5.48. Since the value of $\hat{T}_{ahu}(t|t+k)$ is assumed to be constant for $\forall k \in \{1, 2, ..., n_{pred}\}$ its value is kept constant in the iteration.

$$\frac{\partial x_i(t+j)}{\partial T_{ahu}^{set}(t+k-1)} = \begin{cases} BV_i(t) & j == 1 \\ A\frac{\partial x_i(t+j-1)}{\partial T_{ahu}^{set}(t+k-1)} + B\hat{V}_i(t|t+j-1) & j > 1 \end{cases} \tag{5.47}$$

$$\frac{\partial \hat{T}_i(t|t+j)}{\partial T_{ahu}^{set}(t+k-1)} = C\frac{\partial x_i(t+j)}{\partial T_{ahu}^{set}(t+k-1)} \tag{5.48}$$

5.3.3 Sensitivity to Downstream Outputs of Room Temp Subsystem

The two downstream outputs of the room temperature subsystem i are the flow rate through the VAV box V_i and the temperature T_i. Each downstream output is associated with its own downstream subsystems and sensitivity vectors. The sensitivity vectors corresponding to V_i is first provided below.

The downstream subsystems corresponding to V_i are the AHU pressure and temperature subsystems. The pressure and temperature optimizers send the sensitivity vectors $\boldsymbol{\psi}_{h,eds,i}$ and $\boldsymbol{\psi}_{h,ahu,i}$, respectively, to the room temperature optimizer i. The sensitivity vectors are composed of the sensitivity values corresponding to each time step of the prediction horizon as shown below.

$$\begin{aligned} \boldsymbol{\psi}_{h,eds,i} &= [\psi_{eds,i}(t+1), \psi_{eds,i}(t+2), ..., \psi_{eds,i}(t+n_{pred})] \\ \boldsymbol{\psi}_{h,ahu,i} &= [\psi_{ahu,i}(t+1), \psi_{ahu,i}(t+2), ..., \psi_{ahu,i}(t+n_{pred})] \end{aligned} \tag{5.49}$$

The volume flow rate through a particular VAV box is a fraction of the total volume flow rate through the AHU. Increase in the flow rate of the conditioned air to the room, thus increases the total flow rate through the AHU, thereby increasing the fan energy cost and heat exchanger cooling energy cost. The sensitivity values $\psi_{eds,i}(t+k)$ and $\psi_{ahu,i}(t+k)$ are the measure of how the k-step ahead flow rates of the VAV box i impact the AHU pressure and temperature objective functions, respectively. The sensitivity value $\psi_{eds,i}(t+k)$ is

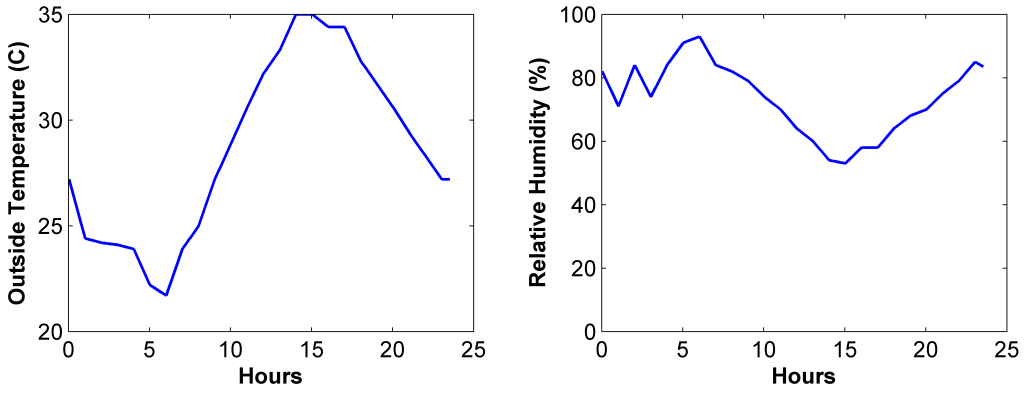


Figure 5.3: Outside Air Temperature and Relative Humidity on Day of Simulation

computed using Equations 5.9 and 5.13 and is expressed as shown below.

$$\psi_{eds,i}(t+k) = r_{fan} \cdot \eta_{fan} \cdot \hat{P}_{eds}(t|t+k) \tag{5.50}$$

The sensitivity value $\psi_{ahu,i}(t+k)$ is computed using Equations 5.17 and 5.20 and is expressed as shown below.

$$\psi_{ahu,i}(t+k) = r_{fan} \cdot \eta_{fan} \cdot \hat{P}_{eds}(t|t+k) \tag{5.51}$$

5.4 Results

Energy Plus simulations are run for a period of 1 day with the outside air conditions taken from the weather file corresponding to the location College Station in Texas on June 1st. The outside air temperature and humidity are shown in Figure 5.3. The occupancy of the rooms of the building is changed randomly between the times 7:00 am and 6:00 pm. The occupancy of the rooms 1 to 4 on the day of the simulation is shown in Figure

The NC DMPC algorithm is applied to compute the setpoints corresponding to the AHU end static pressure and discharge air temperature, and flow rates through the VAV

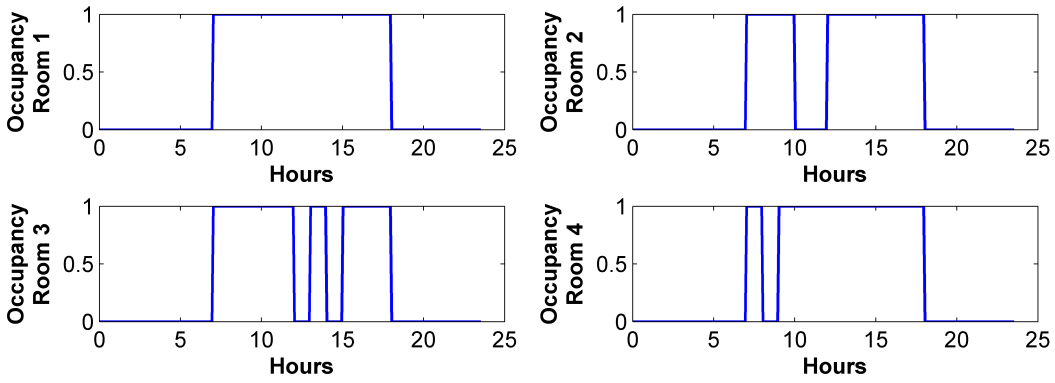


Figure 5.4: Occupancies of Rooms 1 to 4

box of each rooms. The values of the objective function parameters r_{fan}, r_{chw}, and r^i_{sal} corresponding to real world operation which were derived in Chapter 4 are first used to run the NC DMPC simulations . In addition to simulations where the parameters reflect real world operation, simulations are also performed corresponding to two special cases. The special cases reflect operating conditions with high fan energy cost, and high cooling energy cost,respectively. Although the special cases do not reflect a real-world scenario, they provide an insight into how the algorithm regulates the AHU end static pressure and discharge air temperature, and the volume flow rates to the rooms.

