© 2013 by Vikas Chandan. All rights reserved

DECENTRALIZED THERMAL CONTROL OF BUILDING SYSTEMS

 $\mathbf{B}\mathbf{Y}$

VIKAS CHANDAN

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Doctoral Committee:

Professor Andrew Alleyne, Chair Professor Geir Dullerud Professor Dušan Stipanović Dr. John Seem

Abstract

Energy requirements for heating and cooling of buildings constitute a major fraction of end use energy consumed. Therefore, it is important to provide the occupant comfort requirements in buildings in an energy efficient manner. However, buildings are large scale complex systems, susceptible to sensor, actuator or communication network failures in their thermal control infrastructure, that can affect their performance in terms of occupant comfort and energy efficiency. The degree of decentralization in the control architecture determines a fundamental tradeoff between performance and robustness. This thesis studies the problem of thermal control of buildings from the perspective of partitioning them into clusters for decentralized control, to balance underlying performance and robustness requirements. Measures of deviation in performance and robustness between centralized and decentralized architectures in the Model Predictive Control framework are derived. Appropriate clustering algorithms are then proposed to determine decentralized control architectures which provide a satisfactory trade-off between the underlying performance and robustness objectives. Two different partitioning methodologies - the CLF-MCS method and the OLF-FPM method - are developed and compared. The problem of decentralized control design based on the architectures obtained using these methodologies is also considered. It entails the use of decentralized extended state observers to address the issue of unavailability of unknown states and disturbances in the system. The potential use of the proposed control architecture selection and decentralized control design methodologies is demonstrated in simulation on a real world multi-zone building.

To my parents.

Acknowledgments

This thesis would be incomplete without expressing my gratitude towards all the people who have directly or indirectly helped me in the course of my graduate study at the University of Illinois. It starts with Prof. Andrew Alleyne, my research advisor. I am extremely grateful to him for providing me the opportunity to be a part of the Alleyne Research Group (ARG) since January 2008. Through the course of the last five years within the ARG, I had the opportunity to work on some challenging and practical research problems in the area of control and modeling of energy systems. This thesis is a concatenation of those research efforts. I am very obliged to Prof. Alleyne for his time and efforts in directing my research despite his busy schedule. Throughout these years, he has helped me metamorphose from an undergraduate newcomer to the field of research to a serious graduate student capable of taking ownership of research directions and progress. Besides being an adviser to my research, I am also thankful to him for the role he has played in several other aspects of my overall professional development. In particular, his emphasis on leadership, collaboration and communication has helped me throughout my stint as a graduate student and I am confident that these skills would prove beneficial in the course of my future professional career. In addition to these, I am also thankful to him for helping me make a few critical career decisions, keeping in mind my future professional goals.

Next, I would like to thank the members of my PhD preliminary and defense examination committees – Prof. Geir Dullerud, Prof. Dusan Stipanovic, Prof. Prashant Mehta and Dr. John Seem – for their time and effort in evaluating my research and providing me valuable feedback. I am especially grateful to Dr. John Seem for the discussions sessions that I had with him which helped me in improving the quality and the practical appeal of my work. I would also like to express my gratitude to all the professors at Illinois who provided me the opportunity to learn

through interactions both within and outside the classroom. The coursework that I undertook in the area of controls introduced me to several important concepts and tools that I found useful for my research at various points in time. These courses have also enabled me to a build fundamental understanding of controls which I am confident would help me through the course of my research career. I would also like to acknowledge the technical assistance provided by Prasad Mukka at Siemens Corporate Research, Princeton, NJ in the development of the EnergyPlus model that was used in Chapter 6 for the validation and verification of the tools presented in this thesis.

Lastly, I would like to thank all my colleagues within ARG – both past and present – with whom I had the opportunity to work with and interact very closely. In particular, I would like to thank Neera, Tim, Dave, Kira, Bin, Nanjun, Yangmin and Erick with whom I had the privilege to interact most closely during my time at the ARG. I would also like to thank some other members of the group – Brian, Scott, Serena, Doug, Gina, Brandon, Rich, Tom, Justin, Joey, Megan, Kasper and Matt – with whom I shared shorter but valuable stints. Finally, I would like to conclude by thanking my family members whose love and support was critical in keeping me motivated and strong throughout my journey towards a PhD.

Table of Contents

CHAPTER 1	INTRODUCTION	1
1.1 Motiv	vation and background	1
1.2 Resea	arch objectives	4
1.2.1	Control architecture selection	
1.2.2 1.2.3	Decentralized control design Evaluation of proposed tools and methodologies	
1.3 Liter	ature survey	7
1.3.1 1.3.2 1.3.3	Resources for building systems research Modeling and simulation of building thermal dynamics Thermal control of buildings	
1.4 Outli	ne of the thesis	
CHAPTER 2	PHYSICAL DETAILS OF BUILDING SYSTEMS	14
2.1 Introd	duction	14
2.2 Contr	rol systems in a building	14
2.2.1	Thermal systems	14
2.2.2	Lighting systems	
2.2.3	Security systems	13
2.3 VAV	systems	
2.3.1	Architecture	
2.3.2	Thermal control	
CHAPTER 3	CENTRALIZED AND DECENTRALIZED CONTROL	
FRAMEWOR	KS	
3.1 There	nal control of buildings	
3.1.1	Control objectives	
3.1.2	Control aspects	
3.2 Prelin	ninaries	

3.3 Centralized control framework	
3.3.1 Architecture	
3.3.2 Objective function	
3.3.3 Model	
3.3.4 Conversion to Quadratic Program	33
3.4 Decentralized control framework	
3.4.1 Architecture	
3.4.2 Objective function	
3.4.3 Model	
3.4.4 Conversion to Quadratic Program	
CHAPTER 4 CONTROL ARCHITECTURE SELECTION FOR BUILDING	40
THERMAL CONTROL	
4.1 Introduction	42
4.2 CLF-MCS approach	42
4.2.1 Overview	42
4.2.2 Coupling between inputs and clusters of inputs	
4.2.3 Overview of divisive clustering approach	
4.2.4 Optimality and robustness metrics	
4.2.5 Stage level combinatorial optimization	
4.2.6 MINCUT approximation	
4.2.7 Optimal partition selection	
4.2.8 Twelve zone building example	
4.2.9 Nine zone building example	
4.2.10 Remarks	
4.3 OLF-FPM method	59
4.3.1 Overview	59
4.3.2 Optimality analysis for centralized MPC	59
4.3.3 Extraction matrices for decentralized MPC	61
4.3.4 Optimality analysis for decentralized MPC	64
4.3.5 Comparison of centralized and decentralized MPC	
4.3.6 Optimality metric	
4.3.7 Robustness metric	
4.3.8 Optimal partitioning problem and complexity analysis	
4.3.9 Agglomerative clustering	
4.3.10 Nine zone building example	
4.3.11 Eleven zone circular building	
4.3.12 Kemarks	87
CHAPTER 5 CONTROL DESIGN WITH OPTIMAL ARCHITECTURE	88
5.1 Introduction	88

5.2 Cluster level modeling	
5.2.1 3R2C modeling framework5.2.2 Full order cluster level model5.2.3 Model order reduction	90 93 94 99 99
5.3 Observer and controller design	
5.3.1 Observer design5.3.2 Controller design	
5.4 Optimization	
5.4.1 Re-statement of objective function5.4.2 Constraints5.4.3 Quadratic program formulation	
5.5 Concluding remarks	
CHAPTER 6 REAL WORLD BUILDING SIMULATION S	STUDY 121
6.1 Introduction	
6.2 Test building and EnergyPlus model	
 6.2.1 Building Layout and Geometry 6.2.2 Construction data 6.2.3 Weather data 6.2.4 Internal loads 6.2.5 Creation of EnergyPlus model 	123 127 128 129 132
6.3 Generation of linearized model	
6.3.1 Overview of modeling framework6.3.2 Zone level identification6.3.3 Wall surface level identification6.3.4 Construction of LTI model	133 133 143 153
6.4 Control architecture determination	
6.4.1 Modifications to LTI model	
6.5 Control design and analysis	
6.5.1 Control design6.5.2 Closed loop performance assessment	
6.6 Concluding remarks	
CHAPTER 7 CONCLUSIONS	
7.1 Summary	
7.2 Conclusions	

7.3 Contributions	180
7.4 Future extensions	181
7.4.1 Incorporation of HVAC system	181
7.4.2 Experimental investigation	181
7.4.3 Extension to other applications	182
References	183
Appendix A: Codes for 12-zone building example in section 4.2.8	190
Appendix B: Codes for 9-zone building example in section 4.2.9	197
Appendix C: Codes for 9-zone building example in section 4.3.10	210
Appendix D: Codes for 11-zone building example in section 4.3.11	222
Appendix E: Steps used in creating an EnergyPlus model using OpenStudio	229
Appendix F: Codes for performing zone level identification as per optimization framework presented in (6.3) - (6.6)	231
Appendix G: Codes for performing zone level identification as per optimization framework presented in (6.7) - (6.10)	234
Appendix H: Codes for performing wall level identification as per optimization framework presented in section 6.3.3.1	237
Appendix I: Codes for performing wall level identification as per optimization framework presented in section 6.3.3.2.	239
Appendix J: Codes to obtain LTI model of SCR building and perform agglomeratic clustering (section 6.3.4 - 6.4).	ve 242
Appendix K: Codes for cluster level control design (Fig. 6.30)	246
Appendix L: Codes for performing the optimality analysis (section 6.5.2.2)	260
Appendix M: Codes for performing the robustness analysis (section 6.5.2.3)	272

List of Tables

Table 1.1 An overview of control-oriented building simulation programs	11
Table 3.1 Nomenclature of common symbols in Chapter 3	21
Table 4.1 Construction properties used for the 12-zone building model	52
Table 4.2 Resistances and capacitances of walls	52
Table 4.3 Summary of results for 12-zone test building	53
Table 4.4 Nomenclature of additional symbols used (besides Table 3.1)	59
Table 4.5 FPM computation for 3-zone building in Figure 3.3	73
Table 4.6 Optimal partitions vs. partitions using agglomeration for 9-zone building	79
Table 4.7 Resistance and capacitance values for walls of 11-zone building	81
Table 4.8 Zone capacitances for 11-zone building	82
Table 4.9 Partitions using agglomeration for 11-zone building	83
Table 5.1 Nomenclature of common symbols in Chapter 5	89
Table 6.1 Nomenclature of symbols used in Chapter 6	121
Table 6.2 Description of thermal zones in the test office building	127
Table 6.3 Details of construction template used for the test office building	128
Table 6.4 Nominal occupancy, lighting load and equipment load information used in Energ	yPlus
model of test building	130
Table 6.5 Results of zone level optimization for all zones in test building	138
Table 6.6 \bar{C} and \bar{R} values for various wall types in the test building	154
Table 6.7 Thermal resistances due to wall opening between each pair of adjacent zones on	each
floor of the test building	156
Table 6.8 Partitions using agglomeration for test building	157
Table 6.9 Parameters used in designing controllers for each cluster i	160
Table 6.10 Maximum supply air mass flow rate and reheat power available to each zone	160
Table 6.11 Optimality analysis for test building under various control architectures	169
Table 6.12 Closed loop robustness analysis for test building for various control architecture	es in
the event of sensor failure in zone 1	173

List of Figures

Figure 1.1 Energy consumption by end use [3]	1
Figure 1.2 End-use energy consumption in commercial buildings [4]	2
Figure 1.3 End use energy consumption in residential buildings [5]	2
Figure 1.4 Illustration of centralized and decentralized architectures for thermal control of a	
multi-zone building	4
Figure 2.1 Illustration of a rule based lighting control system (Source [64])	16
Figure 2.2 Illustration of components in an automated building security system (Source:[66])	17
Figure 2.3 Architecture of a VAV system (source: [68])	18
Figure 2.4 Illustration of an air-handling unit (source: [69])	18
Figure 2.5 Illustration of a VAV terminal unit (damper and reheat coils are not shown, source:	
Wikipedia)	19
Figure 2.6 Illustration of a thermostat (source: [72])	20
Figure 2.7 Illustration of a digital VAV controller (Source: [73])	20
Figure 3.1 Six zone building used in the case study (return and supply water lines for chilled	
water loop also shown)	24
Figure 3.2 Effect of sensor failure in room 1 of building shown in Fig. 3.1	25
Figure 3.3 An example 3-zone building (bottom surface of each zone faces ground – all other	
external surfaces are exposed to ambient)	27
Figure 3.4 Schematic of centralized MPC architecture	27
Figure 3.5 Building block for RC network model of thermal interactions. Figure represents an	
internal wall flanked by zones on either side of it.	30
Figure 3.6 Graph representation for the 3-zone building in Fig. 3.3. Nodes 1-10 represent walls	s,
11-13 are zones, 14 corresponds to ambient and 15 corresponds to ground	32
Figure 3.7 Schematic of decentralized control architecture	37
Figure 4.1 Overview of the CLF-MCS clustering procedure	46
Figure 4.2 Schematic of combinatorial optimization process for any given stage	47
Figure 4.3 Illustrative example of CLF vs. MCS plot.	50
Figure 4.4 12-zone test building architecture (Zones are numbered)	51
Figure 4.5 CLF vs. MCS plot for case 1 in Table 4.3	53
Figure 4.6 CLF vs. MCS plot for case 2 in Table 4.3	54
Figure 4.7 CLF vs. MCS plot for case 3 in Table 4.3	54
Figure 4.8 CLF vs. MCS plot for case 4 in Table 4.3	55
Figure 4.9 Layout of 9-zone building (side view)	56
Figure 4.10 CLF vs. MCS plot for nine-zone building with $\rho = 1$	57
Figure 4.11 CLF vs. MCS plot for nine-zone building with $\rho = 2$	57
Figure 4.12 CLF vs. MCS plot for nine-zone building with $\rho = 3$	58
Figure 4.13 Scalar illustration of deviation in performance between centralized and decentraliz	ed

MPC when $\overline{\widehat{\mathbf{T}}}_{\mathbf{z}}^* \neq \overline{\widehat{\mathbf{T}}}_{\mathbf{z}}$	71
Figure 4.14 Illustration of agglomerative clustering for a building with 5 zones	75
Figure 4.15 Computational complexity comparison of partitioning approaches	76
Figure 4.16 Illustration of optimality-robustness trade-off curve for OLF-FPM approach	77
Figure 4.17 OLF comparison of all p-partitions (solid circles) of the 9-zone building with	
agglomerative partitions (empty circles). The agglomerative clustering progresses from l	eft
to right, starting with the most decentralized partition $(p = 9)$	79
Figure 4.18 Optimality robustness trade-off <i>curves</i> for 9-zone building using true optimal	
partitions and agglomerative partitions	80
Figure 4.19 Layout of 11-zone circular building (plan view with building height = 15 ft)	81
Figure 4.20 Optimality robustness trade-off curve for 11-zone building	84
Figure 4.21 Plot of J _{partition} for partitions from agglomeration for 11-zone building	84
Figure 4.22 Open loop response analysis for partitions of 11-zone building	86
Figure 4.23 Sinusoidal disturbance profiles used in open loop analysis of 11-zone building	87
Figure 5.1 Schematic of 3R2C modeling paradigm for wall <i>i</i>	91
Figure 5.2 Schematic of 3-zone building used for illustration of Algorithm 5.2	97
Figure 5.3 Reduced order RC network obtained via Algorithm 5.2 for the cluster $\{1,2\}$ of the	3-
zone building in Figure 5.2	99
Figure 5.4 Illustration of controller and observer for i^{in} cluster	109
Figure 6.1 Photograph of the SCR building (source: $[93]$)	124
Figure 6.2 Google SketchUp illustration of the test office building	124
Figure 6.5 Top (plan) view of test office building.	125
Figure 6.4 Isolitettic view of the test office building viewed from the North East direction	125
Figure 6.6 Transverse cut of basement showing conditioned and non-conditioned sections of t	120
office building	126
Figure 6.7 Transverse cut of D block showing the server room of test office building	120
Figure 6.8 Weekday occupancy schedule in each thermal zone in the test building	120
Figure 6.9 Saturday's occupancy schedule in each thermal zone of the test building	131
Figure 6.10 Weekday lighting schedule in each thermal zone in the test building	131
Figure 6.11 Saturday's lighting schedule in each thermal zone in the test building	132
Figure 6.12 Schematic of 3R2C modeling paradigm for wall <i>i</i>	133
Figure 6.13 Zone set point temperature signal from 12:00 AM to 1:00 AM used in EnergyPlus	s
simulations for zone level identification	135
Figure 6.14 Temperature of zone F2 from zone level least squares identification methodology	r
compared with EnergyPlus data between 12:00 AM to 1:00 AM	136
Figure 6.15 Temperature of zone G from zone level least squares identification methodology	
compared with EnergyPlus data between 12:00 AM to 1:00 AM	139
Figure 6.16 Temperature of zone E from zone level least squares identification methodology	
compared with EnergyPlus data between 12:00 AM to 1:00 AM	139
Figure 6.17 Temperature of zone C from zone level least squares identification methodology	
compared with EnergyPlus data between 12:00 AM to 1:00 AM	140
Figure 6.18 Temperature of zone SR from zone level least squares identification methodology	Ý