Table 5.7: Cost Parameters Associated with Different Operating Conditions

Case	r_{fan}	r_{he}	r^{yr_sal} ($)
Real world	1.101e-6	1.078 e-7	20,000
High fan cost	1.101e-4	1.078 e-7	200
High cooling cost	1.101e-6	1.078 e-5	200

Table 5.8: Summary of the Costs Associated with Different Control Methodolgies Under Real World Cost Operation

	PI (low sp)	PI (high sp)	centralized MPC	NC DMPC
Total cost	25.12	103.63	1.79	1.91
Comfort cost	24.44	103.18	1.33	1.61
HVAC operation cost	0.680	0.44	0.46	0.30

Table 5.9: Summary of the Costs Associated with Different Control Methodolgies Under High Fan Cost Operation

	PI (high sp)	PI (low sp)	centralized MPC	NC DMPC
Total cost	60.17	38.92	4.26	4.22
Comfort cost	0.24	1.03	2.49	2.44
HVAC operation cost	59.92	37.89	1.77	1.77

Table 5.10: Summary of the Costs Associated with Different Control Methodolgies Under High Cooling Cost Operation

	PI (high sp)	PI (low sp)	centralized MPC	NC DMPC
Total cost	9.47	8.19	4.98	3.88
Comfort cost	0.24	1.03	0.84	2.73
HVAC operation cost	9.24	7.16	4.14	1.14

5.4.1 Real World Cost Operation

The objective function parameters r_{fan}, r_{chw}, and r^i_{sal} corresponding to real world operation are shown in Table 4.3 . A prediction horizon of $t_{pred} = 30$ minutes is selected which corresponds to 6 control time steps, i.e. $n_{pred} = 6$. The results of applying the NC DMPC algorithm is shown with the help of simulated temperatures of room 1 in Figure 5.5. The figure also shows the temperatures 'higher productivity limit' and 'lower productivity limit' which represent the bounds within which there is no loss in productivity of the occupants. In addition, the figure also shows the simulated temperatures using the existing

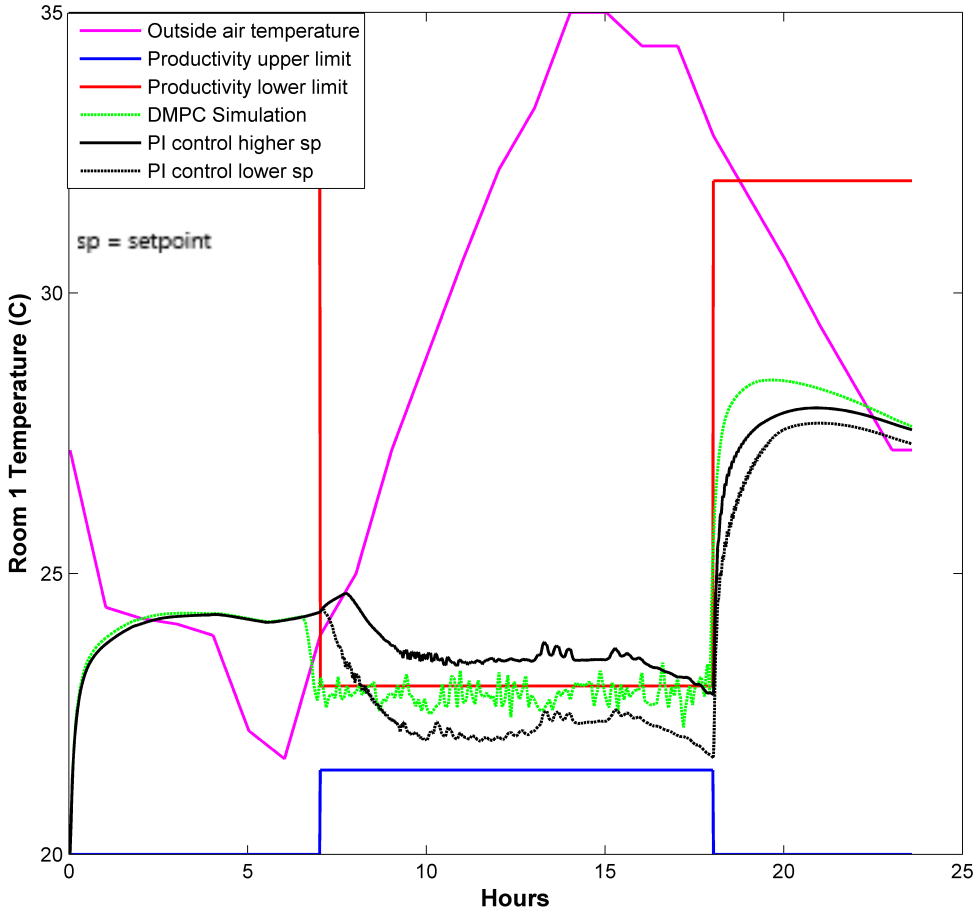


Figure 5.5: Room 1 Temperature Simulation Results

PI-type control methodology to compute the setpoints. There are two simulations that are run using the PI-type control methodology. First, where the setpoint temperatures are equal to the higher productivity limit temperatures, and second where the setpoint temperatures are equal to the lower productivity limit temperatures. Figure shows a comparison of the AHU end static pressure setpoint and the discharge air temperature setpoint computed by the NC DMPC algorithm with the setpoints computed by the two PI control operations.

The results above demonstrate the two methods by which MPC seeks to improve upon

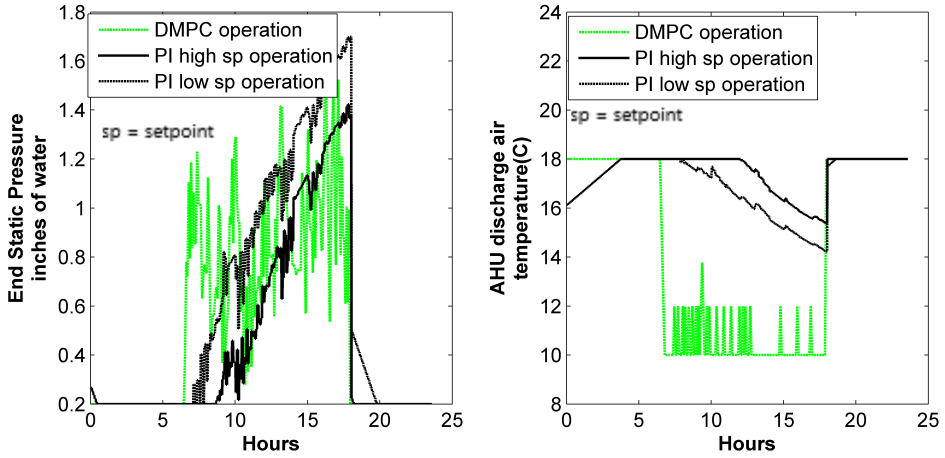


Figure 5.6: End Static Pressure and Discharge Air Temperature Simulation Results

the performance achieved by PI-type control, component optimization and planning ahead by looking at the prediction horizon. There are several different combinations of volume flow rates and discharge air temperatures that can provide the same amount of cooling. However, the most economical operation is achieved by setting the dishcarge air temperature to the lowest allowable value (10^0C) and then providing the necessary amount of cooling by regulating the end static pressure. In addition, the ability to forecast future cooling requirements results in the NCDMPC algorithm pre cooling the room. As can be seen in Figure 5.5, the room starts cooling at 6:30 am using the NCDMPC algorithm and the occupancy of the room starts at 7:00 am.

A comparison of the hourly cost achieved using the NC DMPC algorithm with the two simulations with PI-type (higher productivity setpoint, and lower productivity setpoint) control, and the centralized MPC simulation are shown in Figure 5.7. The two peaks associated with the PI operation at around 7:00 am and 10:30 am, occur due to the fact that PI control does not precool the room. As a result there is a an initial period of time where the occupant's thermal comfort requirements are not met. The time taken to cool the

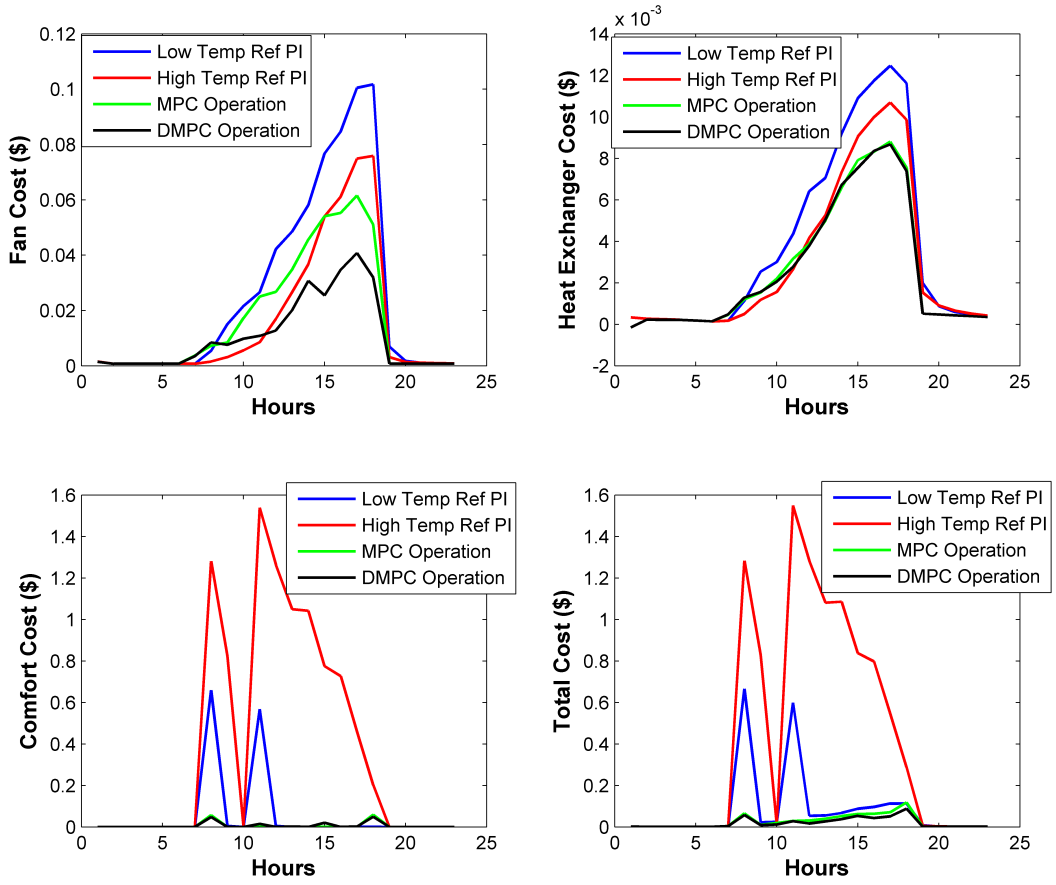


Figure 5.7: A Comparison of the Hourly Costs Achieved Using Various Control Methodologies with Real World Cost Parameters

room to the desired level depends on its thermal capacity. Hence, greater the thermal mass of the building, greater is the time taken to reach the desired temperature. Table shows the total cost achieved over the entire day using each of the aforementioned types of control methodologies. A summary of the costs achieved is shown in Table 5.8.