compared with EnergyPlus data between 12:00 AM to 1:00 AM	140
Figure 6.19 Temperature of zone F1 from zone level least squares identification methodology	7
compared with EnergyPlus data between 12:00 AM to 1:00 AM	141
Figure 6.20 Temperature of zone D from zone level least squares identification methodology	
compared with EnergyPlus data between 12:00 AM to 1:00 AM	141
Figure 6.21 Temperature of zone F2 from zone level least squares identification methodology	/
compared with EnergyPlus data between 12:00 AM to 1:00 AM	142
Figure 6.22 Temperature of zone TCB from zone level least squares identification methodolo	ogy
compared with EnergyPlus data between 12:00 AM to 1:00 AM	142
Figure 6.23 Temperature of zone NTCB from zone level least squares identification	
methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM	143
Figure 6.24 Inside surface temperature of wall 34 from least squares identification methodole	ogy
(section 6.3.3.1) compared with EnergyPlus data ($R_{ww,34,max} = 100 \text{ K/kW}$)	148
Figure 6.25 Inside surface temperature of wall 34 from least squares identification methodole	ogy
(section 6.3.3.1) compared with EnergyPlus data ($R_{ww,34,max} = 40 \text{ K/kW}$)	148
Figure 6.26 Inside surface temperature of wall 34 from least squares identification methodole	ogy
(section 6.3.3.1) compared with EnergyPlus data ($R_{ww,34,max} = 10 \text{ K/kW}$)	149
Figure 6.27 Inside surface temperature of wall 34 from least squares identification methodolo)gy
(section 6.3.3.2) compared with EnergyPlus data	151
Figure 6.28 Illustration of wall opening present between thermal zones E and D in the test	
building	155
Figure 6.29 Optimality-robustness tradeoff curve for test building (p denotes the number of	
clusters in a partition)	156
Figure 6.30 Steps used for cluster level control design	159
Figure 6.31 Ambient temperature obtained from EnergyPlus weather file for the simulation ti	me
	162
Figure 6.32 Set-point temperatures for all zones during the simulation time window	163
Figure 6.33 Evolution of zone temperatures between 12:00 AM to 1:00 AM for centralized	100
Eigure 6.24 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized	103
architecture corresponding to know 2	161
Figure 6.35 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized	104
architecture corresponding to knee 2	164
Figure 6.36 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized	104
architecture corresponding to knee 1	165
Figure 6 37 Evolution of zone temperatures between 12:00 AM to 1:00 AM for fully	105
decentralized architecture	165
Figure 6.38 Control inputs between 12:00 AM to 1:00 AM for centralized architecture	166
Figure 6.39 Control inputs between 12:00 AM to 1:00 AM for decentralized architecture	100
corresponding to knee 3	166
Figure 6.40 Control inputs between 12:00 AM to 1:00 AM for decentralized architecture	
corresponding to knee 2	167
Figure 6.41 Control inputs between 12:00 AM to 1:00 AM for decentralized architecture	-
corresponding to knee 1	167
-	

Figure 6.42 Control inputs between 12:00 AM to 1:00 AM for fully decentralized architecture
Figure 6.43 Evolution of zone temperatures between 12:00 AM to 1:00 AM for centralized
architecture in the event of sensor failure in zone 1 169
Figure 6.44 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized
architecture corresponding to knee 3 in the event of sensor failure in zone 1 170
Figure 6.45 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized
architecture corresponding to knee 2 in the event of sensor failure in zone 1 170
Figure 6.46 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized
architecture corresponding to knee 1 in the event of sensor failure in zone 1
Figure 6.47 Evolution of zone temperatures between 12:00 AM to 1:00 AM for fully
decentralized architecture in the event of sensor failure in zone 1
Figure 6.48 Regulation errors evaluated at the end of 1 hour for all zones in the building under
various control architectures in the event of sensor failure in zone 1 (zone numbers
Figure 6.40 Control input for zone 1 between 12:00 AM to 1:00 AM for decontrolized
architecture corresponding to knog 2 in the event of songer failure in zone 1
Figure 6.50 Control input for zone 4 (lying in same cluster as zone 3) between 12:00 AM to 1:00
AM for decentralized architecture corresponding to knee 3 in the event of sensor failure in
zone 1
Figure 6 51 Control input for zone 6 (lying in same cluster as zone 1) between 12:00 AM to 1:00
AM for decentralized architecture corresponding to knee 3 in the event of sensor failure in
zone 1
Figure 6.52 Control input for zone 2 (in different cluster from zone 1) between 12:00 AM to 1:00
AM for decentralized architecture corresponding to knee 3 in the event of sensor failure in
zone 1

Chapter 1 Introduction

1.1 Motivation and background

In recent times, there has been an increased emphasis, both nationally and internationally on the importance of efficient utilization of energy [1, 2]. It has primarily been driven by concerns related to environmental, economic and sustainability aspects associated with energy. The impact of climate change and the rapid depletion of non-renewable natural resources is more visible today than at any time in the past. Together with a greater emphasis or renewable and non-polluting sources of energy, efficient use of energy can help to mitigate these effects. There are also substantial economic benefits associated with lesser energy consumption in the form of reduced costs both for energy suppliers and consumers. Reduction in energy demand leads to lower utility expenses at the consumers' end. Similarly, it translates into lesser energy supply and therefore reduced capital and operating costs at the suppliers' end.



Figure 1.1 Energy consumption by end use [3]



Figure 1.2 End-use energy consumption in commercial buildings [4]



Figure 1.3 End use energy consumption in residential buildings [5]

The importance of efficient energy usage motivates a strong emphasis on sectors which account for a large fraction of energy consumption, in order to create a meaningful impact on the global energy and emissions scenario. Figure 1.1 shows sector-wise statistics on end use energy consumption in the United States. The buildings sector is important because it accounts for more than 40% of the total energy consumption and a similar share of greenhouse gas emissions in the United States [3]. In most buildings, more than one-third of the energy usage is attributed to space heating and cooling (Figure 1.2 and Figure 1.3). Therefore, improvements related to energy efficiency in building thermal management can significantly impact the utilization, costs and environmental sustainability aspects of the overall energy consumption.

The problem of efficient thermal management in buildings is inherently multidisciplinary and presents diverse opportunities for several different areas of technology such as design, architecture, alternative energy, modeling and control. In this regard, the opportunities offered by the field of controls are particularly important for existing buildings where re-modeling and design retrofits may be infeasible due to engineering or economic reasons. Strong arguments for energy efficiency in the existing buildings stock [6] have recently been made, therefore underlining the usefulness of controls in achieving such goals. Hence, control of the building heating, ventilation and air-conditioning (HVAC) systems for energy efficient operation has received considerable attention [7, 8, 9].

The underlying control objectives in the context of building thermal management are manifold. The primary objective is to achieve the thermal demands corresponding to the various zones in the building, which are specified by desired levels of temperature, humidity and other indexes of occupant thermal comfort. As discussed above, another important objective is to achieve these thermal demands in an energy efficient manner, which can lead to reduced energy consumption, equipment operating costs and emissions. It is also desired to satisfy the thermal demands robustly, meaning that the control design and architecture should ensure resilience against failures such as thermostat malfunction. Lastly, the controllers should preferably be easily tunable and scalable when going from one building to another.

Two key aspects associated with the control of large scale complex systems such as buildings are control architecture and control algorithm. This thesis intends to analyze the role of these aspects and address them appropriately with the aim of achieving the objectives in building thermal control mentioned above. The outcome is a set of modeling and control tools that appear promising when subjected to detailed simulation studies to examine their efficacy in meeting these objectives. The next section describes the specific objectives of this thesis in more detail.

1.2 Research objectives

The primary objective of this research, as mentioned in the previous section is to aid the development of novel and promising modeling and control tools capable of satisfactorily addressing the underlying objectives in the thermal control of buildings. The specific research objectives pursued in this thesis are described below.

1.2.1 Control architecture selection

From a systems engineering perspective, buildings are multi-time scale, complex systems with multiple states, inputs, outputs and disturbances. For such systems, the closed loop performance is affected by the choice of the control architecture. In theory, a centralized controller (see illustration in Figure 1.4) using complete information of the system dynamics, and having access to building-wide sensory data could control the building optimally, i.e. satisfy the thermal comfort requirements in the various zones of the building with the least energy consumption. In this framework, control decisions for the entire plant are made by a single controller.



Figure 1.4 Illustration of centralized and decentralized architectures for thermal control of a multi-zone building

However, a key limitation of centralized decision making is potentially inferior robustness to sensor and communication network failures. A faulty reading by one sensor can affect the control decisions communicated to all actuators, thus distributing the effect of a local failure plant-wide [10, 11]. In the context of building thermal control, it implies that many of the conditioned zones will be affected until the fault is detected and diagnosed.

Due to these robustness concerns, decentralized decision making (see illustration in Figure 1.4) may be preferable for such large scale systems [12, 13]. It is more resilient to sensor and communication network faults because of its ability to contain them locally. Other benefits of decentralization include flexibility in operation, and simplified design and tuning. A decentralized control architecture consists of multiple disjoint control clusters, where each cluster determines only a subset of the plant-wide control inputs. The clusters do not communicate, i.e. decisions for the control inputs within a cluster are independent of any other cluster. Thus, failures originating in one cluster are prevented from affecting other clusters. It is therefore clear that with smaller clusters, the effect of such failures is more localized.

Although decentralization has merits from a robustness perspective, control decisions in a centralized architecture are better informed than in a decentralized architecture because the latter disregards any inter-cluster interactions in the decision making process. Therefore, it is expected that a decentralized control scheme yields suboptimal performance with respect to a centralized scheme and furthermore, the performance deterioration increases with the extent of decentralization. Hence, the `degree of decentralization' results in a trade-off between optimality and robustness.

Decentralized control has been applied to a wide variety of applications such as coordination of multi-robot systems [14, 15], control of satellite formations [16], and control of automated manufacturing systems [17, 18]. However, the decentralized control architecture is chosen in such a way that each controller caters to an individual physical unit of the overall system such as a robot, a satellite or a single machine. Decentralization in the context of building thermal control has also been studied previously [19-23]. However, similar to the applications previously mentioned, the most common architecture is a multi-agent scheme where each control agent is matched to a single zone in the building. This choice results in the smallest possible cluster size, which is beneficial from a robustness point of view. What is desired is a systematic

decentralization procedure that can quantify the specific trade-offs under consideration that exist in a control design context. In this thesis we seek to address this need in the specific context of building thermal control, by developing methodologies to determine appropriate decentralized control architectures, which provide a satisfactory trade-off between optimality and robustness objectives.

1.2.2 Decentralized control design

The design of controllers based on decentralized architectures obtained using the methodologies developed in this thesis is another important research objective. A key issue in control design for thermal control of buildings is the unavailability of accurate and reliable information about certain aspects relevant to the thermal dynamics. In particular, thermostats installed in buildings only measure the zone air temperatures which are associated with the thermal comfort of occupants. Therefore, temperatures of walls which also participate in the building's thermal dynamics are usually not known. Additionally, thermal contribution from factors such occupants, lighting, appliances and radiation are difficult to quantify and predict accurately, resulting in potentially large uncertainties in the description of thermal dynamics inside a building.

We observe that existing literature in the area of building thermal control seek to address these issues for a particular building by (a) using data-driven or parameter identification type of modeling approaches [24], (b) describing the dynamics only in terms of zone temperature states [8], (c) adding additional sensors for prediction of unknown states or thermal loads [25, 26], and (d) using high-fidelity models such as EnergyPlus [27] for prediction of states and loads which are otherwise not known [28]. In this thesis, we aim to explore control design methodologies which can be applied to a general class of buildings without the need to add additional sensors or develop potentially expensive high-fidelity models. We also seek to address other challenges associated with control design for thermal control of buildings such as potentially large dimension of the state-space for the underlying thermal dynamics [29] and the presence of constraints originating from practical considerations.

1.2.3 Evaluation of proposed tools and methodologies

It is important to verify the performance of the modeling and control tools developed in this work to achieve the above mentioned objectives. For this purpose, a realistic testing environment – either experimental or simulated – needs to be developed. Therefore, in the absence of experimental facilities, an important objective of this research is to develop a simulated test environment and employ it to validate the usefulness of the tools proposed for control architecture selection and decentralized control design.

1.3 Literature survey

1.3.1 Resources for building systems research

Some important resources providing information on the statistics, challenges and past and current efforts related to energy management of buildings are as follows.

1.3.1.1 ASHRAE

The American Society of Heating, Refrigeration and Air-Conditioning Engineers (ASHRAE) [30], founded in 1894 is an international organization of engineers, industrialists, scientists and researchers associated with the HVAC field. A few ASHRAE publications cater specially to building systems such as High Performing Buildings (a quarterly magazine presenting case studies on exemplary buildings designed for sustainability), Building Information Modeling Guide (available for free online) and the Load Calculation Applications Manual. In addition to these, the ASHRAE Journal, a monthly magazine, often features articles which focus on issues and technologies related to the design, operation and control of building HVAC systems. The society also publishes four handbooks related to the field (Fundamentals, HVAC Systems and Equipment, HVAC Applications and Refrigeration) which are periodically updated. These provide detailed technical descriptions of various HVAC components, together with general and component specific physical and modeling insights. ASHRAE also releases standards and guidelines to aid the design, selection and operation of HVAC systems.

1.3.1.2 Energy Information Administration

The Energy Information Administration (EIA) [31], created by the US Congress in 1977 is

an independent statistical agency within the US Department of Energy. The following articles are periodically published by the EIA and made available online, which contain both overall and sector-wise statistics regarding the national and international energy supply and demand:

- 1. Short Term Energy Outlook: Energy projections for the next 18 months, updated monthly.
- Annual Energy Outlook: Projection and analysis of US energy supply, demand, and prices through 2030 based on EIA's National Energy Modeling System.
- International Energy Outlook: Assessment of the outlook for international energy markets through 2030.
- 4. Monthly Energy Review: Statistics on monthly and annual US national energy consumption going back approximately 30 years, broken down by source.
- 5. Annual Energy Review: Primary report of historical annual energy statistics.

The statistics are presented sector-wise and at various levels of detail. For the building sector, both heating and cooling data is made available based on geographical region, building type and building features.

1.3.1.3 Europe's Energy Portal

Europe's Energy Portal [32] is an independently run commercial organization located within the European Union (EU). It features articles presenting statistics, issues, and technological and policy initiatives concerning emissions and energy in Europe. It also publishes EU directives related to energy and the environment. Detailed country-wise and sector-wise data, news and analysis are also provided.

1.3.1.4 Other resources

Some other general resources that provide background information and updates on activities related to the building energy area are as follows:

- USGBC [33]: The U.S. Green Building Council (USGBC), founded in 1993, is a nonprofit trade organization that promotes sustainability in how buildings are designed, built and operated. USGBC provides online resources related to energy efficiency in buildings systems including technical information, statistics, and case studies in the form of articles, webcasts, videos and presentations.
- 2. Facilitiesnet [34]: This is an online portal containing articles related to building technologies

and building management strategies. It also includes some case studies and links to other resources on energy efficient design and operation of buildings and data centers.

- 3. Building Technologies Program [35]: The Building Technologies Program (BTP) is funded by the US Department of Energy to promote research and technology development to reduce commercial and residential building energy usage. The program's website features resources such as guidelines for best practices and also links to other agencies and online information repositories.
- ENERGY STAR [36]: It is a joint program of the U.S. Environmental Protection Agency and the US Department of Energy. It provides online resources such as strategies and guidelines for the design of energy efficient buildings and plants.
- The Green Grid [37]: The Green Grid is a consortium of IT companies and professionals seeking to improve energy efficiency especially in data centers. Its website contains articles, survey findings, forum discussions and news updates.

1.3.2 Modeling and simulation of building thermal dynamics

A summary of papers and other resources on control oriented modeling and simulation of building thermal dynamics is presented here. Lumped parameter methods constitute the most common approach employed in literature to model the thermal interactions inside a building for control design. This is because other potentially more accurate characterizations, such as the use of partial differential equations to represent conductive and convective heat transfers, would require computationally intensive, finite-element solution methodologies, involving high dimensional state vectors. This limits their suitability for use in a control design procedure for a complex, interconnected system such as a building.

One of the first attempts at developing lumped parameter dynamic models of buildings was considered in [38]. A first order representation of the wall thermal dynamics based on construction properties was proposed, using the concept of "accessibility factors". Zones were also represented as first order systems and were connected to walls through resistances, hence resulting in a resistance capacitance (RC) network representation of the building thermal dynamics. More details on the underlying framework, also known as the 2R1C framework are provided in Section 3.3 of this thesis. This framework was experimentally validated by [39]. A

higher order lumped parameter framework, which represents each wall as a combination of three capacitors and two resistors (also known as the 3R2C framework) was proposed in [40]. In this paper, the resistances and capacitances were obtained by applying a model reduction procedure involving nonlinear constrained optimization on a higher order model. More details on the 3R2C framework are provided in Section 5.2 of this thesis. A related work is presented in [41], where a genetic algorithm based parameter identification methodology is proposed to obtain resistances and capacitances from experimental data to construct 3R2C representation of walls. A recent improvement on 3R2C modeling has been claimed in [42], which proposes a rule based methodology involving the concept of "dominant layer model" to compute resistances and capacitances. The method was applied to a real construction to demonstate improvements in accuracy over previously proposed 3R2C modeling approaches. The 3R2C framework has been further investigated in [29] to propose an aggregation based model reduction methodology which was shown to provide sufficient accuracy even after a large reduction in model order. An advantage of this method over other model reduction methods, as claimed by the authors is that the reduced order models retain the 3R2C framework.