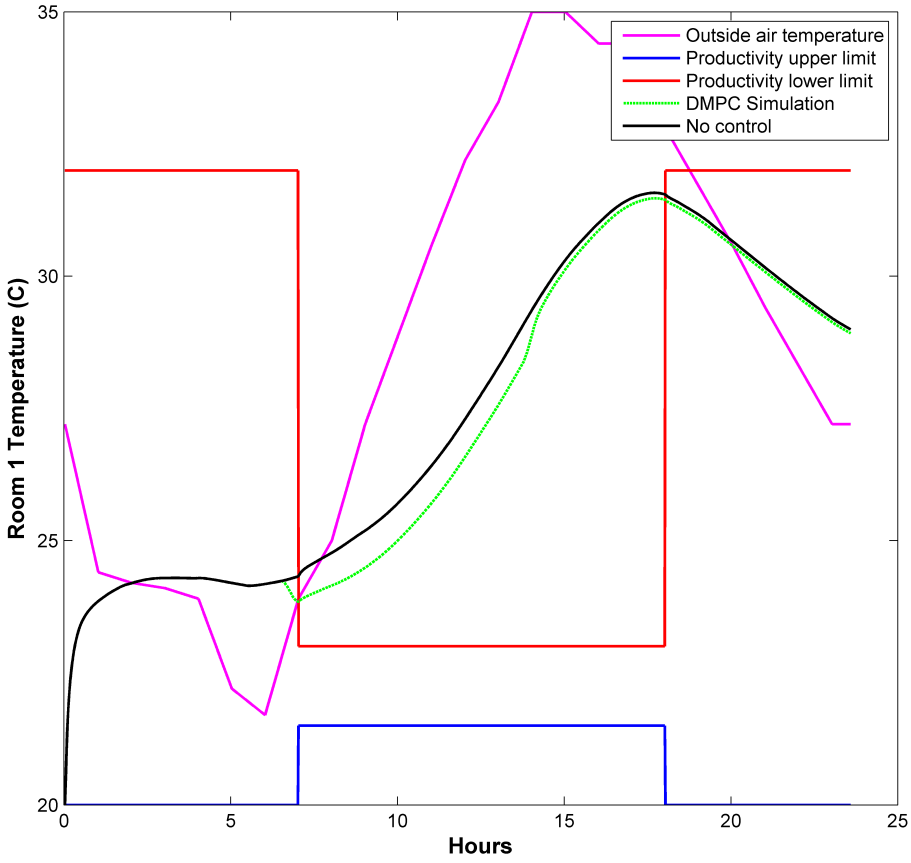


Figure 5.8: Room 1 Temperature Simulation Results with High Fan Cost

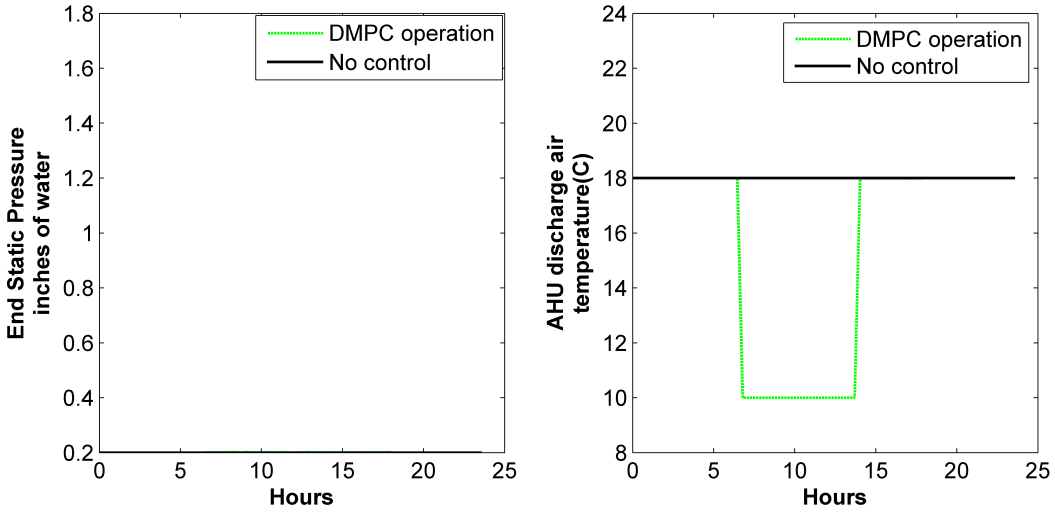


Figure 5.9: AHU End Static Pressure and Discharge Air Temperature Setpoints Computed by the NC DMPC Algorithm

5.4.2 High Fan Energy Cost

In order to simulate high fan energy cost conditions the parameter r_{fan} is increased 100 fold, and r_i^{sal} is decreased 100 times as shown in Table 4.3. Using the same conditions as those described in real-world-cost scenario, the simulations are run with the new set of parameters. The simulted temperature of room 1 under different control methodologies is shown in Figure 5.8. In addition to the DMPC and PI type a control simulations, a baseline simulation was run with no control algorithm applied. The end static pressure , and discharge air temperature, and room flow setpoints were assigned the minimum permissble values. The simulation where the no control methodology is applied is represented by the legend 'no control' in the figure. A summary of the costs achieved is shown in Table 5.9.

The AHU end static pressure and discharge air temperature setpoints corresponding to high fan cost operation is shown in Figure 5.9. A comparison of the hourly cost achieved using the various control methodologies and the cost parameters corresponding to high fan

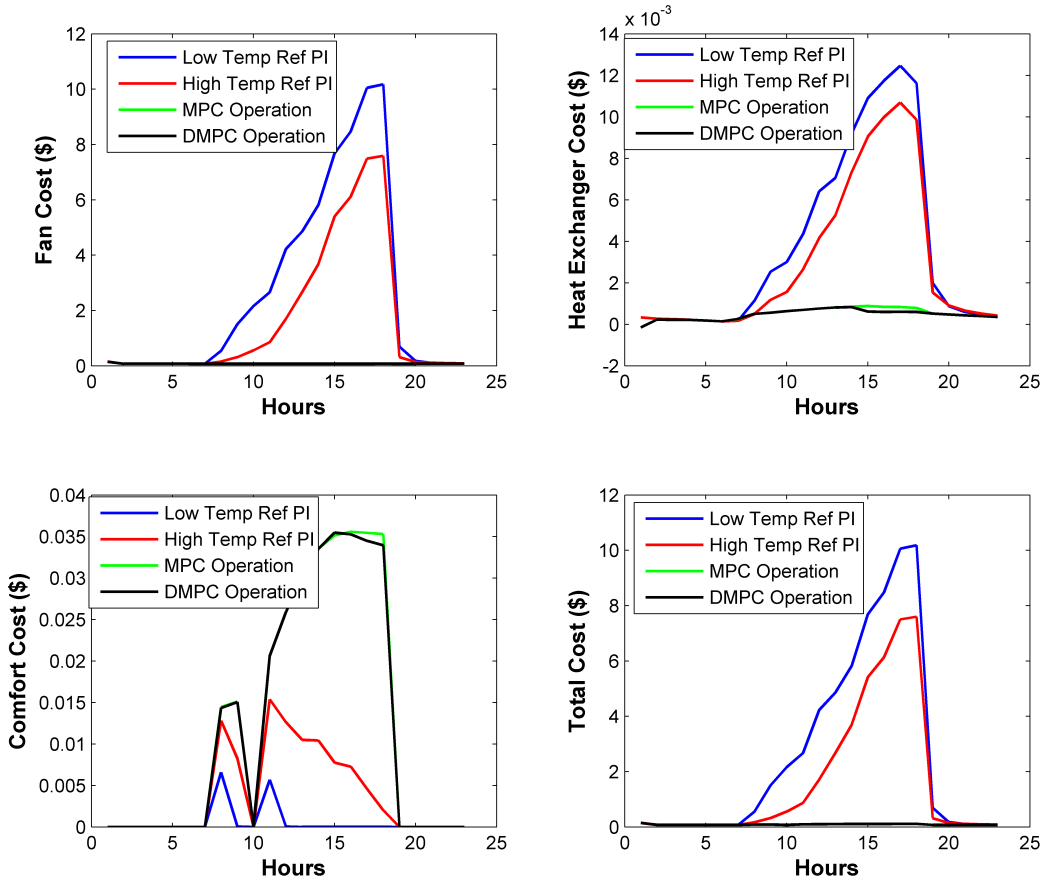


Figure 5.10: A Comparison of the Hourly Costs Achieved Using Various Control Methodologies and High Fan Energy Cost Parameters

energy operation is shown in Figure 5.10.

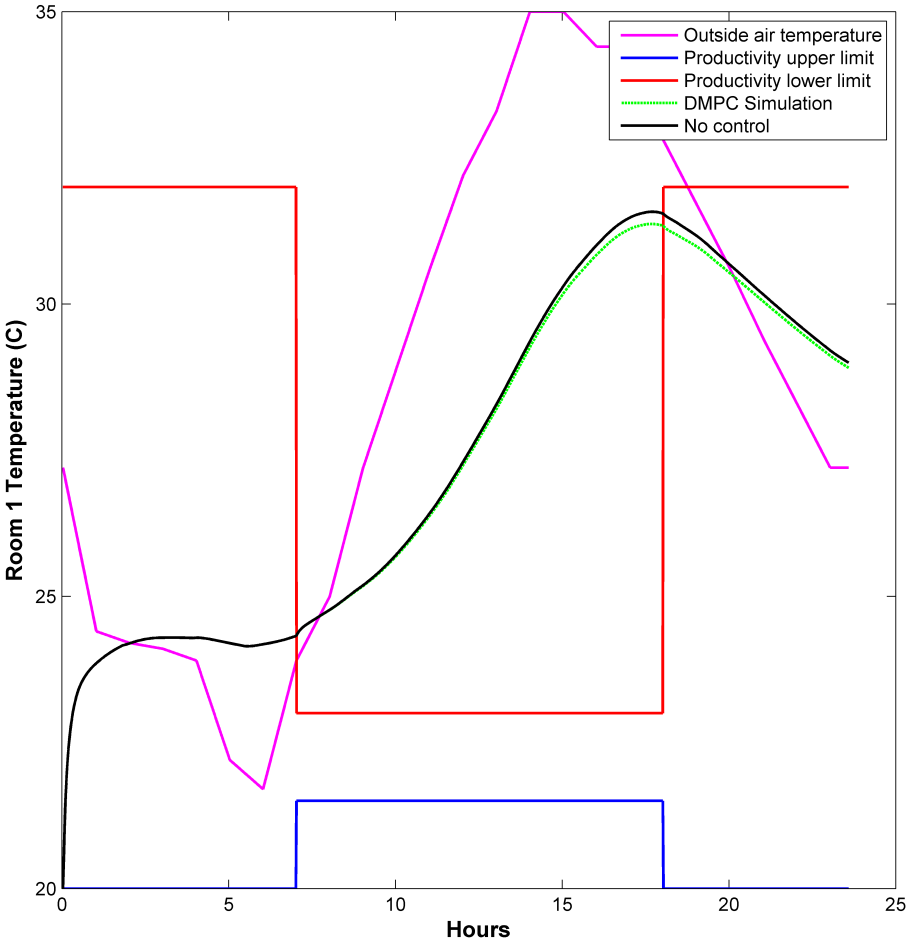


Figure 5.11: Room 1 Temperature Simulation Results with High Cooling Cost

5.4.3 High Cooling Energy Cost

The cost parameters associated with the high cooling energy cost are shown in Table 4.3. The simulations are again run under the same conditions as the previous two

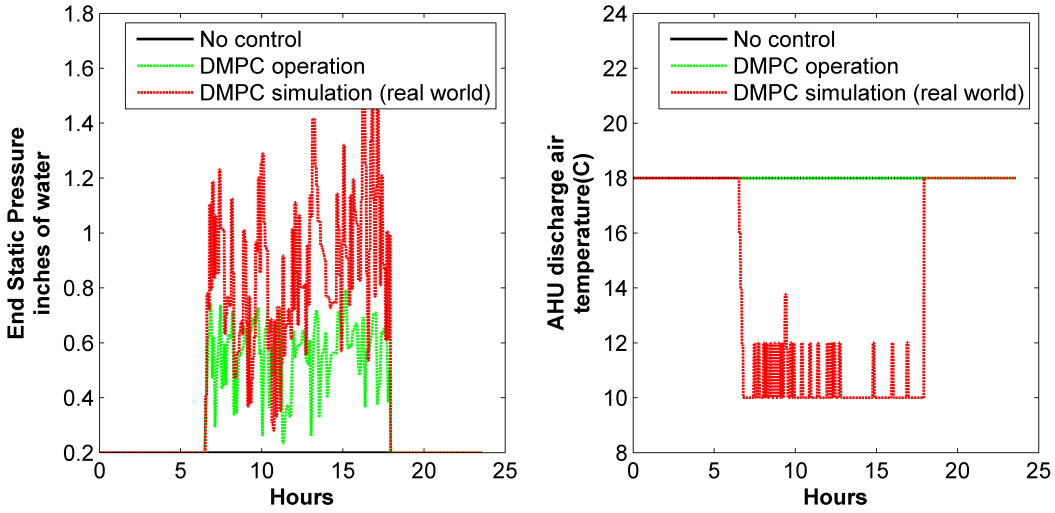


Figure 5.12: AHU End Static Pressure and Discharge Air Temperature Setpoints Computed by the NC DMPC Algorithm with High Cooling Cost

cases, but with the high cooling energy cost parameters. The simulated temperature obtained using the NC DMPC algorithm, with the defult 'no control' algorithm is shown in Figure 5.11. The AHU end static pressure and discharge air temperature computed by the NC DMPC algorithm is shown in Figure 5.12. The figure also shows a comparison with the setpoints computed by the algorithm in case 1 (real world cost parameters). Finally Figure 5.13 shows the comparison of the hourly costs associated with different control methodologies with cost parameters corresponding to high cooling energy operation. A summary of the costs achieved is shown in Table 5.10.