Other types of lumped parameter modeling methods have also been inverstigated in literature, besides 2R1C and 3R2C. Grey-box modeling methodologies were investigated in [24, 43]. A semiparametric regression analysis was proposed in [24] to estimate unknown parameters and thermal loads in a grey-box model for building thermal dynamics. The methodology was used in conjunction with model predicive control to show reduction in energy consumption on an experimental test-bed. An Unscented Kalman Filter based approach was proposed in [43] to estimate the parameters of a grey-box model of building thermal dynamics. The approach was validated using EnergyPlus simulation data. Black-box system identification methods were proposed in [44, 28] to identify lumped parameter models from high-fidelity EnergyPlus models. Subspace identification methods were used in [44] and the identified models were experimentally validated. In [28], balanced model reduction was employed to reduced the order of the identified black-model for design of a model predictive controller. A model reduction method was proposed in [45] for non-linear models of building thermal dynamics. This method exploited the structure of the non-linear models and was shown to retain sufficient accuracy when compared to the full order model.

The US Department of Energy provides a list [46] of simulation tools that are available for free of cost and can be used to simulate the thermal dynamics in builings. Most of these tools use static or slowly-sampled modeling paradigms, which limits their use for control design or analysis. Therefore, they are primarily intended to provide the ability to test and improve the design of building construction and HVAC systems. Still, some of these tools can be used for control design and anlysis and are compared in Table 1.1.

Program	Level of state resolution	Level of time resolution	Software platform
EnergyPlus [27]	Zone and wall temperatures	Upto 1 minute	Fortran compiler with text based input and output interfaces
ESP-r [47]	Zone and wall temperatures	Upto 1 minute	C/Fortran
HAMLAB [48]	Zone and wall temperatures	< 1 minute allowed	MATLAB/SIMULINK
BuildingSim [49]	Zone and wall temperatures	Upto 1 minute	Java
SMILE [50]	Zone and wall temperatures	< 1 minute allowed	C and Python

 Table 1.1 An overview of control-oriented building simulation programs [46]

Among the programs listed above, EnergyPlus is a popularly used modeling environment which used detailed models to simulate the thermal dynamics of a building. The interested reader is directed to Chapter 6 and online tutorials provided in [27] for more information on EnergyPlus. Analysis of other energy simulation programs is beyond the scope of this thesis.

1.3.3 Thermal control of buildings

In this section, we provide a survey of literature on thermal control of buildings. Model Predictive Control (MPC) has been applied extensively to this area because of its ability to handle large scale, constrained optimal control problems. Furthermore, the computational complexity concerns which are typically associated with MPC are mitigated because of the slow evolution of thermal dynamics in buildings. For a background and technical details on MPC, the reader is directed to [51, 52].

MPC was used in [53, 54] to determine optimal zonal set-points and charging and discharging strategies for thermal energy storage, so as to optimize the energy usage at the building level. In particular, a detailed investigation of weather forecasting accuracy on the closed loop performance was undertaken. Field results from experimental investigation of the proposed MPC framework were reported. A stochastic MPC strategy for building climate control was proposed in [33] that takes into account weather predictions and comfort constraints. Nonlinear models with stochastic uncertainty were used for control design. The control strategy was experimentally investigated under various weather conditions and occupancy scenarios.

A quadratic MPC framework was employed in [55, 25] on a heating system for an experimental multi-zone building, and improvement in energy consumption over a baseline PID scheme was demonstrated. Learning based MPC was applied in [24] to demonstrate improvements in energy consumption over baseline control strategies on an experimental building. The proposed approach uses statistical techniques to learn the unmodeled dynamics and therefore improve model accuracy.

A min-max MPC framework with shrinking horizon lengths and pre-cooling was proposed in [28] to minimize the energy cost associated with building thermal management. The methodology was applied in simulation on an EnergyPlus model of a building to demonstrate improvements over baseline strategies. The potential of occupancy information to reduce energy consumption in buildings was investigated in [26]. Decentralized MPC strategies using current and predicted occupancy information were implemented at the zone level. Simulation results on a real world building model were presented to show improvements in energy efficiency. A supervisory MPC scheme was explored in [56] to minimize the electrical utility cost in buildings. A special emphasis was placed on the difficulties in optimization due to the demand charge component of the utility cost. The proposed framework was implemented on an EnergyPlus model of a building using MATLAB-EnergyPlus co-simulation approaches, and energy savings with respect to baseline control strategies were reported.

Distributed MPC for thermal control of multi-zone buildings was studied in [57] to

address the computational challenges associated with centralized MPC. A multi-agent control architecture involving coordination among agents was proposed, where each control agent uses Sequential Quadratic Programming, proximal minimization and dual decomposition to handle nonlinearities in the optimization framework. Simulation results were presented to demonstrate the improvements in energy efficiency over a baseline control strategy. Distributed MPC using coordination among agents was also investigated in [58]. The proposed methodology was based on linear ARX models (auto-regressive models with external inputs) and quadratic objective functions. Improvements in energy efficiency and computational complexity over a baseline controller and a centralized controller respectively were reported in simulation.

Robust MPC methodologies to address the issue of uncertainty in the thermal dynamics of buildings were investigated in [59]. Closed loop and open loop formulations of robust MPC were compared in simulation and it was concluded that the former outperforms the latter in terms of energy efficiency and robustness to disturbances. Lastly, very few studies in literature have focused on non MPC based strategies for thermal control of buildings. For example, a mean field decentralized control approach using a game-theoretic framework was proposed in [60] to address the complexity of centralized control. Optimal control strategies were derived based on the Hamilton-Jacobi-Bellman (HJB) principle and implemented in simulation to demonstrate energy savings over a baseline PID strategy.

1.4 Outline of the thesis

The remainder of this thesis is organized as follows. Chapter 2 provides a physical description of building systems with a special focus on variable air volume HVAC systems. Chapter 3 describes centralized and decentralized control architectures for the thermal control of buildings. These details are then used in Chapter 4 for the development of appropriate tools to enable control architecture decisions that balance the optimality and robustness requirements in the thermal control of buildings. Chapter 5 presents a control design framework for decentralized architectures obtained using the methodologies developed in Chapter 4. A simulated real world building example is studied in Chapter 6 to demonstrate the applicability of the control architecture selection and control design tools developed in this thesis. Lastly, the conclusions and research contributions from this work are presented in Chapter 7.

Chapter 2 Physical Details of Building Systems

2.1 Introduction

This chapter presents the physical details of building systems. The important control systems in most modern buildings are briefly described. This is followed by a detailed discussion on popularly used building thermal management systems known as variable air volume systems.

2.2 Control systems in a building

Most modern buildings consist of various control systems to meet the requirements of occupants such as thermal comfort, lighting, power and security, which are together referred to as a building management system. In this section, we describe the main features of systems used for thermal comfort, lighting and security systems. For more details the reader is directed to [61, 62].

2.2.1 Thermal systems

Thermal management in buildings is accomplished using heating, ventilation and airconditioning (HVAC) systems. HVAC systems used in modern buildings perform the following key functions:

- 1. Production of thermal energy by conversion from other forms such as mechanical, electrical, chemical etc.
- 2. Distribution of thermal energy to conditioned spaces in buildings.
- 3. Control and monitoring through sensors, controllers and actuators.

HVAC systems in buildings range from small scale window units to large scale district heating and cooling systems. For the building examples considered later in this work, we assume that thermal management is provide by use variable air volume (VAV) systems which are t employed in most medium and large scale buildings. More details on VAV systems are provided in Section 2.3. Discussion of other types of HVAC systems is beyond the scope of this thesis but the interested reader is directed to [63] for more details.

2.2.2 Lighting systems

Most modern buildings employ lighting control systems which provide several benefits over individual switching, such as reduced energy consumption, synchronization of lighting levels with activities, longer bulb life and reduced carbon emission footprints. A lighting control system is usually centralized and is implemented using an embedded processor or an industrial computer unit. It is typically based on rule based program logic which uses if-then-else constructs and/or logical operators to determine lighting levels (on/off states and intensity of lighting) at various locations in the building. The rules are based on one or a combination of the following factors:

- 1. Schedules based on the time of the day.
- Schedules based on the day of the week (weekday/weekend) or season of the year (winter/summer)
- 3. Occupancy based lighting schedules
- 4. Daylight based lighting schedules (daylight harvesting)
- 5. Rules for special events such as social occasions or holidays
- 6. Alarm triggers, e.g. "all lights on" in case of suspected intrusion.

An illustration of a rule based lighting control system is shown in Figure 2.1. The interested reader is directed to the online resource [65] for more information on lighting control systems such as equipment, architecture and protocols.

2.2.3 Security systems

Security systems are provided in residential and commercial buildings to prevent, alert or take remedial actions against undesired events such as intrusions, fire, excessive heat, flooding and carbon monoxide risks. Modern security systems employ multiple sensors at various locations in a building such as infrared/ultrasonic intrusion detectors, glass break detectors, video surveillance systems e.g. security cameras, and smoke, heat and/or carbon monoxide detectors. The trigger/alarm signals from each sensor is transmitted to one or more control units through wires or wireless means.

Depending upon the type of alarm, its location in the building, time of day, and other factors, the control units can automatically initiate various actions such as raising an alarm over the public announcement system, or calling an ambulance service, fire department or police department immediately. They may also be programmed to first call the property manager to verify if the alarm is genuine. The security control system can also trigger other systems such as the lighting control system to illuminate the entire building to facilitate evacuation, if necessary. A schematic of the various constituents of an automated security system is shown in Figure 2.2. The interested reader is directed to [67] for detailed information.



Figure 2.1 Illustration of a rule based lighting control system (Source [64])



Figure 2.2 Illustration of components in an automated building security system (Source: [66])

2.3 VAV systems

A variable air volume (VAV) system is a type of HVAC system typically used for airconditioning medium and large scale buildings. Physical details of VAV systems are described in this section.

2.3.1 Architecture

The architecture of a VAV system is illustrated in Figure 2.3. It consists of an airhandling unit (AHU), VAV terminal units, ducts and air terminals. A building can have one or more VAV systems depending on its size and layout. The AHU (Figure 2.4) recirculates the return air from the section of the building conditioned by it, which is then mixed with outside air. The mixing ratio is controlled using dampers. A fan is then used to transport the mixed air through a bank of cooling coils to cool the air and also reduce its humidity. If necessary, the air can also be heated and humidified through heating coils provided at the exit of the AHU. The conditioned air is then circulated to the terminal VAV units through ducts.



Figure 2.3 Architecture of a VAV system (source: [68]). Note that dampers are not shown in the air handling unit.



Figure 2.4 Illustration of an air-handling unit (source: [69])

The VAV terminal unit (Figure 2.5), also known as a VAV box is a zone level flow control device. A VAV box has two actuators – an air damper and a reheat coil. It is connected to a local or a central control system which typically seeks to achieve a specified set-point temperature in the zone by using the damper to regulate the mass flow rate of air supplied to the zone via air terminals. In the event that the zone temperature is lower than set-point and the mass flow rate of air cannot be reduced further, reheat coils are used to heat the air supplied to the zone.



Figure 2.5 Illustration of a VAV terminal unit (damper and reheat coils are not shown, source: Wikipedia)

The source of cooling in the cooling coils in an AHU is usually chilled water provided by a chiller unit installed in the building or a district cooling system to which the building is connected. Similarly, the source of heating in the heating coils in an AHU and the reheat coils in a VAV box is usually hot water or steam generated by a local heating unit in the building (e.g. a boiler, air furnace or geothermal pump) or provided by a district heating system. Such water or steam based heating and cooling systems are also known as hydronic systems. The reader is directed to Chapter 2 of [70] for a detailed discussion of such systems.

In addition to the heating and cooling roles, a VAV system also provides ventilation to the conditioned spaces due to continuous circulation of air. Therefore, a separate ventilation unit is not required. The interested reader can find more details on VAV systems in the handbook [71].

2.3.2 Sensing and actuation

Thermostats (Figure 2.6) installed in the zones in a building measure the zone temperatures which are used by the controllers to manipulate the dampers and/or the reheat coil power as mentioned earlier. Therefore, from a thermal control perspective, the sensors correspond to the thermostats and the actuators correspond to the dampers and reheat coils in the

VAV boxes. It should be noted that the temperature of air provided by the AHU is fixed (usually around 13 ⁰C).

The differential pressure down the duct changes as a result of changes in the damper positions in the VAV boxes, and is measured using a pressure sensor. Therefore, a fan controller is employed to change the fan speed via a variable frequency drive to regulate the differential static pressure down the duct around a specified set-point.



Figure 2.6 Illustration of a thermostat (source: [72])

2.3.3 Thermal control

The zones in a building are subjected to thermal loads originating from sources such as occupants, appliances, lighting, solar radiation and ambient. The primary purpose of thermal control is to offset these loads by manipulating the mass flow rate and temperature of supply air from the air terminals of the VAV system so as to maintain the zones at specified set-point temperatures. As described in Section 2.3.2, the controllers use thermostats as sensors and the dampers and reheat coils in the VAV boxes as actuators. Historically, pneumatic control was employed, but direct digital control systems (Figure 2.7) have become popular in recent times. The control architecture can be centralized (one single control agent for all VAV boxes) or decentralized (a different control agent for each VAV box) and is usually Proportional Integral Derivative (PID) [74].



Figure 2.7 Illustration of a digital VAV controller (Source: [73])

Chapter 3 Centralized and Decentralized Control Frameworks

Centralized and decentralized Model Predictive Control (MPC) frameworks for the problem of building thermal control are presented in this chapter. These frameworks form the basis for the development of control architecture selection methodologies presented in Chapter 4. The list of common symbols used in this chapter is shown in Table 3.1.

Symbol	Description
Sz	Set of all zones in the building
S _{zi}	<i>ith</i> cluster in a <i>p</i> -partition
N _w	Number of walls in the building
N _{wi}	Number of walls in cluster S_{zi}
Nz	Number of zones in the building
N _{zi}	Number of zones in cluster S_{zi}
$\mathbf{T}_{\mathbf{w}}(k)$	Vector of wall temperatures at time <i>k</i>
$\mathbf{T}_{\mathbf{wi}}(k)$	Vector of wall temperatures within i^{th} cluster at time k
$\mathbf{T_z}(k)$	Vector of zone temperatures at time k
$\mathbf{T}_{\mathbf{zi}}(k)$	Vector of zone temperatures within i^{th} cluster at time k

 Table 3.1: Nomenclature of common symbols in Chapter 3
u (<i>k</i>)	Vector of control inputs at time k		
$\mathbf{u_i}(k)$	Vector of control inputs for i^{th} cluster at time k		
$T_a(k)$	Ambient temperature at time k		
$T_g(k)$	Ground temperature at time k		
$\mathbf{d}_{\mathbf{w}}(k)$	Vector of unmodeled thermal loads acting on walls at time k		
d _z (<i>k</i>)	Vector of unmodeled thermal loads acting on zones at time k		
$\mathbf{d}_{\mathbf{wi}}(k)$	Vector of unmodeled thermal loads in i^{th} cluster's walls at time k		
$\mathbf{d}_{\mathbf{z}\mathbf{i}}(k)$	Vector of unmodeled thermal loads in i^{th} cluster's zones at time k		
$\mathbf{T}_{\mathbf{z},\mathbf{ref}}(k)$	Vector of zone temperature set-points at time <i>k</i>		
$\mathbf{T}_{\mathbf{zi},\mathbf{ref}}(k)$	Vector of i^{th} cluster zone temperature set-points at time k		
α	Vector of weights on cost objective		
α	Vector of weights on i^{th} cluster's cost objective		
β	Vector of weights on performance objective		
β _i	Vector of weights on i^{th} cluster's performance objective		
N	Number of samples in the control and prediction horizon		
T _s	Sample time for discretization of thermal dynamics		
x(k+l k)	Predicted value of quantity x, after l time steps in future, given $x(k)$		
I _N	Identity matrix of dimension $N \times N$		
0 _{<i>N</i>,<i>M</i>}	Zero matrix of dimension $N \times M$		

3.1 Thermal control of buildings

3.1.1 Control objectives

In this section, we revisit the objectives in the thermal control of buildings described in Secion 1.1. The primary objective in the thermal control of buildings is to provide desired levels of occupant comfort in their air conditioned sections. An important part of the occupant comfort requirements is to achieve desired temperature set-points that are prescribed manually by the users or auto-programmed by the Building Automation System (BAS). Depending on the specific requirments of occupants, activity levels, etc., the set-points can vary from one zone to another at any given time, as well as for the same zone at different times in the day. Another important control objective is to reduce the operating cost or power required by the heating, ventilation and air-conditioning (HVAC) systems while seeking to provide the occupant comfort requirements. This is motivated by the need to operate buildings efficienctly, as described in Chapter 1.

From a controls perspective, a building is a complex multi-input, multi-output (MIMO) system employing multiple sensors and actuators to meet the air-conditioning requirements as discussed in Chapter 2. Such a system can be susceptible to failures originating in the sensors, actuators or the commucation infrastructure which integrates them with various elements of the control network. It is desired that any such failure should have a limited effect on the satisfaction of building-wide occupant comfort requirements before the fault is detected and diagnosed. Therefore, resilience to such failures is another important objective that should be considered in the thermal control of buildings. Also, from an implementation perspective, it is desired that the control framework be scalable across buildings, irrespetive of their size or layout.

3.1.2 Control aspects

As mentioned in Chapter 1, two important aspects that need to be considered while designing controllers to achieve the afore-mentioned objectives are (i) control architecture, and (ii) control methodology. These aspects are explained below.

3.1.2.1 Control architecture

The existence of a tradeoff between optimality and robustness with respect to the degree of decentralization of the control architecture was discussed in Section 1.2.1. To further explain the demerits of centralized control, we consider a simulation case study performed on a 6 room water-cooled building shown in Figure 3.1. The underlying control objective is to achieve prescribed set-point temperatures (25° C) in all the rooms. A centralized control scheme based on Model Predictive Control (MPC) and a decentralized control scheme (at the room level) based on single input single output (SISO), proprtional-integral (PI) control were implemented. Details of the plant and the controllers with the relevant codes and models are provided in the media accompanying this thesis. The desired objective of temperature regulation was met by both these controllers under normal circumstances. However, the performance when a fault was introduced in the sensor in the atrium (room 1) is shown in Figure 3.2. It was observed that with centralized control, sensor failure in room 1 significantly affected the performance in the other rooms of the building (Figure 3.2 (a)). With decentralized control, however, the effect was limited to room 1, where the fault originated (Figure 3.2 (b)).



Figure 3.1: Six zone building used in the case study (return and supply water lines for chilled water loop also shown).



(a) Regulation errors under centralized control



(b) Regulation errors under decentralized control

Figure 3.2: Effect of sensor failure in room 1 of building shown in Fig. 3.1

Therefore, the control architecture plays an important role in achieving the control objectives listed in section 3.1.1. Besides centralized and decentralized, other configurations such as hierarchical or overlapping architectures [75] can also be used for control. However, the scope of this thesis is limited to the study of centralized and decentralized control architectures only, because of the natural tradeoff between optimality and robustness associated with them as explained above and in Section 1.2.1.

3.1.2.2 Control methodology

Various control methodologies have been implemented in literature for the thermal control of buildings, ranging from model-free approaches such as on-off or PID [76,77] to model-based optimal control methods such as Linear Quadratic Regulation (LQR) [78] and model-predictive control (MPC) [8,79]. Among these approaches, MPC has been of considerable interest recently [8,79,80,81], (also see references in Section 1.3.3) because of its proven effectiveness in handling large scale, constrained, optimal control problems. Therefore, the control methodology used in this work is based on an MPC framework.

The centralized and decentralized MPC frameworks presented in this chapter are developed to serve as a basis for the development of appropriate procedures for control architecture selection in Chapter 4. However, appropriate modifications to these frameworks are necessary for their implementation. These modifications will be discussed in chapter 5.

3.2 Preliminaries

A few preliminaries are required before a formal description of the centralized and decentralized control frameworks can be presented.

Definition 3.1 (Zones): A zone in a building is defined as a cluster of rooms for which the thermal demands are met using a common actuator. The set of all zones is denoted by S_z , which has N_z elements.

Definition 3.2. (*p-partition*): A *p-partition* (or simply a partition) of the building is defined as any set of *p* non-empty and non-overlapping subsets of S_z that cover all of S_z , where $p \in$ $\{1,2,...,N_z\}$. The elements which constitute a *p-partition* are called its *clusters*, denoted by S_{zi} , where i = 1,2,...,p. The number of elements in S_{zi} is denoted by N_{zi} . The above properties can be formally stated as:

 $1. S_{zi} \neq \emptyset \quad for \ all \ i \in \{1, 2, \dots, p\},$

2.
$$\bigcup_{i=1}^{p} S_{zi} = S_z$$
, and

3. $S_{zi} \cap S_{zj} = \emptyset$ for all $i, j \in \{1, 2, \dots, p\}$ and $i \neq j$.

As an illustration of these definitions, consider a simple 3-zone building shown in Figure 3.3. The set of zones, S_z for this example is $\{1, 2, 3\}$. It has exactly three 2-partitions which are $\{\{1,2\},\{3\}\},\{\{1\},\{2,3\}\}\}$ and $\{\{1,3\},\{2\}\}$. Furthermore, the only *1-partition* and 3-partition of

 S_z are {{1,2,3}} and {{1},{2},{3}} respectively.



Figure 3.3: An example 3-zone building (bottom surface of each zone faces ground – all other external surfaces are exposed to ambient)

3.3 Centralized control framework

3.3.1 Architecture

The proposed centralized MPC architecture for the thermal control of buildings, illustrated in Figure 3.4 consists of a single control agent which determines the plant-wide control inputs, based on the feedback of building-wide sensory data (zone and wall temperatures), and appropriate forecasts of disturbances such as ambient temperature, ground temperatures and thermal loads acting on the building's walls and zones. The control decisions are arrived at using a discrete time MPC approach, where an objective function is minimized using a centralized system model that serves the purpose of constraints in the optimization. The control inputs represent the rates of energy transfer – positive for heating and negative for cooling – provided to the zones by the HVAC system.



Figure 3.4: Schematic of centralized MPC architecture

3.3.2 Objective function

As explained in Section 3.1.1, a primary objective in the thermal control of buildings is to achieve desired temperature set-points for the various zones. A secondary control objective is to reduce the HVAC operating cost or energy consumption associated with meeting the specified temperature set-points. A weighted sum of these objectives is used to construct an overall objective function, as shown in (3.1), for optimization over a finite time-horizon in a discrete time setting. The size of the time-horizon, measured in terms of the number of samples, is denoted by N. The first term, $J^{c,perf}$ represents the temperature set-point regulation objective across all zones whereas the second term, $J^{c,cost}$ represents either the HVAC operating cost or power consumption. It should be noted that the power consumed by the HVAC system in conditioning a zone depends upon the absolute value of the energy transfer irrespective of its sign (heating or cooling). Therefore, a quadratic function of these energy transfers is used in $J^{c,cost}$. Further, the choice of quadratic functions to represent the constituent terms in the objective function imparts strict convexity, which is a desired property in static optimization problems [82]. To render the framework less restrictive, the weights α and β , in $J^{c,perf}$ and $I^{c,cost}$ respectively, are specified as vector quantities which allow the flexibility of assigning different weights for different zones. The notation used is defined in the nomenclature (Table 3.1) and is consistent with standard practice in MPC literature [12].

$$J^c = J^{c,perf} + J^{c,cost} \tag{3.1}$$

With,

$$J^{c,perf} = \sum_{j=1}^{N} \left(\mathbf{T}_{\mathbf{z}}(k+j|k) - \mathbf{T}_{\mathbf{z},ref}(k) \right)^{T} diag(\boldsymbol{\beta}) \left(\mathbf{T}_{\mathbf{z}}(k+j|k) - \mathbf{T}_{\mathbf{z},ref}(k) \right)$$
(3.2)

$$J^{c,cost} = \sum_{j=0}^{N-1} \mathbf{u}(k+j|k)^T diag(\mathbf{\alpha}) \ \mathbf{u}(k+j|k)$$
(3.3)

3.3.3 Model

An appropriate model is required to characterize the effect of the control variables on the feedback variables of interest at each time step in the optimization. The zone temperatures are dynamically interconnected by heat flow occurring through internal walls. The ambient temperature, T_a and ground temperature, T_g also affect the thermal behavior in the zones through

the external walls (building envelope), and can be treated as disturbances in the context of the overall system dynamics. A s reviewed in Section 1.3.2, a simple approach for modeling these thermal interactions, which is widely used in literature is to treat the building as a lumped resistive-capacitive (RC) network. Other potentially more accurate characterizations, such as the use of partial differential equations to represent conductive and convective heat transfers, would require computationally intensive, finite-element solution methodologies, involving high dimensional state vectors. This limits their suitability for use in a control design or analysis procedure for a complex, interconnected system such as a building.

The model used in this work is based on [38], revisited in [40], where the walls and zones are represented by a capacitor each, with capacitance equal to the corresponding thermal mass (Figure 3.5). The system states are the (lumped) temperatures of the walls and zones. The control inputs correspond to the energy transfer rates in kW (heating or cooling) that the HVAC system provides to the zones. In a variable air volume (VAV) air-conditioning system, these can be modulated by adjusting dampers in the VAV boxes to set air flow rates or by manipulating the supply air temperature provided by the Air Handling Units (AHU) [79]. The heat transfer between a wall and any of its adjacent zones or the ambient/ground (in case of external walls) is characterized by a resistor, with resistance set to the inverse of the corresponding heat transfer coefficient. Various other factors also affect the thermal dynamics, such as heat flows contributed by occupants, lights, appliances, direct or indirect solar radiation, and thermal infiltration. In this work, these factors are not modeled separately and only their lumped contribution to each zone and wall is represented using thermal disturbance vectors \mathbf{d}_z and \mathbf{d}_w having units of kW. This is because as seen in Chapter 4, these disturbances do not affect the control architecture selection methodologies.

The resulting linear, discrete time, state space model for the building thermal dynamics using the afore-mentioned assumptions is as shown.

$$\begin{bmatrix} \mathbf{T}_{\mathbf{w}} \\ \mathbf{T}_{\mathbf{z}} \end{bmatrix} (k+1) = \underbrace{\begin{bmatrix} \mathbf{A}_{\mathbf{w},\mathbf{w}} & \mathbf{A}_{\mathbf{w},\mathbf{z}} \\ \mathbf{A}_{\mathbf{z},\mathbf{w}} & \mathbf{A}_{\mathbf{z},\mathbf{z}} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{T}_{\mathbf{w}} \\ \mathbf{T}_{\mathbf{z}} \end{bmatrix} (k) + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{\mathbf{z}} \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} \mathbf{B}_{\mathbf{w},\mathbf{a}} & \mathbf{B}_{\mathbf{w}g} & \mathbf{B}_{\mathbf{w},\mathbf{d}w} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{\mathbf{z},\mathbf{d}z} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{a} \\ \mathbf{T}_{g} \\ \mathbf{d}_{w} \\ \mathbf{d}_{z} \end{bmatrix} (k) (3.4)$$



Figure 3.5 Building block for RC network model of thermal interactions. Figure represents an internal wall flanked by zones on either side of it. The unmodeled thermal loads are also shown via current sources.

Here, the state transition matrix, **A** is partitioned into sub-matrices $A_{w,w}$, $A_{w,z}$, $A_{z,w}$ and $A_{z,z}$. These sub-matrices, together with the other matrices B_z , $B_{w,a}$, $B_{w,g}$, $B_{w,dw}$ and $B_{z,dz}$ appearing in the model can be obtained for any general building from a knowledge of the underlying resistance and capacitance values in the RC network via the procedure described in Algorithm 3.1. In this algorithm, a weighted graph is used to represent the resistances in the network. A Laplacian matrix is then constructed, which represents the net flow of energy into each node in the graph, thus allowing the application of the First Law of Thermodynamics (conservation of energy) at the nodes.

Algorithm 3.1. Generation of state space model for building thermal dynamics from a RC network

STEP 1: A weighted graph is created with nodes for each of the walls, the zones, the ambient and ground (see Figure 3.6). While numbering the nodes, those representing walls are numbered first, followed by the zones, the ambient and lastly the ground. Each wall node is connected by undirected edges to the two nodes to which it is thermally connected. This results in nodes

representing external walls to be connected to a zone node and the ambient/ground temperature node. Similarly, the internals walls are connected to a pair of zone nodes. The weight of each edge is set to be the inverse of the corresponding thermal resistance between the two nodes it connects. The resulting weighted graph is denoted by G = (V, E) along with the weight function $w: E \mapsto \mathbb{R}^+$. Here, V and E are the sets of vertices and edges, respectively, in graph G. We also define capacitance matrices, C_w and C_z which are diagonal matrices of the thermal capacitances associated with the walls and the zones respectively. The diagonal entries in these matrices are entered in the order of the corresponding node numbers in G.

STEP 2: The Laplacian matrix of G, denoted by L_G is then obtained as:

$$\mathbf{L}_{\mathbf{G}} = \mathbf{D}_{\mathbf{G}} - \mathbf{A}_{\mathbf{G}} \tag{3.5}$$

Where,

$$\mathbf{A}_{\mathbf{G}}(i,j) = \begin{cases} w(i,j) & \text{if}(i,j) \in E\\ 0 & \text{otherwise} \end{cases}$$
$$\mathbf{D}_{\mathbf{G}}(i,i) = \sum_{j} A_{G}(i,j)$$

We extract a square sub-matrix from L_G which corresponds to its first $N_w + N_z$ rows and columns, and denote the result by L_{Gx} . Next, a column vector denoted by L_{Ga} is extracted which corresponds to the first N_w rows and the $(N_w + N_z + 1)^{th}$ column of L_G . Similarly, another column vector denoted by L_{Gg} is extracted which corresponds to the first N_w rows and the last column of L_G .

STEP 3: The following matrices are now defined:

$$\mathbf{A}_{\text{cont}} = \begin{bmatrix} \mathbf{C}_{\mathbf{w}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{z}}^{-1} \end{bmatrix} \mathbf{L}_{\mathbf{G}\mathbf{x}}$$
(3.6)

$$\mathbf{B}_{\mathbf{a},\mathbf{cont}} = \mathbf{C}_{\mathbf{w}}^{-1} \mathbf{L}_{\mathbf{Ga}} \tag{3.7}$$

$$\mathbf{B}_{\mathbf{g},\mathbf{cont}} = \mathbf{C}_{\mathbf{w}}^{-1} \mathbf{L}_{\mathbf{Gg}} \tag{3.8}$$

$$\mathbf{B}_{\mathrm{dz,cont}} = \mathbf{B}_{\mathrm{z,cont}} = \mathbf{C}_{\mathrm{z}}^{-1} \tag{3.9}$$

$$\mathbf{B}_{\mathrm{dw,cont}} = \mathbf{C}_{\mathbf{w}}^{-1} \tag{3.10}$$



Figure 3.6: Graph representation for the 3-zone building in Fig. 3.3. Nodes 1-10 represent walls, 11-13 are zones, 14 corresponds to ambient and 15 corresponds to ground.

STEP 4: The continuous time model for the building thermal dynamics is obtained as shown in (3.11).

$$\frac{d}{dt} \begin{bmatrix} \mathbf{T}_{\mathbf{w}} \\ \mathbf{T}_{\mathbf{z}} \end{bmatrix} = \mathbf{A}_{\text{cont}} \begin{bmatrix} \mathbf{T}_{\mathbf{w}} \\ \mathbf{T}_{\mathbf{z}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{\mathbf{z},\text{cont}} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{B}_{\mathbf{a},\text{cont}} & \mathbf{B}_{\mathbf{g},\text{cont}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{\mathbf{dz},\text{cont}} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{a} \\ \mathbf{T}_{g} \\ \mathbf{d}_{w} \\ \mathbf{d}_{z} \end{bmatrix}$$
(3.11)

The model (3.4) is obtained by the discretization of (3.11) using an appropriately chosen sample time T_s . In the examples used in this thesis, the discretization is performed in MATLAB using the 'zero order hold' method. It should be noted that A_{ww} , A_{zz} , B_z , $B_{z,dz}$ and $B_{w,dw}$ are diagonal matrices as a result of the construction procedure described above.

In most modern buildings, the zone temperature measurements are available using thermostats. However, wall temperatures measurements may not be available. We consider the estimation of wall temperatures in Chapter 5 when designing the centralized and decentralized controllers. For the purposes of control architecture selection however, we assume that full state measurement is available. This implies that at each time instant k, the wall temperatures $\mathbf{T}_{\mathbf{w}}(k)$ and zone temperatures $\mathbf{T}_{\mathbf{z}}(k)$ are known.

3.3.4 Conversion to Quadratic Program

Over the selected prediction horizon, the model given by (3.4) is used to predict the future states $\{\mathbf{T}_{\mathbf{w}}(k+l|k)\}_{l=1}^{l=N}$ and $\{\mathbf{T}_{\mathbf{z}}(k+l|k)\}_{l=1}^{l=N}$ in terms of any chosen current and future control inputs $\{\mathbf{u}(k+l|k)\}_{l=0}^{l=N-1}$, current state measurements $\mathbf{T}_{\mathbf{w}}(k)$, $\mathbf{T}_{\mathbf{z}}(k)$, forecasted values of unmodeled thermal loads $\{\mathbf{d}_{\mathbf{w}}(k+l)\}_{l=0}^{l=N-1}$ and $\{\mathbf{d}_{\mathbf{z}}(k+l)\}_{l=0}^{l=N-1}$, ambient temperatures $\{T_{a}(k+l)\}_{l=0}^{l=N-1}$ and ground temperatures $\{T_{g}(k+l)\}_{l=0}^{l=N-1}$.