As expected under the high cooling cost conditions, the DMPC algorithm sets the AHU discharge air temperature to the maximum allowable value (18^0C). A high cooling cost value also affects the amount of air that passes through the AHU. A greater volume flow rate would put a greater cooling load on the heat exchanger. Hence, as can be seen in Figure 5.12, the end static pressure setpoint values corresponding to high cooling cost operation is lower than the setpoint values computed with real world cost parameters.

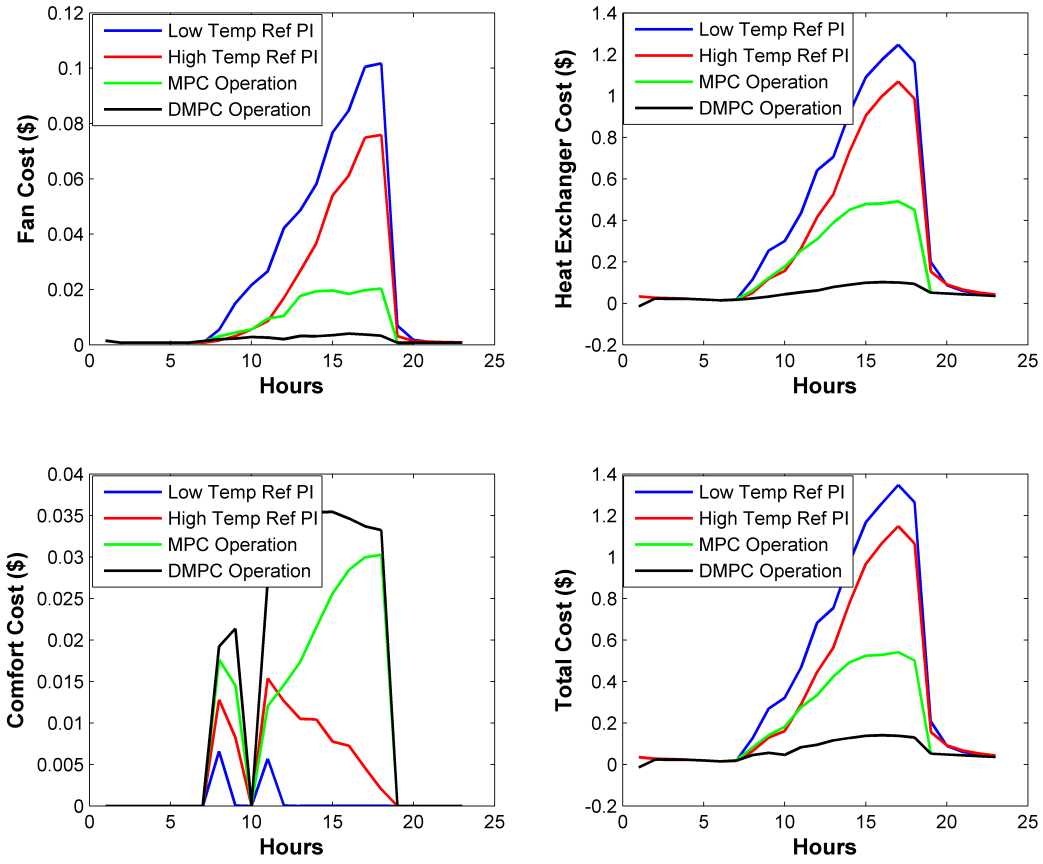


Figure 5.13: A Comparison of the Hourly Costs Achieved Using Various Control Methodologies and High Cooling Energy Cost Parameters

Whereas, high fan operation cost parameters affects only the end static pressure setpoint values, a high cooling cost operation results in higher AHU discharge air temperature setpoints as well as lower end static pressure setpoints.

Since the setpoints calculated by PI -type control is independent of the cost parameters, it is able to achieve better occupancy comfort. However, since the cost parameters were selected such that the cooling costs far outweigh the comfort cost, the NC DMPC algorithm is able to achive a much lower cost of operation. The NC DMPC algorithm is also able to achieve a lower cost of operation than centralized MPC. This is probably due to the fact that the numerical algorithm used to minimize the objective function of the centralized MPC cost did not find the global optimal solution but instead found local minimas.

6. CONCLUSIONS AND FUTURE WORK

The dissertation identified three common reasons as to why traditional control algorithms such as PI-type and RBCs result in inefficient operation of building HVAC systems, improper tuning of the gains of the PI controllers resulting in hunting behavior, decentralized architecture resulting in suboptimal performance, and the lack of planning for future disturbances and operating conditions. This dissertation seeks to address the aforementioned problems by proposing solutions in the form of alternative control architectures such as cascaded control, and optimal control algorithms such as MPC. THe contribution of the dissertation classified into three categories summarized as follows.

6.1 Algorithm to Detect the Presence of Hunting

The first step in reducing the phenomenon of hunting is to detect its presence. A simple algorithm that can detect the presence of hunting in process variables of HVAC system operation such as CHW valve opening and fan speed is proposed. The algorithm is then implemented on data from 10 buildings at the Texas A&M University campus. The results of the survey showed that hunting is a wide spread pheonomenon. The dissertation also provided a brief description of the contribution of Price [41] where a solution is offered in the form of an alternative control architecture called as cascaded control. A simulation example of the AHU heat exchanger is provided where the cascaded controller is shown to eliminate the behavior of hunting.

6.2 Automated Black-Box Modeling Algorithm

In order to be able to implement advanced control algorithms such as MPC, models that can accurately describe the dynamics of the various components involved are required. This dissertation proposes a linear black-box modeling algorithm that can automate the

process of developing models which becomes especially useful for large HVAC systems which have hundreds of components and buildings comprising multiple zones. In addition to being easy to compute, the proposed models can be developed just by using data from BEMS. The modeling algorithm is verified on data obtained from a real working office building. Static models are developed for the AHU pressure system, and dynamics models are developed for the AHU heat exchanger, and the building room temperatures. The models are able to provide accurate predictions of the process variables by using data under non experimental conditions. In addition, the proposed modeling method is compared to another black-box method popular in literature, subspace identification. No significant differences were found in the accuracy of their predictions.