Let
$$\mathbf{x} = \begin{bmatrix} \mathbf{T}_{\mathbf{w}} \\ \mathbf{T}_{\mathbf{z}} \end{bmatrix}$$
, $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{\mathbf{w},\mathbf{w}} & \mathbf{A}_{\mathbf{w},\mathbf{z}} \\ \mathbf{A}_{\mathbf{z},\mathbf{w}} & \mathbf{A}_{\mathbf{z},\mathbf{z}} \end{bmatrix}$, $\mathbf{B}_{\mathbf{u}} = \begin{bmatrix} \mathbf{0}_{N_{\mathbf{w}} \times N_{\mathbf{z}}} \\ \mathbf{B}_{\mathbf{z}} \end{bmatrix}$, $\mathbf{B}_{\mathbf{g}} = \begin{bmatrix} \mathbf{B}_{\mathbf{w},\mathbf{g}} \\ \mathbf{0}_{N_{z} \times 1} \end{bmatrix}$, $\mathbf{B}_{\mathbf{g}} = \begin{bmatrix} \mathbf{B}_{\mathbf{w},\mathbf{g}} \\ \mathbf{0}_{N_{z} \times 1} \end{bmatrix}$, $\mathbf{B}_{\mathbf{d}\mathbf{w}} = \begin{bmatrix} \mathbf{B}_{\mathbf{w},\mathbf{d}\mathbf{w}} \\ \mathbf{0}_{N_{z} \times N_{w}} \end{bmatrix}$ and $\mathbf{B}_{\mathbf{d}\mathbf{z}} = \begin{bmatrix} \mathbf{0}_{N_{w} \times N_{z}} \\ \mathbf{B}_{\mathbf{z},\mathbf{d}\mathbf{z}} \end{bmatrix}$. Using (3.4) we obtain
 $\mathbf{x}(k+1|k) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_{\mathbf{u}}\mathbf{u}(k) + \mathbf{B}_{\mathbf{a}}\mathbf{T}_{\mathbf{a}}(k) + \mathbf{B}_{\mathbf{g}}\mathbf{T}_{\mathbf{g}}(k) + \mathbf{B}_{\mathbf{d}\mathbf{z}}\mathbf{d}_{\mathbf{z}}(k)$
 $\mathbf{x}(k+2|k) = \mathbf{A}\mathbf{x}(k+1|k) + \mathbf{B}_{\mathbf{u}}\mathbf{u}(k+1|k) + \mathbf{B}_{\mathbf{a}}\mathbf{T}_{\mathbf{a}}(k+1|k) + \mathbf{B}_{\mathbf{g}}\mathbf{T}_{\mathbf{g}}(k+1|k)$
 $+ \mathbf{B}_{\mathbf{d}\mathbf{w}}\mathbf{d}_{\mathbf{w}}(k+1|k) + \mathbf{B}_{\mathbf{d}\mathbf{z}}\mathbf{d}_{\mathbf{z}}(k+1|k)$
 $= \mathbf{A}^{2}x(k) + \mathbf{A}\mathbf{B}_{\mathbf{u}}\mathbf{u}(k) + \mathbf{B}_{\mathbf{u}}\mathbf{u}(k+1|k) + \mathbf{A}\mathbf{B}_{\mathbf{a}}\mathbf{T}_{\mathbf{a}}(k) + \mathbf{B}_{\mathbf{a}}\mathbf{T}_{\mathbf{a}}(k+1|k)$
 $+ \mathbf{A}\mathbf{B}_{\mathbf{g}}\mathbf{T}_{\mathbf{g}}(k) + \mathbf{B}_{\mathbf{g}}\mathbf{T}_{\mathbf{g}}(k+1|k) + \mathbf{A}\mathbf{B}_{\mathbf{d}\mathbf{w}}\mathbf{d}_{\mathbf{w}}(k) + \mathbf{B}_{\mathbf{d}\mathbf{w}}\mathbf{d}_{\mathbf{w}}(k+1|k) + \mathbf{A}\mathbf{B}_{\mathbf{d}\mathbf{z}}d_{z}(k)$

÷

$$\mathbf{x}(k+N|k) = \mathbf{A}^{N}\mathbf{x}(k) + \sum_{i=0}^{N-1} \mathbf{A}^{N-1-i} [\mathbf{B}_{\mathbf{u}}\mathbf{u}(k+i|k) + \mathbf{B}_{\mathbf{a}}\mathbf{T}_{\mathbf{a}}(k+i|k) + \mathbf{B}_{\mathbf{g}}\mathbf{T}_{\mathbf{g}}(k+i|k) + \mathbf{B}_{\mathbf{d}}\mathbf{w}(k+i|k) + \mathbf{B}_{\mathbf{d}}\mathbf{z}(k+i|k)]$$

The above equations can be succintly written as

$$\bar{\mathbf{x}} = \mathbf{M}\mathbf{x}(k) + \mathbf{S}_{\mathbf{u}}\bar{\mathbf{u}} + \mathbf{S}_{\mathbf{a}}\bar{\mathbf{T}}_{\mathbf{a}} + \mathbf{S}_{\mathbf{g}}\bar{\mathbf{T}}_{\mathbf{g}} + \mathbf{S}_{\mathbf{dw}}\bar{\mathbf{d}}_{\mathbf{w}} + \mathbf{S}_{\mathbf{dz}}\bar{\mathbf{d}}_{\mathbf{z}}.$$
 (3.12)

Here,

$$\begin{split} \bar{\mathbf{x}} &= \begin{bmatrix} \mathbf{x}(k+1|k) \\ \mathbf{x}(k+2|k) \\ \vdots \\ \mathbf{x}(k+N|k) \end{bmatrix}, \ \bar{\mathbf{T}}_{a} &= \begin{bmatrix} \mathbf{T}_{a}(k) \\ \mathbf{T}_{a}(k+1|k) \\ \vdots \\ \mathbf{T}_{a}(k+N-1|k) \end{bmatrix}, \ \bar{\mathbf{T}}_{g} &= \begin{bmatrix} \mathbf{T}_{g}(k) \\ \mathbf{T}_{g}(k+1|k) \\ \vdots \\ \mathbf{T}_{g}(k+N-1|k) \end{bmatrix}, \\ \bar{\mathbf{u}} &= \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1|k) \\ \vdots \\ \mathbf{u}(k+N-1|k) \end{bmatrix}, \ \bar{\mathbf{d}}_{w} &= \begin{bmatrix} \mathbf{d}_{w}(k) \\ \mathbf{d}_{w}(k+1|k) \\ \vdots \\ \mathbf{d}_{w}(k+N-1|k) \end{bmatrix}, \ \bar{\mathbf{d}}_{z} &= \begin{bmatrix} \mathbf{d}_{z}(k) \\ \mathbf{d}_{z}(k+1|k) \\ \vdots \\ \mathbf{d}_{z}(k+N-1|k) \end{bmatrix}, \ \mathbf{M} &= \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^{2} \\ \vdots \\ \mathbf{A}^{N} \end{bmatrix}, \\ \mathbf{S}_{u} &= \begin{bmatrix} \mathbf{B}_{u} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{AB}_{u} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{u} & \mathbf{0} \\ \mathbf{A}^{N-1}\mathbf{B}_{u} & \dots & \mathbf{AB}_{u} & \mathbf{B}_{u} \end{bmatrix}, \ \mathbf{S}_{a} &= \begin{bmatrix} \mathbf{B}_{a} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{AB}_{a} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{a} & \mathbf{0} \\ \mathbf{A}^{N-1}\mathbf{B}_{a} & \dots & \mathbf{AB}_{a} & \mathbf{B}_{a} \end{bmatrix}, \\ \mathbf{S}_{g} &= \begin{bmatrix} \mathbf{B}_{g} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{AB}_{g} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{g} & \mathbf{0} \\ \mathbf{A}^{N-1}\mathbf{B}_{g} & \dots & \mathbf{AB}_{g} & \mathbf{B}_{g} \end{bmatrix}, \ \mathbf{S}_{dw} &= \begin{bmatrix} \mathbf{B}_{dw} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{AB}_{dw} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{dw} & \mathbf{0} \\ \mathbf{A}^{N-1}\mathbf{B}_{dw} & \dots & \mathbf{AB}_{dw} & \mathbf{B}_{dw} \end{bmatrix}, \text{ and } \\ \mathbf{S}_{dz} &= \begin{bmatrix} \mathbf{B}_{dz} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{AB}_{dz} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{dz} & \mathbf{0} \\ \mathbf{A}^{N-1}\mathbf{B}_{dz} & \dots & \mathbf{AB}_{dz} & \mathbf{0}_{dz} \end{bmatrix}. \end{split}$$

From (3.1) J^c can be re-stated as

$$J^{c} = \left(\overline{\mathbf{T}}_{z} - \overline{\mathbf{T}}_{z,ref}\right)^{T} \mathbf{Q}_{1} \left(\overline{\mathbf{T}}_{z} - \overline{\mathbf{T}}_{z,ref}\right) + \overline{\mathbf{u}}^{T} \mathbf{Q}_{2} \overline{\mathbf{u}}.$$
 (3.13)

Where,

$$\begin{split} \mathbf{\bar{T}}_{z} &= \begin{bmatrix} \mathbf{T}_{z}(k+1|k) \\ \mathbf{T}_{z}(k+2|k) \\ \vdots \\ \mathbf{T}_{z}(k+N|k) \end{bmatrix}, \mathbf{\bar{T}}_{z,ref} = \begin{bmatrix} \mathbf{T}_{z,ref}(k) \\ \mathbf{T}_{z,ref}(k) \\ \vdots \\ \mathbf{T}_{z,ref}(k) \end{bmatrix}_{N \times N_{z}}^{,}, \\ \mathbf{Q}_{1} &= \begin{bmatrix} diag(\boldsymbol{\beta}) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & diag(\boldsymbol{\beta}) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & diag(\boldsymbol{\beta}) \end{bmatrix}_{(N_{z},N) \times (N_{z},N)}^{,}, \\ \text{and } \mathbf{Q}_{2} &= \begin{bmatrix} diag(\boldsymbol{\alpha}) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & diag(\boldsymbol{\alpha}) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & diag(\boldsymbol{\alpha}) \end{bmatrix}_{(N_{z},N) \times (N_{z},N)}^{,}. \\ \text{Let } \mathbf{C} &= \begin{bmatrix} \mathbf{0}_{N_{z} \times N_{w}} & \mathbf{I}_{N_{z}} \end{bmatrix} \text{ and } \mathbf{\bar{C}} &= \begin{bmatrix} \mathbf{C} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & diag(\boldsymbol{\alpha}) \end{bmatrix}_{(N_{z},N) \times (N_{z},N)}^{,}. \\ \text{Clearly, } & \mathbf{\bar{T}}_{z} &= \mathbf{\bar{C}}\mathbf{\bar{x}} . \end{split}$$

$$(3.14)$$

Using (3.12), (3.14) can be written as

$$\overline{\mathbf{T}}_{\mathbf{z}} = \overline{\mathbf{C}} \Big(\mathbf{M} \mathbf{x}(k) + \mathbf{S}_{\mathbf{u}} \overline{\mathbf{u}} + \mathbf{S}_{\mathbf{a}} \overline{\mathbf{T}}_{\mathbf{a}} + \mathbf{S}_{\mathbf{g}} \overline{\mathbf{T}}_{\mathbf{g}} + \mathbf{S}_{\mathbf{d}w} \overline{\mathbf{d}}_{\mathbf{w}} + \mathbf{S}_{\mathbf{d}z} \overline{\mathbf{d}}_{\mathbf{z}} \Big) \,. \tag{3.15}$$

Substituting $\overline{\mathbf{T}}_{z}$ from (3.15) in (3.13) and ignoring the term $(\mathbf{M}\mathbf{x}(k) + \mathbf{S}_{a}\overline{\mathbf{T}}_{a} + \mathbf{S}_{g}\overline{\mathbf{T}}_{g} + \mathbf{S}_{dw}\overline{\mathbf{d}}_{w} + \mathbf{S}_{dz}\overline{\mathbf{d}}_{z})^{T}(\overline{\mathbf{C}})^{T}\mathbf{Q}_{1}\overline{\mathbf{C}}(\mathbf{M}\mathbf{x}(k) + \mathbf{S}_{a}\overline{\mathbf{T}}_{a} + \mathbf{S}_{g}\overline{\mathbf{T}}_{g} + \mathbf{S}_{dw}\overline{\mathbf{d}}_{w} + \mathbf{S}_{dz}\overline{\mathbf{d}}_{z})$ which is independent of $\overline{\mathbf{u}}$ and therefore does not affect the optimization, we obtain

$$J^{c} = \overline{\mathbf{u}}^{T} \mathbf{H}_{c} \overline{\mathbf{u}} + \mathbf{f}_{c}^{T} \overline{\mathbf{u}}$$
(3.16)

where,

where,

$$\mathbf{H}_{\mathbf{c}} = \mathbf{S}_{\mathbf{u}}^{T} \bar{\mathbf{C}}^{T} \mathbf{Q}_{\mathbf{1}} \bar{\mathbf{C}} \mathbf{S}_{\mathbf{u}} + \mathbf{Q}_{\mathbf{2}}, \qquad (3.17)$$

$$\mathbf{f}_{c} = 2\mathbf{S}_{u}^{T} \overline{\mathbf{C}}^{T} \mathbf{Q}_{1} \left[\overline{\mathbf{C}} \left(\mathbf{M} \mathbf{x}(\mathbf{k}) + \mathbf{S}_{a} \overline{\mathbf{T}}_{a} + \mathbf{S}_{g} \overline{\mathbf{T}}_{g} + \mathbf{S}_{dw} \overline{\mathbf{d}}_{w} + \mathbf{S}_{dz} \overline{\mathbf{d}}_{z} \right) - \overline{\mathbf{T}}_{z,ref} \right].$$
(3.18)

Therefore, the optimization problem corresponding to centralized MPC can be re-stated as the Quadratic Program (QP):

$$\overline{\mathbf{u}}_{\mathbf{c}}^* = \arg\min_{\overline{\mathbf{u}}} g_c(\overline{\mathbf{u}}). \tag{3.19}$$

$$g_c(\overline{\mathbf{u}}) = \overline{\mathbf{u}}^T \mathbf{H}_c \overline{\mathbf{u}} + \mathbf{f}_c^T \overline{\mathbf{u}}.$$
 (3.20)

3.4 Decentralized control framework

3.4.1 Architecture

We consider any general *p*-partition of a building (definition 3.2) with constitutive clusters S_{zi} , where $i \in 1, 2, ..., p$. A multi-agent MPC scheme is considered which is decentralized with respect to these clusters. In this architecture (Figure 3.7), each agent determines the control inputs for the corresponding cluster, based on the temperature measurements of zones and walls constituting the cluster, and appropriate forecasts of disturbances such as ambient temperature, ground temperature and thermal loads acting on the walls and zones constituting the cluster. The control decisions are still arrived at using a discrete time MPC approach, but the objective function and the model used for prediction are local to the cluster corresponding to a particular control agent.

3.4.2 Objective function

The objective function for the *i*th control agent, J_i^{dc} is obtained in (3.21) by extracting only those terms in $J^{c,perf}$ and $J^{c,cost}$ which correspond to the zones in the cluster S_{zi} .

$$J_i^{dc} = J_i^{dc,perf} + J_i^{dc,cost}$$
(3.21)

where,

$$J_{i}^{dc,perf} = \sum_{j=1}^{N} \left(\mathbf{T}_{\mathbf{z}\mathbf{i}}(k+j|k) - \mathbf{T}_{\mathbf{z}\mathbf{i},\mathbf{ref}}(k) \right)^{T} diag(\boldsymbol{\beta}_{\mathbf{i}}) \left(\mathbf{T}_{\mathbf{z}\mathbf{i}}(k+j|k) - \mathbf{T}_{\mathbf{z}\mathbf{i},\mathbf{ref}}(k) \right), (3.22)$$
$$J_{i}^{dc,cost} = \sum_{j=0}^{N-1} \mathbf{u}_{\mathbf{i}}(k+j|k)^{T} diag(\boldsymbol{\alpha}_{\mathbf{i}}) \mathbf{u}_{\mathbf{i}}(k+j|k). \tag{3.23}$$

In the above equations, α_i and β_i are vectors obtained from α and β respectively by extracting entries corresponding to the zones in the *i*th cluster.

3.4.3 Model

Similar to the centralized MPC framework, an appropriate model is required to characterize the effect of the cluster level control variables, i.e. the thermal energy transferred to each zone in a cluster by the HVAC system, on the cluster-level state variables of interest. This relationship is obtained by first recognizing the states (T_{wi} and T_{zi}), the control inputs, u_i and the

disturbances, \mathbf{d}_{wi} and \mathbf{d}_{zi} that are associated with the walls and zones constituting the cluster, and then characterizing their interdependencies by extracting suitable sub-matrices from the fullorder state space matrices in (3.4). Here, \mathbf{T}_{wi} is the vector of temperatures of all walls which are adjacent, in terms of the graph *G* described in Algorithm 1, to the zones constituting the *i*th cluster. The corresponding model for the *i*th cluster can be expressed in the form shown in (3.24), which uses the fact that $\mathbf{A}_{w,w}$, $\mathbf{A}_{z,z}$, $\mathbf{B}_{w,dw}$ and $\mathbf{B}_{z,dz}$ in (3.4) are diagonal matrices. The last term in the right hand side of (3.24) represents the influence that the zone temperatures in other clusters have on the dynamics of the *i*th cluster.