6.3 Applying MPC on Simulation Models

The modeling method is then applied to high-fidelity simulation models of the building HVAC components. The developed models are then used to implement MPC on the simulation system. Comparison of the simulation results between MPC and the existing PI-type control showed that the performance of MPC is significantly greater when tested under a wide range of operating conditions. The dissertation also provides a comparison of the different control architectures by which MPC can be applied. Although, the control performance of both the distributed and centralized are comparable, the computation requirements of centralized MPC will grow exponentially as the number of components increase. Hence a distributed architecture is more suitable when dealing with large scale systems.

6.4 Future Work

There are several areas related to the research of this dissertation that have not yet been fully explored. The first area is the application of MPC on a real building system. MPC is applied in this dissertation on high-fidelity simulation models, which although

accurate do not capture all the disturbances and dynamics that a real building might be subjected to. Hence demonstrating the proposed methodology on a real building system would give building managers more confidence in implementing the procedure on a wider scale. The second area in which future work can be done is by including more components under the ambit of MPC. This dissertation considered only the AHU fan, heat exchanger and building room temperature models. By including other components such as chiller, cooling tower, room humidity etc. MPC can provide a more efficient operation.

REFERENCES

[1] U. DoE, "Buildings energy databook," *Energy Efficiency & Renewable Energy Department*, 2011.

[2] L. Perez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[3] R. Energy, "Energy efficiency trends in residential and commercial buildings," 2010.

[4] J. Siroky, F. Oldewurtel, J. Cigler, and S. Privara, "Experimental analysis of model predictive control for an energy efficient building heating system," *Applied Energy*, vol. 88, no. 9, pp. 3079–3087, 2011.

[5] R. Chintala, C. Price, S. Liang, and B. Rasmussen, "Identification and elimination of hunting behavior in hvac systems," *ASHRAE Transactions*, vol. Accepted, 2015.

[6] S. Privara, J. Cigler, Z. Vana, F. Oldewurtel, and E. Zacekova, "Use of partial least squares within the control relevant identification for buildings," *Control Engineering Practice*, vol. 21, no. 1, pp. 113–121, 2013.

[7] X. Li and J. Wen, "Review of building energy modeling for control and operation," *Renewable and Sustainable Energy Reviews*, vol. 37, no. 0, pp. 517 – 537, 2014.

[8] S. Estrada-Flores, I. Merts, B. De Ketelaere, and J. Lammertyn, "Development and validation of grey-box models for refrigeration applications: a review of key concepts," *International Journal of Refrigeration*, vol. 29, no. 6, pp. 931–946, 2006.

[9] D. B. Crawley, L. K. Lawrie, C. O. Pedersen, and F. C. Winkelmann, "Energy plus: energy simulation program," *ASHRAE journal*, vol. 42, no. 4, pp. 49–56, 2000.

[10] "Transient system simulation tool." `http://www.trnsys.com`.

[11] P. May-Ostendorp, G. P. Henze, C. D. Corbin, B. Rajagopalan, and C. Felsmann, "Model-predictive control of mixed-mode buildings with rule extraction," *Building and Environment*, vol. 46, no. 2, pp. 428–437, 2011.

[12] C. D. Corbin, G. P. Henze, and P. May-Ostendorp, "A model predictive control optimization environment for real-time commercial building application," *Journal of Building Performance Simulation*, vol. 6, no. 3, pp. 159–174, 2013.

[13] J. E. Braun and N. Chaturvedi, "An inverse gray-box model for transient building load prediction," *HVAC&R Research*, vol. 8, no. 1, pp. 73–99, 2002.

[14] J. E. Braun, "Reducing energy costs and peak electrical demand through optimal control of building thermal storage," *ASHRAE transactions*, vol. 96, no. 2, pp. 876–888, 1990.

[15] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model predictive control for the operation of building cooling systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 796–803, 2012.

[16] K.-h. Lee and J. E. Braun, "Reducing peak cooling loads through model-based control of zone temperature setpoints," in *American Control Conference, 2007. ACC'07*, pp. 5070–5075, IEEE, 2007.

[17] W. Zhang, J. Lian, C.-Y. Chang, and K. Kalsi, "Aggregated modeling and control of air conditioning loads for demand response," *IEEE transactions on power systems*, vol. 28, no. 4, pp. 4655–4664, 2013.

[18] M. Sourbron, C. Verhelst, and L. Helsen, "Building models for model predictive control of office buildings with concrete core activation," *Journal of building performance simulation*, vol. 6, no. 3, pp. 175–198, 2013.

[19] Y. Ma, A. Kelman, A. Daly, and F. Borrelli, "Predictive control for energy efficient buildings with thermal storage: Modeling, stimulation, and experiments," *IEEE Control Systems*, vol. 32, no. 1, pp. 44–64, 2012.

[20] Y. Lin, T. Middelkoop, and P. Barooah, "Identification of control-oriented thermal models of rooms in multi-room buildings," in *Proceedings of the 2012 IEEE 51st Annual Conference on Decision and Control (CDC), Maui, HI, USA*, pp. 10–13, 2012.

[21] P. Bacher and H. Madsen, "Identifying suitable models for the heat dynamics of buildings," *Energy and Buildings*, vol. 43, no. 7, pp. 1511–1522, 2011.

[22] H. Frausto, J. Pieters, and J. Deltour, "Modelling greenhouse temperature by means of auto regressive models," *Biosystems Engineering*, vol. 84, no. 2, pp. 147 – 157, 2003.

[23] D. Loveday and C. Craggs, "Stochastic modelling of temperatures for a full-scale occupied building zone subject to natural random influences," *Applied Energy*, vol. 45, no. 4, pp. 295–312, 1993.

[24] J. Boaventura Cunha, C. Couto, and A. Ruano, "Real-time parameter estimation of dynamic temperature models for greenhouse environmental control," *Control Engineering Practice*, vol. 5, no. 10, pp. 1473–1481, 1997.

[25] S. Wang and X. Xu, "Simplified building model for transient thermal performance estimation using ga-based parameter identification," *International Journal of Thermal Sciences*, vol. 45, no. 4, pp. 419–432, 2006.

[26] G. Mustafaraj, J. Chen, and G. Lowry, "Development of room temperature and relative humidity linear parametric models for an open office using bms data," *Energy and Buildings*, vol. 42, no. 3, pp. 348–356, 2010.

[27] T. Bohlin and S. F. Graebe, "Issues in nonlinear stochastic grey box identification," *International journal of adaptive control and signal processing*, vol. 9, no. 6, pp. 465–490, 1995.

[28] R. Z. Homod, "Review on the hvac system modeling types and the shortcomings of their application," *Journal of Energy*, vol. 2013, 2013.

[29] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.

[30] W. Grunenfelder and J. Todtli, "The use of weather predictions and dynamic programming in the control of solar domestic hot water systems," in *3rd Mediterranean Electrotechnical Conference (Melecon). Madrid, Spain*, 1985.

[31] G. P. Henze, C. Felsmann, and G. Knabe, "Evaluation of optimal control for active and passive building thermal storage," *International Journal of Thermal Sciences*, vol. 43, no. 2, pp. 173–183, 2004.

[32] G. P. Henze, D. E. Kalz, C. Felsmann, and G. Knabe, "Impact of forecasting accuracy on predictive optimal control of active and passive building thermal storage inventory," *HVAC&R Research*, vol. 10, no. 2, pp. 153–178, 2004.

[33] F. Oldewurtel, D. Gyalistras, M. Gwerder, C. Jones, A. Parisio, V. Stauch, B. Lehmann, and M. Morari, "Increasing energy efficiency in building climate control using weather forecasts and model predictive control," in *Clima-RHEVA World Congress*, no. EPFL-CONF-169735, 2010.

[34] D. Sturzenegger, D. Gyalistras, M. Gwerder, C. Sagerschnig, M. Morari, and R. S. Smith, "Model predictive control of a swiss office building," in *Clima-RHEVA World Congress*, pp. 3227–3236, 2013.

[35] P. Haves, "Model predictive control of hvac systems: Implementation and testing at the university of california, merced," *Lawrence Berkeley National Laboratory*, 2010.

[36] R. Scattolini, "Architectures for distributed and hierarchical model predictive control–a review," *Journal of process control*, vol. 19, no. 5, pp. 723–731, 2009.

[37] P.-D. Moroşan, R. Bourdais, D. Dumur, and J. Buisson, "Building temperature regulation using a distributed model predictive control," *Energy and Buildings*, vol. 42, no. 9, pp. 1445–1452, 2010.

[38] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Distributed model predictive control of large-scale systems," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, pp. 591–605, Springer, 2007.

[39] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, "Cooperative distributed model predictive control," *Systems & Control Letters*, vol. 59, no. 8, pp. 460–469, 2010.

[40] M. Elliott and B. P. Rasmussen, "Optimal setpoints for hvac systems via iterative cooperative neighbor communication," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 1, p. 011006, 2015.

[41] C. Price, P. Liang, Shuangshuang, and P. P. Rasmussen, Bryan, "Hvac nonlinearity compensation using cascaded control architectures," *ASHRAE Transactions*, vol. 121, pp. 217–231, 2015.

[42] A. Singhal and T. I. Salsbury, "A simple method for detecting valve stiction in oscillating control loops," *Journal of Process Control*, vol. 15, no. 4, pp. 371 – 382, 2005.

[43] T. Hagglund, "A control-loop performance monitor," *Control Engineering Practice*, vol. 3, no. 11, pp. 1543–1551, 1995.

[44] N. F. Thornhill and T. Hägglund, "Detection and diagnosis of oscillation in control loops," *Control Engineering Practice*, vol. 5, no. 10, pp. 1343–1354, 1997.

[45] T. Miao and D. E. Seborg, "Automatic detection of excessively oscillatory feedback control loops," in *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, vol. 1, pp. 359–364, IEEE, 1999.

[46] N. F. Thornhill, B. Huang, and H. Zhang, "Detection of multiple oscillations in control loops," *Journal of Process Control*, vol. 13, no. 1, pp. 91–100, 2003.

[47] M. S. Elliott and B. P. Rasmussen, "On reducing evaporator superheat nonlinearity with control architecture," *international journal of refrigeration*, vol. 33, no. 3, pp. 607–614, 2010.

[48] X. Zhou and J. E. Braun, "A simplified dynamic model for chilled-water cooling and dehumidifying coilsâĂŤpart 1: Development (rp-1194)," *HVAC&R Research*, vol. 13, no. 5, pp. 785–804, 2007.

[49] D. R. Wulfinghoff, *Energy efficiency manual*, vol. 3936.

[50] R. H. Chintala and B. P. Rasmussen, "Automated multi-zone linear parametric black-box modeling approach for building hvac systems," 2015.

[51] L. Ljung, "System identification: theory for the user," *PTR Prentice Hall Information and System Sciences Series*, 1999.

[52] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory-Implementation-Applications*. Springer Science & Business Media, 2012.

[53] S. J. Qin, "An overview of subspace identification," *Computers & chemical engineering*, vol. 30, no. 10, pp. 1502–1513, 2006.

[54] H. U. Frausto, J. Pieters, and J. Deltour, "Modelling greenhouse temperature by means of auto regressive models," *Biosystems Engineering*, vol. 84, no. 2, pp. 147–157, 2003.

[55] S. Wang and X. Jin, "Model-based optimal control of vav air-conditioning system using genetic algorithm," *Building and Environment*, vol. 35, no. 6, pp. 471–487, 2000.

[56] E. Toffoli, G. Galdan, G. Albertin, L. Schenato, A. Chiuso, and A. Beghi, "Thermodynamic identification of buildings using wireless sensor networks," *Proceedings of the 17th IFAC World Congress*, vol. 17, no. 1, pp. 8860–8865, 2008.

[57] L. H. Hansen, *Stochastic modelling of central heating systems*. PhD thesis, Technical University of DenmarkDanmarks Tekniske Universitet, Department of Informatics and Mathematical ModelingInstitut for Informatik og Matematisk Modellering, 1997.

[58] D. Wyon, "Indoor environmental effects on productivity," in *Paths to better building environments*, 1997.

[59] R. Kosonen and F. Tan, "Assessment of productivity loss in air-conditioned buildings using pmv index," *Energy and Buildings*, vol. 36, no. 10, pp. 987–993, 2004.

[60] P. O. Fanger *et al.*, "Thermal comfort. analysis and applications in environmental engineering.," *Thermal comfort. Analysis and applications in environmental engineering.*, 1970.

[61] M. S. Elliott, C. J. Bay, and B. P. Rasmussen, "Pareto optimal setpoints for hvac networks via iterative nearest neighbor communication," in *ASME 2013 Dynamic Systems and Control Conference*, pp. V001T13A004–V001T13A004, American Society of Mechanical Engineers, 2013.