Figure 3.7 Schematic of decentralized control architecture

$$\begin{bmatrix} \mathbf{T}_{wi} \\ \mathbf{T}_{zi} \end{bmatrix} (k+1) = \underbrace{\begin{bmatrix} \mathbf{A}_{wi,wi} & \mathbf{A}_{wi,zi} \\ \mathbf{A}_{zi,wi} & \mathbf{A}_{zi,zi} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{T}_{wi} \\ \mathbf{T}_{zi} \end{bmatrix} (k) + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{zi} \end{bmatrix} \mathbf{u}_{i}(k)$$
$$+ \begin{bmatrix} \mathbf{B}_{wi,a} & \mathbf{B}_{wi,g} & \mathbf{B}_{wi,dw} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{zi,dz} \end{bmatrix} \begin{bmatrix} T_{a} \\ T_{g} \\ \mathbf{d}_{wi} \\ \mathbf{d}_{zi} \end{bmatrix} (k) + \sum_{j \in \{1,2,\dots,p\}, j \neq i} \begin{bmatrix} \mathbf{B}_{wi,zj} \\ \mathbf{0} \end{bmatrix} \mathbf{T}_{zj}(k) \quad (3.24)$$

Imposition of a control architecture that is decentralized with respect to the clusters implies that the i^{th} control agent does not have access to the sensory data from the other p - 1 clusters. Therefore, the $\mathbf{T}_{zj}(k)$, for each $j \in \{1, 2, ..., p\}, j \neq i$ appearing in (19) must be replaced with an appropriate guess or estimate, $\mathbf{\hat{T}}_{zj}(k)$. For example, if the operating temperature range of the building zones is known, say $[15^{\circ}C, 25^{\circ}C]$, the estimates can be heuristically chosen values which lie in this range. A specific choice for these estimates is the set-point temperatures for the corresponding zones, assuming that the controllers are able to satisfactorily regulate the temperatures around these set-points. Therefore, the appropriate model for use by the i^{th} control agent is given by

$$\begin{bmatrix} \mathbf{T}_{wi} \\ \mathbf{T}_{zi} \end{bmatrix} (k+1) = \underbrace{\begin{bmatrix} \mathbf{A}_{wi,wi} & \mathbf{A}_{wi,zi} \\ \mathbf{A}_{zi,wi} & \mathbf{A}_{zi,zi} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{T}_{wi} \\ \mathbf{T}_{zi} \end{bmatrix} (k) + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{zi} \end{bmatrix} \mathbf{u}_{i}(k)$$
$$+ \begin{bmatrix} \mathbf{B}_{wi,a} & \mathbf{B}_{wi,g} & \mathbf{B}_{wi,dw} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{zi,dz} \end{bmatrix} \begin{bmatrix} T_{a} \\ T_{g} \\ \mathbf{d}_{wi} \\ \mathbf{d}_{zi} \end{bmatrix} (k) + \sum_{j \in \{1,2,\dots,p\}, j \neq i} \begin{bmatrix} \mathbf{B}_{wi,zj} \\ \mathbf{0} \end{bmatrix} \widehat{\mathbf{T}}_{zj}(k). \quad (3.25)$$

3.4.4 Conversion to Quadratic Program

The optimization of the cost function J_i^{dc} for the i^{th} control agent can be converted to a QP by proceeding similarly as in the case of centralized MPC. Over the selected prediction horizon, the model given by (3.25) is used to predict the future states $\{\mathbf{T}_{wi}(k+l|k)\}_{l=1}^{l=N}$ and $\{\mathbf{T}_{zi}(k+l|k)\}_{l=1}^{l=N}$ in terms of any chosen current and future control inputs $\{\mathbf{u}_i(k+l|k)\}_{l=0}^{l=N-1}$, current state measurements $\mathbf{T}_{wi}(k), \mathbf{T}_{zi}(k)$, forecasted values of unmodeled thermal loads

$$\{\mathbf{d}_{wi}(k+l)\}_{l=0}^{l=N-1} \text{ and } \{\mathbf{d}_{zi}(k+l)\}_{l=0}^{l=N-1}, \text{ ambient temperatures } \{T_{a}(k+l)\}_{l=0}^{l=N-1}, \text{ ground temperatures } \{T_{g}(k+l)\}_{l=0}^{l=N-1}, \text{ and state estimates from other clusters } \{\widehat{\mathbf{T}}_{zi}(k+l)\}_{l=0}^{l=N-1}, \text{ ground temperatures } \{T_{g}(k+l)\}_{l=0}^{l=N-1}, \text{ and state estimates from other clusters } \{\widehat{\mathbf{T}}_{zi}(k+l)\}_{l=0}^{l=N-1}, \text{ ground temperatures } \{T_{a}(k+l)\}_{l=0}^{l=N-1}, \text{ and state estimates from other clusters } \{\widehat{\mathbf{T}}_{zi}(k+l)\}_{l=0}^{l=N-1}, \text{ ground temperatures } \{T_{a}(k+l)\}_{l=0}^{l=N-1}, p_{a}(k+l)\}_{l=0}^{l=N-1}, p_{a}(k+l)\}_{l=N-1}^{l=N-1}, p_{a}(k+l)\}_{l=0}^{l=N-1}, p_{a}(k+l)\}_{l=0}^{l=N-1}, p_{a}(k+l)\}_{l=0}^{l=N-1}, p_{a}(k+l)\}_{l=0}^{l=N-1}, p_{a}(k+l)\}_{l=0}^{l=N-1}, p_{a}(k+l)\}_{l=0}^{l=N-1}, p_{a}(k+l)\}_{l=0}^{l=N-1}, p_{a}(k+l)\}_{l=N-1}^{l=N-1}, p_{a}(k+l)\}_{l=N-1}$$

+
$$\mathbf{A}_{i}\mathbf{B}_{\mathbf{d}\mathbf{z},i}\mathbf{d}_{\mathbf{z}i}(k)$$
 + $\mathbf{B}_{\mathbf{d}\mathbf{z},i}\mathbf{d}_{\mathbf{z}i}(k+1|k)$ + $\sum_{j\neq i} \left(\mathbf{A}_{i}\mathbf{B}_{ij}\widehat{\mathbf{T}}_{\mathbf{z}j}(k) + \mathbf{B}_{ij}\widehat{\mathbf{T}}_{\mathbf{z}j}(k+1|k)\right)$
:

$$\mathbf{x}_{i}(k+N|k) = \mathbf{A}_{i}^{N}\mathbf{x}_{i}(k) + \sum_{i=0}^{N-1} \mathbf{A}_{i}^{N-1-i} [\mathbf{B}_{ui}\mathbf{u}_{i}(k+i|k) + \mathbf{B}_{ai}\mathbf{T}_{a}(k+i|k) + \mathbf{B}_{gi}\mathbf{T}_{g}(k+i|k) + \mathbf{B}_{dw,i}\mathbf{d}_{wi}(k+i|k) + \mathbf{B}_{dz,i}\mathbf{d}_{zi}(k+i|k) + \sum_{j\neq i} \mathbf{B}_{ij}\mathbf{\widehat{T}}_{zj}(k+i|k)]$$

The above equations can be succintly written as

$$\bar{\mathbf{x}}_{i} = \mathbf{M}_{i} \mathbf{x}_{i}(k) + \mathbf{S}_{ui} \overline{\mathbf{u}}_{i} + \mathbf{S}_{ai} \overline{\mathbf{T}}_{a} + \mathbf{S}_{gi} \overline{\mathbf{T}}_{g} + \mathbf{S}_{dw,i} \overline{\mathbf{d}}_{wi} + \mathbf{S}_{dz,i} \overline{\mathbf{d}}_{zi} + \sum_{j \neq i} \mathbf{S}_{ij} \overline{\mathbf{T}}_{zj}$$
(3.26)

Here,

$$\begin{split} \bar{\mathbf{x}}_{i} &= \begin{bmatrix} \mathbf{x}_{i}(k+1|k) \\ \mathbf{x}_{i}(k+2|k) \\ \vdots \\ \mathbf{x}_{i}(k+N|k) \end{bmatrix}, \ \bar{\mathbf{T}}_{a} &= \begin{bmatrix} \mathbf{T}_{a}(k) \\ \mathbf{T}_{a}(k+1|k) \\ \vdots \\ \mathbf{T}_{a}(k+N-1|k) \end{bmatrix}, \ \bar{\mathbf{T}}_{g} &= \begin{bmatrix} \mathbf{T}_{g}(k) \\ \mathbf{T}_{g}(k+1|k) \\ \vdots \\ \mathbf{T}_{g}(k+N-1|k) \end{bmatrix}, \\ \bar{\mathbf{d}}_{wi} &= \begin{bmatrix} \mathbf{d}_{wi}(k) \\ \mathbf{d}_{wi}(k+1|k) \\ \vdots \\ \mathbf{d}_{wi}(k+N-1|k) \end{bmatrix}, \ \bar{\mathbf{d}}_{zi} &= \begin{bmatrix} \mathbf{d}_{zi}(k) \\ \mathbf{d}_{zi}(k+1|k) \\ \vdots \\ \mathbf{d}_{zi}(k+N-1|k) \end{bmatrix}, \ \bar{\mathbf{T}}_{zj} &= \begin{bmatrix} \mathbf{T}_{g}(k) \\ \mathbf{T}_{g}(k+1|k) \\ \vdots \\ \mathbf{T}_{zj}(k+1|k) \\ \vdots \\ \mathbf{T}_{zj}(k+N-1|k) \end{bmatrix} \\ \mathbf{M}_{i} &= \begin{bmatrix} \mathbf{A}_{i} \\ \mathbf{A}_{i}^{2} \\ \vdots \\ \mathbf{A}_{i}^{N} \end{bmatrix}, \ \mathbf{S}_{ui} &= \begin{bmatrix} \mathbf{B}_{ui} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}_{i}\mathbf{B}_{ui} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{ui} & \mathbf{0} \\ \mathbf{A}_{i}^{N-1}\mathbf{B}_{ui} & \dots & \mathbf{A}_{i}\mathbf{B}_{ui} \\ \mathbf{B}_{ui} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{gi} & \mathbf{0} \\ \mathbf{A}_{i}^{N-1}\mathbf{B}_{gi} & \dots & \mathbf{A}_{i}\mathbf{B}_{gi} \\ \mathbf{B}_{gi} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{gi} & \mathbf{0} \\ \mathbf{A}_{i}^{N-1}\mathbf{B}_{gi} & \dots & \mathbf{A}_{i}\mathbf{B}_{gi} \\ \mathbf{B}_{dz,i} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{dz,i} \\ \mathbf{S}_{dz,i} &= \begin{bmatrix} \mathbf{B}_{dz,i} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}_{i}\mathbf{B}_{dz,i} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{dz,i} & \mathbf{0} \\ \mathbf{A}_{i}^{N-1}\mathbf{B}_{dz,i} & \dots & \mathbf{A}_{i}\mathbf{B}_{dz,i} \\ \mathbf{S}_{dz,i} &= \begin{bmatrix} \mathbf{B}_{dz,i} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}_{i}\mathbf{B}_{dz,i} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{dz,i} & \mathbf{0} \\ \mathbf{A}_{i}\mathbf{A}_{i}\mathbf{B}_{dz,i} & \dots & \mathbf{A}_{i}\mathbf{B}_{dz,i} \\ \mathbf{A}_{i}\mathbf{B}_{i}\mathbf{B}_{i} \\ \mathbf{A}_{i}^{N-1}\mathbf{B}_{ij} & \dots & \mathbf{A}_{i}\mathbf{B}_{ij} \\ \mathbf{B}_{ij} \end{bmatrix}, \text{and } \mathbf{S}_{ij} = \begin{bmatrix} \mathbf{B}_{ij} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}_{i}^{N-1}\mathbf{B}_{ij} & \dots & \mathbf{A}_{i}\mathbf{B}_{ij} \\ \mathbf{B}_{ij} \end{bmatrix}$$

From (3.21), J_i^{dc} can be re-stated as

$$J_i^{\ dc} = \left(\overline{\mathbf{T}}_{\mathbf{z}\mathbf{i}} - \overline{\mathbf{T}}_{\mathbf{z}\mathbf{i},\mathbf{ref}}\right)^T \mathbf{Q}_{\mathbf{1},\mathbf{i}} \left(\overline{\mathbf{T}}_{\mathbf{z}\mathbf{i}} - \overline{\mathbf{T}}_{\mathbf{z}\mathbf{i},\mathbf{ref}}\right) + \overline{\mathbf{u}}_{\mathbf{i}}^T \mathbf{Q}_{\mathbf{2},\mathbf{i}} \overline{\mathbf{u}}_{\mathbf{i}}, \qquad (3.27)$$

where,

$$\overline{\mathbf{T}}_{zi} = \begin{bmatrix} \mathbf{T}_{zi}(k+1|k) \\ \mathbf{T}_{zi}(k+2|k) \\ \vdots \\ \mathbf{T}_{zi}(k+N|k) \end{bmatrix}, \ \overline{\mathbf{T}}_{zi,ref} = \begin{bmatrix} \mathbf{T}_{zi,ref}(k) \\ \mathbf{T}_{zi,ref}(k) \\ \vdots \\ \mathbf{T}_{zi,ref}(k) \end{bmatrix}_{N \times N_{zi}},$$

$$\mathbf{Q}_{1,i} = \begin{bmatrix} diag(\boldsymbol{\beta}_{i}) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & diag(\boldsymbol{\beta}_{i}) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & diag(\boldsymbol{\beta}_{i}) \end{bmatrix}_{(N_{zi},N) \times (N_{zi},N)},$$

and
$$\mathbf{Q}_{2,\mathbf{i}} = \begin{bmatrix} diag(\mathbf{\alpha}_{i}) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & diag(\mathbf{\alpha}_{i}) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & diag(\mathbf{\alpha}_{i}) \end{bmatrix}_{(N_{Zi}\cdot N) \times (N_{Zi}\cdot N)}$$

Let $\mathbf{C}_{\mathbf{i}} = [\mathbf{0}_{N_{Zi} \times N_{Wi}} \quad \mathbf{I}_{N_{Zi}}]$ and $\overline{\mathbf{C}}_{\mathbf{i}} = \begin{bmatrix} \mathbf{C}_{\mathbf{i}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{i}} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{\mathbf{i}} \end{bmatrix}_{(N_{Zi}\cdot N) \times ((N_{Wi}+N_{Zi})\cdot N)}$.
Clearly, $\overline{\mathbf{T}}_{\mathbf{zi}} = \overline{\mathbf{C}}_{\mathbf{i}} \overline{\mathbf{x}}_{\mathbf{i}}$. (3.28)

Using (3.26), (3.28) can be written as

$$\overline{\mathbf{T}}_{\mathbf{z}\mathbf{i}} = \overline{\mathbf{C}}_{\mathbf{i}} \Big(\mathbf{M}_{\mathbf{i}} \mathbf{x}_{\mathbf{i}}(k) + \mathbf{S}_{\mathbf{u}\mathbf{i}} \overline{\mathbf{u}}_{\mathbf{i}} + \mathbf{S}_{a\mathbf{i}} \overline{\mathbf{T}}_{a} + \mathbf{S}_{g\mathbf{i}} \overline{\mathbf{T}}_{g} + \mathbf{S}_{dw,\mathbf{i}} \overline{\mathbf{d}}_{w\mathbf{i}} + \mathbf{S}_{dz,\mathbf{i}} \overline{\mathbf{d}}_{z\mathbf{i}} + \sum_{j \neq i} \mathbf{S}_{i\mathbf{j}} \overline{\mathbf{T}}_{z\mathbf{j}} \Big). \quad (3.29)$$

Substituting \overline{T}_{zi} from (3.29) in (3.27) and ignoring the terms which are independent of \overline{u}_i and therefore do not affect the optimization, we obtain

$$J_i^{\ dc} = \overline{\mathbf{u}}_i^{\ T} \mathbf{H}_{\mathbf{dc},i} \overline{\mathbf{u}}_i + \mathbf{f}_{\mathbf{dc},i}^{\ T} \overline{\mathbf{u}}_i$$
(3.30)

where,

$$\mathbf{H}_{\mathbf{dc},i} = \mathbf{S}_{\mathbf{u}i}^T \bar{\mathbf{C}}_i^T \mathbf{Q}_{\mathbf{1},i} \bar{\mathbf{C}}_i \mathbf{S}_{\mathbf{u}i} + \mathbf{Q}_{\mathbf{2},i}, \qquad (3.31)$$

$$\mathbf{f}_{\mathbf{dc},\mathbf{i}} = 2\mathbf{S}_{\mathbf{u}i}^T \mathbf{\bar{C}}_i^T \mathbf{Q}_{\mathbf{1},\mathbf{i}} \left[\mathbf{\bar{C}}_i \Big(\mathbf{M}_i \mathbf{x}_i(\mathbf{k}) + \mathbf{S}_{ai} \mathbf{\bar{T}}_a + \mathbf{S}_{gi} \mathbf{\bar{T}}_g + \mathbf{S}_{dw,i} \mathbf{\bar{d}}_{wi} + \mathbf{S}_{dz,i} \mathbf{\bar{d}}_z + \sum_{j \neq i} \mathbf{S}_{ij} \mathbf{\bar{T}}_{zj} \Big) - \mathbf{\bar{T}}_{zi,ref} \right]. (3.32)$$

Therefore, the optimization problem corresponding to decentralized MPC can be restated as the Quadratic Program (QP):

$$\overline{\mathbf{u}}_{dc,i}^* = \arg\min_{\overline{\mathbf{u}}_i} g_{dc,i}(\overline{\mathbf{u}}_i).$$
(3.33)

$$g_{dc,i}(\overline{\mathbf{u}}_{i}) = \overline{\mathbf{u}}_{i}^{T} \mathbf{H}_{dc,i} \overline{\mathbf{u}}_{i} + \mathbf{f}_{dc,i}^{T} \overline{\mathbf{u}}_{i}.$$
(3.34)

where,

Chapter 4 Control Architecture Selection for Building Thermal Control

4.1 Introduction

The role of control architecture in achieving the thermal control objectives was described in sections 1.2.1 and 3.1.2 In particular, the fundamental tradeoff between optimality and robustness with regard to the 'degree' of decentralization was examined. The focus of this chapter is on the development of appropriate decentralized control architecture selection methodologies which consider the afore-mentioned tradeoff between optimality and robustness. Two different approaches – CLF-MCS method and OLF-FPM method – are presented and demonstrated using examples. The reader is directed to sections 3.3.1 and 3.4.1 for details on centralized and decentralized frameworks, which form the basis for these methodologies.

4.2 CLF-MCS approach

4.2.1 Overview

The CLF-MCS approach is a procedure for partitioning a building based on Coupling Loss factor (CLF) and Mean Cluster Size (MCS) as optimality and robustness metrics respectively. A divisive clustering procedure employs these metrics to create a family of partitions using combinatorial optimization. These partitions are then analyzed to select those which provide satisfactory trade-offs between optimality and robustness. A MINCUT approximation for the underlying combinatorial optimization problem is also presented to address computational concerns. The methodology is demonstrated using simulated examples.

4.2.2 Coupling between inputs and clusters of inputs

The concepts of coupling between inputs and clusters of inputs are presented here as preliminaries used in the development of the Coupling Loss Factor as an optimality metric.

4.2.2.1 Coupling between inputs

We consider the centralized MPC framework described in Section 3.3. For a prediction horizon of length *N* samples, we introduce a vector $\bar{\mathbf{v}}$ defined at any time instant *k* as

$$\bar{\mathbf{v}} = \begin{bmatrix} \bar{\mathbf{v}}_1 \\ \bar{\mathbf{v}}_2 \\ \vdots \\ \bar{\mathbf{v}}_{N_z} \end{bmatrix}$$
(4.1)

where,

 $\bar{\mathbf{v}}_{i} = \begin{bmatrix} u_{i}(k|k) \\ u_{i}(k+1|k) \\ \vdots \\ u_{i}(k+N-1|k) \end{bmatrix} \text{ for } i \in \{1, 2, \dots, N_{z}\}.$ (4.2)

Here, u_i denotes the i^{th} component of the vector of control inputs **u**. The vector $\overline{\mathbf{u}}$ was defined in (3.12) as

$$\overline{\mathbf{u}} = \begin{bmatrix} \mathbf{u}(k|k) \\ \mathbf{u}(k+1|k) \\ \vdots \\ \mathbf{u}(k+N-1|k) \end{bmatrix}$$
(4.3)

Comparing (4.1) and (4.3), we observe that the entries of $\bar{\mathbf{v}}$ form a permutation of the entries of $\bar{\mathbf{u}}$. Therefore, $\bar{\mathbf{v}}$ can be expressed as shown in (4.4), where $\mathbf{P}_{\mathbf{vu}}$ is a permutation matrix.

$$\bar{\mathbf{v}} = \mathbf{P}_{\mathbf{v}\mathbf{u}}\bar{\mathbf{u}} \,. \tag{4.4}$$

Using (4.4) in (3.20), the objective function for centralized MPC can be written as

$$g_c(\bar{\mathbf{u}}) = g'_c(\bar{\mathbf{v}}) = \bar{\mathbf{v}}^T \overline{\mathbf{H}} \bar{\mathbf{v}} + \bar{\mathbf{f}}^T \bar{\mathbf{v}}$$
(4.5)

Where,

$$\overline{\mathbf{H}} = \mathbf{P}_{\mathbf{vu}} \mathbf{H}_{\mathbf{c}} \mathbf{P}_{\mathbf{vu}}^{-1}, \tag{4.6}$$

and,

$$\bar{\mathbf{f}} = \mathbf{P}_{\mathbf{vu}} \mathbf{f}_{\mathbf{c}}^T. \tag{4.7}$$

The quadratic part, $\overline{\mathbf{v}}^T \overline{\mathbf{H}} \overline{\mathbf{v}}$, can be written in the expanded form as:

$$\begin{bmatrix} \overline{\mathbf{v}}_1 \\ \overline{\mathbf{v}}_2 \\ \vdots \\ \overline{\mathbf{v}}_{N_z} \end{bmatrix}^T \begin{bmatrix} \overline{\mathbf{H}}_{1,1} & \overline{\mathbf{H}}_{1,2} & \dots & \overline{\mathbf{H}}_{1,N_z} \\ \overline{\mathbf{H}}_{2,1} & \overline{\mathbf{H}}_{2,2} & \dots & \overline{\mathbf{H}}_{2,N_z} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\mathbf{H}}_{N_z,1} & \overline{\mathbf{H}}_{N_z,2} & \dots & \overline{\mathbf{H}}_{N_z,N_z} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{v}}_1 \\ \overline{\mathbf{v}}_2 \\ \vdots \\ \overline{\mathbf{v}}_{N_z} \end{bmatrix}$$

Each off-diagonal term, $\overline{\mathbf{H}}_{i,j} \in \mathbb{R}^{N \times N}$, $i \neq j$ represents the coupling between u_i and u_j in $g_c(\overline{\mathbf{u}})$. Therefore, we use $\|\overline{\mathbf{H}}_{i,j}\|_2$ as a measure of coupling¹ between u_i and u_j and extend this to define coupling between between a pairs of input clusters in Section 4.2.2.2.

4.2.2.2 Coupling between clusters of inputs

Consider a p-partition of the building from definition 3.2. Consider any two clusters S_{zi} and S_{zj} from this partition. The coupling matrix between these clusters, $\overline{\mathbf{H}}_{S_{zi},S_{zi}}$ is defined as

where, $p_1, p_2 \dots \in S_{zi}$ and $q_1, q_2 \dots \in S_{zj}$.

The coupling $\mathcal{C}(S_{zi}, S_{z,j})$ between S_{zi} and $S_{z,j}$ is then defined as

$$\mathcal{C}(S_{zi}, S_{zj}) = \left\| \overline{\mathbf{H}}_{S_{zi}, S_{zj}} \right\|_{2}.$$
(4.9)

4.2.3 Overview of divisive clustering approach

The CLF-MCS clustering procedure is carried out in a divisive sequence as illustrated in Figure 4.1. The input to each stage is a set of *parent clusters*, and the output is a set of *child clusters*. The child clusters are obtained from the parent clusters via combinatorial analysis. The input to the first stage is the root cluster containing all the control inputs, which represents the completely centralized case. The output of the last stage is a set where each control input is a cluster by itself and hence represents a fully decentralized architecture. For any intermediate

¹ It is important to scale the system first so that coupling metrics corresponding to different pairs of input channels can be compared with one another. For a discussion on scaling see [83].

stage STG_i , the input (set of parent clusters) is the same as the output (set of child clusters) of the previous stage STG_{i-1} . Two metrics representing optimality and robustness are computed for each stage. A plot of one metric versus the other is then used to identify the stage which results in a satisfactory tradeoff between robustness and optimality.

4.2.4 Optimality and robustness metrics

Two dimensionless metrics - Coupling Loss Factor (CLF) and Mean Cluster Size (MCS) are computed for each partitioning stage.

4.2.4.1 Coupling Loss Factor

The CLF for stage STG_i is a normalized measure of the inter-cluster coupling among its child clusters that are denoted by $S_{z,j}^i$, where $j = 1, 2, ..., n_i$. Here, n_i is the total number of such child clusters. First, we introduce the coupling loss vector $\boldsymbol{\mu}_i$ for this stage STG_i as the vector of the couplings $\mathcal{C}(S_{z,m}^i, S_{z,n}^i)$ for each pair of child clusters, $(S_{z,m}^i, S_{z,n}^i)$ with $m \neq n$. More formally,

$$\mathbf{\mu}_{i} = \begin{pmatrix} \mu_{i,1} & \mu_{i,2} & \dots & \mu_{i,n_{i}-1} \end{pmatrix}^{T}$$
(4.10)

where, $\mu_{i,l} = \left[\mathcal{C}(S_{z,l}^{i}, S_{z,l+1}^{i}) \ \mathcal{C}(S_{z,l}^{i}, S_{z,l+2}^{i}) \ \dots \ \mathcal{C}(S_{z,l}^{i}, S_{z,n_{i}}^{i}) \right]$ for all $l \in \{1, 2, \dots, n_{i}\}$.

The CLF for stage STG_i , CLF_i is then defined as

$$CLF_{i} = \frac{\|\mu_{i}\|_{2}}{\|\bar{\mathbf{H}}\|_{2}}.$$
(4.11)

Here \mathbf{H} is the matrix defined in 4.6. The CLF for the parent partition to stage 1, which represents the fully centralized scenario, is clearly zero. CLF_i measures the coupling that gets ignored if the system were partitioned according to the child clusters of stage STG_i . Therefore, it is desired to partition the system such that the corresponding CLF is small so that the resulting deviation in optimality from centralized control is small.

PARTITIONING STAGES



Figure 4.1 Overview of the CLF-MCS clustering procedure

4.2.4.2 Mean Cluster Size

We use λ_j^i to denote the number of elements in child cluster $S_{z,j}^i$. MCS_i for stage STG_i is defined as the average number of zones per child cluster normalized with respect to the total number of zones.

$$MCS_{i} = \frac{\sum_{j=1}^{n_{i}} \lambda_{j}^{i}}{n_{i}N_{z}} = \frac{1}{n_{i}}.$$
 (4.12)

It is clear that $MCS_i \in (0,1]$. In a decentralized control architecture, the effect of a sensor or communication related fault is confined to the cluster where it originates. Therefore, the MCS is an indicator of robustness to such faults - a small value indicates that the number of clusters is large and thus the effect of failures is less widespread. Hence, it is desired to partition the building such that the corresponding MCS is small.

4.2.5 Stage level combinatorial optimization

As shown in Figure 4.1, for each stage, the input is a set of parent clusters and the output is a set of child clusters. The objective of the stage level optimization is to appropriately split the parent clusters to obtain corresponding child clusters. This process is based on a combinatorial optimization procedure which is explained below and illustrated in Figure 4.2.



Figure 4.2 Schematic of combinatorial optimization process for any given stage.

Since the parent clusters for any stage STG_i are the child clusters from its preceding stage STG_{i-1} , they are given by $S_{z,j}^{i-1}$ where $j = 1, 2, ..., n_{i-1}$. An *intermediate cluster pair* for any parent cluster is defined as a set of two non-empty clusters obtained by splitting that parent cluster. Therefore, the number of intermediate cluster pairs, $n_{z,j,int}^i$ obtained from the parent cluster $S_{z,j}^{i-1}$ is given by by the *Stirling number of the second kind* [84], $S(\lambda_j^{i-1}, 2)$ expressed as

$$n_{z,j,int}^{i} = S(\lambda_j^{i-1}, 2) = 2^{\lambda_j^{i-1} - 1} - 1.$$
(4.13)

The **Intermediate Loss Factor** (ILF) is defined for each intermediate cluster pair $\{S_{z,j,int,l}^{i}, S_{z,j,int,l}^{*i}\}$ of $S_{z,j}^{i-1}$, indexed by *l* as

$$\mathrm{ILF}_{z,j,int,l}^{i} = \frac{c(s_{z,j,int,l}^{i}, s_{z,j,int,l}^{*i})}{c(s_{z,j}^{i-1}, s_{z,j}^{i-1})}$$
(4.14)

where, $l = 1, 2, ..., n_{z,j,int}^i$, and $j = 1, 2, ..., n_{i-1}$.

The underlying optimization problem for the i^{th} stage, STG_i is to find the parent cluster (indicated by j^* , $j^* \in \{1, 2, ..., n_{i-1}\}$) and its corresponding intermediate cluster pair (indicated by $l^*, l^* \in \{1, 2, ..., n_{z,j,int}^i\}$) which yield the smallest ILF. This is stated as

$$\{j^*, l^*\} = \underset{\{j,l\}}{\operatorname{arg\,min}} \quad \operatorname{ILF}_{z,j,int,l}^i$$
 (4.15)

The optimal parent cluster is then split to create the optimal intermediate cluster pair, whereas the other parent clusters are retained. In other words, the cluster S_{z,j^*}^{i-1} is partitioned into the clusters S_{z,j^*,int,l^*}^i and S_{z,j^*,int,l^*}^{*i} . The result is a set of child clusters having one more cluster than the set of parent clusters.

The ILF defined in (4.14) measures the 'amount of coupling' information lost when a parent cluster is split into two child clusters, normalized with respect to the coupling originally present in the parent. Therefore, the above optimization involves determination of the optimal split in the sense that such a split results in the smallest loss of coupling information among all possible splits.

4.2.6 MINCUT approximation

The exponential computational complexity, characterized by (4.13), of the combinatorial optimization motivates the development of a more tractable approach for the minimization problem (4.15). In what follows, for simplicity, we denote the size λ_j^{i-1} of the parent cluster $S_{z,j}^{i-1}$ by *n*. The elements of $S_{z,j}^{i-1}$ are accordingly denoted by p_r , where r = 1, 2, ..., n.

A matrix $\overline{\mathbf{H}}_{j}$ is constructed for the j^{th} parent cluster $S_{z,j}^{i-1}$ from its elements in a manner analogous to the construction of the coupling matrix in (4.8)

$$\overline{\mathbf{H}}_{j} = \begin{pmatrix} \overline{\mathbf{H}}_{p_{1},p_{1}} & \overline{\mathbf{H}}_{p_{1},p_{2}} & \dots & \overline{\mathbf{H}}_{p_{1},p_{n}} \\ \overline{\mathbf{H}}_{p_{2},p_{1}} & \overline{\mathbf{H}}_{p_{2},p_{2}} & \dots & \overline{\mathbf{H}}_{p_{2},p_{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\mathbf{H}}_{p_{n},p_{1}} & \overline{\mathbf{H}}_{p_{n},p_{2}} & \dots & \overline{\mathbf{H}}_{p_{n},p_{n}} \end{pmatrix}.$$
(4.16)

For any given intermediate cluster pair, $\{S_{z,j,int,l}^i, S_{z,j,int,l}^{*i}\}$ a matrix, $\overline{\mathbf{H}}_j^l$ can be obtained from $\overline{\mathbf{H}}_j$ by setting to zero all blocks which correspond to elements in one intermediate cluster only. More precisely,

$$\bar{\mathbf{H}}_{j}^{l} = \begin{pmatrix} \theta_{p_{1},p_{1}} \bar{\mathbf{H}}_{p_{1},p_{1}} & \theta_{p_{1},p_{2}} \bar{\mathbf{H}}_{p_{1},p_{2}} & \dots & \theta_{p_{1},p_{n}} \bar{\mathbf{H}}_{p_{1},p_{n}} \\ \theta_{p_{2},p_{1}} \bar{\mathbf{H}}_{p_{2},p_{1}} & \theta_{p_{2},p_{2}} \bar{\mathbf{H}}_{p_{2},p_{2}} & \dots & \theta_{p_{2},p_{n}} \bar{\mathbf{H}}_{p_{2},p_{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{p_{n},p_{1}} \bar{\mathbf{H}}_{p_{n},p_{1}} & \theta_{p_{n},p_{1}} \bar{\mathbf{H}}_{p_{n},p_{2}} & \dots & \theta_{p_{n},p_{n}} \bar{\mathbf{H}}_{p_{n},p_{n}} \end{pmatrix},$$
(4.17)

where,
$$\theta_{p_r,p_s} = \begin{cases} 0 & \text{if } p_r, p_s \in S_{z,j,int,l}^i \text{ or } p_r, p_s \in S_{z,j,int,l}^{*i} \\ 1 & \text{otherwise.} \end{cases}$$
 (4.18)

Using the above definitions, $ILF_{z,j,int,l}^{i}$ defined in (4.14) can be expressed as

$$ILF_{z,j,int,l}^{i} = \frac{\left\|\bar{\mathbf{H}}_{j}^{l}\right\|_{2}}{\left\|\bar{\mathbf{H}}_{j}\right\|_{2}}.$$
(4.19)

From the above expression, the problem of minimizing $ILF_{z,j,int,l}^{i}$ over intermediate cluster pairs indexed by l, for a particular parent, denoted by a fixed j, corresponds to the minimization of $\|\overline{\mathbf{H}}_{j}^{l}\|_{2}$ over l. Assuming that $\overline{\mathbf{H}}_{j}^{l}$ is sufficiently sparse, we now approximate $\|\overline{\mathbf{H}}_{j}^{l}\|_{2}$ by the 2-norm of the vector \mathbf{v}_{j}^{l} consisting of the elements of $\overline{\mathbf{H}}_{j}^{l}$.

To make this procedure more formal, we introduce a binary vector $\mathbf{x} \in \mathbb{R}^n$ whose elements, \mathbf{x}_r ($r \in \{1, 2, ..., n\}$) are defined as follows:

$$\mathbf{x}_{r} = \begin{cases} 1 & \text{if } p_{r} \in S_{z,j,int,l}^{i} \\ -1 & \text{if } p_{r} \in S_{z,j,int,l}^{*i} \end{cases}$$
(4.20)

 $\left(\left\|\mathbf{v}_{j}^{l}\right\|_{2}\right)^{2}$ can be expressed as

$$\left(\left\|\mathbf{v}_{j}^{l}\right\|_{2}\right)^{2} = \frac{\mathbf{z}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}(\overline{\mathbf{H}}_{j})^{2}\mathbf{Q}\mathbf{z} - \mathbf{x}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}(\overline{\mathbf{H}}_{j})^{2}\mathbf{Q}\mathbf{x}}{2}$$
(4.21)

Here, $(\overline{\mathbf{H}}_j)^2$ denotes the matrix obtained by taking element-wise square of $\overline{\mathbf{H}}_j$. In other words, $(\overline{\mathbf{H}}_j)^2$ is the Hadamard product of $\overline{\mathbf{H}}_j$ with itself. **Q** and **z** are defined below.

$$\mathbf{Q} = \begin{pmatrix} \mathbf{e}_{N} & \mathbf{0}_{N \times 1} & \dots & \mathbf{0}_{N \times 1} \\ \mathbf{0}_{N \times 1} & \mathbf{e}_{N} & \dots & \mathbf{0}_{N \times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{N \times 1} & \mathbf{0}_{N \times 1} & \dots & \mathbf{e}_{N} \end{pmatrix}_{N.n \times n} \text{ where, } \mathbf{e}_{N} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{N \times 1} \text{ and } \mathbf{z} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{n \times 1}$$

Hence, the problem of minimizing $ILF_{z,j,int,l}^{i}$ over l for a particular parent j can be

approximated by the following Boolean maximization, which represents a MINCUT problem in graph theory:

maximize	$\mathbf{x}^T \mathbf{Q}^T (\overline{\mathbf{H}}_j)^2 \mathbf{Q} \mathbf{x}$
subject to	$\mathbf{x}_r \in \{1, -1\}$ for all $r \in \{1, 2,, n\}$

The above maximization can be performed using numerical techniques such as [85] available for solving the MINCUT problem. In this way, for each parent j, the minimum ILF can be found and compared across all parents to solve the original minimization problem (4.15).

4.2.7 Optimal partition selection

Since it is desired to have both CLF and MCS small, this problem is analogous to dual objective optimization in a pareto-optimal setting [82]. Motivated by this, the optimal partition is obtained from a plot of CLF_i versus MCS_i illustrated in Figure 4.3. The optimal partition should be a knee point. Therefore, navigating along the curve about that point in either direction would result in a large increase in one metric but only a relatively small decrease in the other metric. The plot must be studied in the ascending order of the partitioning stages (right to left) for knee points. For instance, if the first knee is not 'sufficiently' sharp, then the second knee (if any) should be studied.



Figure 4.3 Illustrative example of CLF vs. MCS plot.

4.2.8 Twelve zone building example

The layout of the building used in this example is shown in Figure 4.4. It consists of 3 floors, with a total of 12 rooms of equal dimensions $(5 \times 5 \times 5 m^3)$ numbered as shown. The building is assumed to be surrounded by ambient on all sides. The walls were modeled as RC circuits (Figure 3.5) based on the accessibility factor method described in [38], and the zones were modeled as isolated capacitors. Each room has 6 walls – 4 side walls, 1 ceiling and 1 floor. The construction details are presented in Table 4.1, from which the resistances and capacitances for the walls were computed as shown in Table 4.2. Note that in case of external walls, R₂ denotes the resistance between the wall and ambient. The zonal thermal capacities were assumed to be 250 kJ/K based on air at 25 C and 10⁵ Pa. An overall system model of the form (3.4) was obtained by constructing an RC network using these details, and then applying Algorithm 3.1 where discretization was performed using the zero-order-hold method with step size of 10 minutes. This choice was justified based on the fact that it was close to one-tenth of the smallest time constant in the model. The Hessian Matrix **H** was then created using (4.6) with prediction horizon, N = 12 samples (2 hours) and the weights $\alpha = 0.1e_{12}$ and $\beta = e_{12}$ (see nomenclature).



Figure 4.4 12-zone test building architecture (Zones are numbered)

Element	Layering	Thickness (m)	
	Brick	0.122	
	Insulation	0.050	
External walls	C-Block	0.112	
	Plaster	0.013	
	Plaster	0.013	
Internal walls	C-Block	0.122	
	Plaster	0.013	

 Table 4.1 Construction properties used for the 12-zone building model [40]

Table 4.2 Resistances and capacitances of walls (Refer to Figure 3.5)

Type of wall	C _{wall} (kJ/kg)	R_1 (K/kW)	$R_2(K/kW)^*$
Horizontal External	8329.15	29.99	81.08
Vertical External	8329.15	36.84	82.00
Horizontal Internal	4660.00	21.32	21.32
Vertical Internal	4660.00	21.32	21.32

As seen in Table 4.2, due to symmetry, the resistances corresponding to both horizontal and vertical internal walls have the same value $R_{nom} = 21.32$ K/kW. In the case study presented, the resistances of the horizontal internal walls (floor separators) are multiplied by a factor of ρ_1 to introduce thermal anisotropy. Similarly, the resistances of the *column separating* vertical internal walls (e.g. between zones 1 and 3, 2 and 4 etc.) are scaled by ρ_2 , and those of the *symmetrically splitting* vertical internal walls (e.g. between zones 1 and 2, 5 and 6, etc.) by ρ_3 .

The clustering methodology presented in Figure 4.1 was applied for specific values of ρ_i and the results have been summarized in Table 4.3. The relevant codes are provided in Appendix A. The combinatorial optimization in each stage was performed by comparing all possible intermediate cluster pairs. Evaluation of the MINCUT approximation is considered in a different example presented in section 4.2.9. The corresponding CLF vs. MCS plots are shown in Figure 4.5 to Figure 4.8. Important observations are as follows:

1. For the nominal case (case 1, Figure 4.5), a knee point is not obvious. Therefore, stage 3 was chosen to be the optimal partition where both CLF and MCS are satisfactorily small.

2. For cases 2, 3 and 4 the knee points identified for optimal partitioning (Figure 4.6 to Figure 4.8) are stages 3, 3 and 2 respectively. The corresponding clusters obtained (Table 4.3) can be explained on the basis of physical intuition. In case 2, the internal horizontal walls are more insulating than the vertical walls, therefore the building must be partitioned horizontally (along floors). In cases 3 and 4 this is reversed and a subset of vertical walls becomes more insulating. Therefore optimal clustering corresponds to partitions along such walls.

3. Figure 4.6 to Figure 4.8 indicate that clustering stages downstream of stage 3 are not optimal. This can be explained because these stages cause partitions along the low-insulation walls after separation along all high insulation walls has already been completed, therefore causing larger relative losses in coupling.

Case	$ ho_1$	$ ho_2$	$ ho_3$	optimal partition
1	1	1	1	{1,2,3,4,7,8}, {5,6,9,10}, {11,12}
2	3	1	1	$\{1,2,3,4,7,8\}, \{5,6,9,10\}, \{11,12\}$
3	1	3	1	$\{1,2\}, \{3,4,5,6\}, \{7,8,9,10,11,12\}$
4	1	1	3	$\{1,2,5,7,9\}, \{2,4,6,8,10\}$

Table 4.3 Summary of results for 12-zone test building



Figure 4.5 CLF vs. MCS plot for case 1 in Table 4.3



Figure 4.6 CLF vs. MCS plot for case 2 in Table 4.3



Figure 4.7 CLF vs. MCS plot for case 3 in Table 4.3



Figure 4.8 CLF vs. MCS plot for case 4 in Table 4.3

4.2.9 Nine zone building example

The layout of the building used in this example is shown in Figure 4.9. It consists of 3 floors, with a total of 9 rooms of equal dimensions $(5 \times 5 \times 5 \ m^3)$ numbered as shown. Similar to the previous example, the walls were modeled as RC circuits (Figure 3.5) and the zones were modeled as isolated capacitors. Each room has 6 walls - 4 side walls, 1 ceiling and 1 floor. The construction details and the resulting values of capacitances and resistances computed from them are same as for the 12-zone example, as presented in Table 4.1 and Table 4.2. The zonal thermal capacities were assumed to be 250 kJ/K based on air at 25 C and 10⁵ Pa. The dicrete-time system model for this system was then obtained in the same way as for the 12-zone building. The Hessian Matrix, $\overline{\mathbf{H}}$ was then created using (4.6) with prediction horizon, N = 24 samples (4 hours) and the weights $\boldsymbol{\alpha} = 0.1\mathbf{e}_9$ and $\boldsymbol{\beta} = \mathbf{e}_9$ (see nomenclature).



Figure 4.9 Layout of 9-zone building (side view)

The resistance values for all horizontal and vertical internal walls were found to be the same (see Table 4.2) as expected due to symmetry. We denote this value by $R_{nom} = 21.32$ K/kW. In the case study presented, the resistances of the horizontal internal walls are scaled up by a factor of $\rho > 0$, i.e. set to ρR_{nom} . Correspondingly, the resistances of the vertical internal walls are scaled down by a factor of ρ , i.e. set to R_{nom}/ρ . Therefore, the ratio of horizontal wall resistances and vertical wall resistances is augmented by ρ^2 .

The clustering methodology presented in Figure 4.1 was applied for various values of the factor ρ . The relvant codes are provided in the Appendix B. The optimization in each stage was performed using the combinatorial approach (involving all possible intermediate cluster pairs) as well as the MINCUT approximation presented in section 4.2.7. The CLF vs. MCS plots for some selected values of ρ are shown in Figure 4.10 to Figure 4.12. Key observations are as follows.

1. The CLF vs. MCS plots using the MINCUT procedure and the combinatorial procedure exactly coincide in Figure 4.11 and Figure 4.12. However, they differ in Figure 4.10. This suggests that the MINCUT approximation to the combinatorial optimization problem can be potentially accurate in asymmetric situations. In general, it trades accuracy for computational simplicity as indicated by a run-time of 0.29 seconds when compared to 4.92 seconds for the combinatorial procedure².

2. For the nominal case ($\rho = 1$), a knee point is not immediately obvious in Figure

 $^{^2}$ Values are for the case ho=1, implemented on a 2.0 GHz, 960 MB, AMD Athlon machine

4.10. Therefore, stage 3 was chosen to be the optimal partition where both CLF and MCS are satisfactorily small.

3. When $\rho > 1$ (Figure 4.11 and Figure 4.12), the optimal partition is provided by stage 3 since it corresponds to a 'sharp' knee. It corresponds to the clusters {1,2,3}, {4,5,6} and {7,8,9}. This can be justified on the basis of physical intuition. When $\rho > 1$, the horizontal walls are more insulating than the vertical walls, therefore the building must be sliced horizontally.



Figure 4.10 CLF vs. MCS plot for nine-zone building with $\rho = 1$



Figure 4.11 CLF vs. MCS plot for nine-zone building with $\rho = 2$


Figure 4.12 CLF vs. MCS plot for nine-zone building with $\rho = 3$

4.2.10 Remarks

The examples in Sections 4.2.8 and 4.2.9 demonstrate the capability of the CLF-MCS approach in determining architectures for decentralized control of a building that are consistent with physical intuition. In addition, Section 4.2.9 verifies that the proposed MINCUT approximation to address the high computational complexity of combinatorial optimization results in only a small loss in accuracy. However, the optimality and robustness metrics were chosen in a heuristic manner. The CLF defined in (4.11) does not directly quantify the loss in performance (optimality) in going from a centralized architecture to a decentralized architecture. Similarly, the MCS defined in (4.12) does not directly represent the true effect of fault propagation in the event of a sensor, actuator or communication infrastructure related failure in a building. Also, the divisive partitioning procedure in Figure 4.1 was found to have exponential computational complexity due to (4.13). The MINCUT approach which was proposed to address this issue is based on the assumption that $\overline{\mathbf{H}}_j^l$ is sufficiently sparse (see section 4.2.7). However, the desired level of sparsity is not quantified.

The above limitations of the CLF-MCS approach motivate the development of an approach which uses analytically derived metrics, and is computationally tractable with quantifiable computational complexity. The OLF-FPM approach presented in the next section satisfies these requirements and is proposed as an alternative to the CLF-MCS approach.

4.3 OLF-FPM method

4.3.1 Overview

The OLF-FPM approach is a procedure for partitioning a building based on Optimality Loss factor (OLF) and Fault Propagation Metric (FPM) as optimality and robustness metrics respectively. An agglomerative clustering procedure employs these metrics to create a family of partitions. Similar to the CLF-MCS approach, these partitions are then analyzed to select those which provide satisfactory trade-offs between optimality and robustness. The partitioning methodology is demonstrated using simulated examples. The optimality analysis of centralized and decentralized MPC frameworks and their comparison are presented first as necessary prerequisites for the development of OLF. The nomenclature in Chapter 3 is used several times in the remainder of Section 4.3. Additional nomenclature used is listed in Table 4.4

Symbol	Description
\mathbf{e}_n	Vector of size $n \times 1$ with all entries 1
$T_{z,i}$	Temperature of <i>i</i> th zone in building
$C_{z,i}$	Capacitance of <i>i</i> th zone in building
$d_{z,i}$	Unmodeled thermal disturbance acting on i^{th} zone in building
$d_{w,i}$	Unmodeled thermal disturbance acting on i^{th} wall in building

Table 4.4: Nomenclature of additional symbols used (besides Table 3.1)

4.3.2 Optimality analysis for centralized MPC

Assuming that α and β in (3.2) and (3.3) are component-wise positive vectors, the objective function J^c in (3.1) is strictly convex in the $(2N_z + N_w)N$ dimensional space of real variables consisting of components of $\{\mathbf{u}(k+l|k)\}_{l=0}^{l=N-1}, \{\mathbf{T}_z(k+l|k)\}_{l=1}^{l=N}$ and $\{\mathbf{T}_w(k+l|k)\}_{l=1}^{l=N}$. The linear constraint (3.4) represents hyperplanes in this space. Therefore, the optimization problem corresponding to centralized MPC is strictly convex with a unique global

minimum [82]. In terms of the unconstrained re-statement (3.19), this implies that $g_c(.)$ is a strictly convex function of $\overline{\mathbf{u}}$, with a positive definite invertible Hessian Matrix, \mathbf{H}_c in (3.20). The fact that \mathbf{H}_c is positive definite can also be verified from (3.17) by noting that \mathbf{Q}_1 and \mathbf{Q}_2 are positive definite matrices when α and β are component-wise positive vectors. The closed form expression for the unique minimizer, $\overline{\mathbf{u}}_c^*$ of $g_c(.)$ in (3.19) is given by

$$\overline{\mathbf{u}}_{\mathbf{c}}^{*} = -\frac{1}{2}\mathbf{H}_{\mathbf{c}}^{-1}\mathbf{f}_{\mathbf{c}}$$
$$= -\mathbf{H}_{\mathbf{c}}^{-1}\mathbf{S}_{\mathbf{u}}^{T}\overline{\mathbf{C}}^{T}\mathbf{Q}_{1}\left[\overline{\mathbf{C}}\left(\mathbf{M}\mathbf{x}(\mathbf{k}) + \mathbf{S}_{\mathbf{a}}\overline{\mathbf{T}}_{\mathbf{a}} + \mathbf{S}_{\mathbf{g}}\overline{\mathbf{T}}_{\mathbf{g}} + \mathbf{S}_{\mathbf{d}w}\overline{\mathbf{d}}_{\mathbf{w}} + \mathbf{S}_{\mathbf{d}z}\overline{\mathbf{d}}_{z}\right) - \overline{\mathbf{T}}_{z,ref}\right]. \quad (4.22)$$

A transformed form $\bar{\mathbf{v}}_{\mathbf{c}}^*$ is defined for $\bar{\mathbf{u}}_{\mathbf{c}}^*$ which has the structure given by (4.1). Using (4.4), $\bar{\mathbf{v}}_{\mathbf{c}}^*$ can be expressed as

$$\bar{\mathbf{v}}_{\mathbf{c}}^* = \mathbf{P}_{\mathbf{v}\mathbf{u}}\bar{\mathbf{u}}_{\mathbf{c}}^* \tag{4.23}$$

Substituting (4.23) in (4.22), we obtain

$$\bar{\mathbf{v}}_{c}^{*} = -\mathbf{P}_{vu}\mathbf{H}_{c}^{-1}\mathbf{S}_{u}^{T}\bar{\mathbf{C}}^{T}\mathbf{Q}_{1}\left[\bar{\mathbf{C}}\left(\mathbf{M}\mathbf{x}(\mathbf{k}) + \mathbf{S}_{a}\bar{\mathbf{T}}_{a} + \mathbf{S}_{g}\bar{\mathbf{T}}_{g} + \mathbf{S}_{dw}\bar{\mathbf{d}}_{w} + \mathbf{S}_{dz}\bar{\mathbf{d}}_{z}\right) - \bar{\mathbf{T}}_{z,ref}\right]. \quad (4.24)$$

Equation (4.24) can be written as

$$\bar{\mathbf{v}}_{c}^{*} = -\frac{1}{2}\bar{\mathbf{H}}_{c}^{-1}\left[\mathbf{K}_{1}^{c}\mathbf{T}_{z}(k) + \mathbf{K}_{2}^{c}\mathbf{T}_{w}(k) + \mathbf{K}_{3}^{c}\bar{\mathbf{d}}_{w} + \mathbf{K}_{4}^{c}\bar{\mathbf{d}}_{z} + \mathbf{K}_{5}^{c}\overline{\mathbf{T}}_{a} + \mathbf{K}_{6}^{c}\overline{\mathbf{T}}_{g} + \mathbf{K}_{7}^{c}\overline{\mathbf{T}}_{z,ref}(k)\right]. \quad (4.25)$$

where,

$$\overline{\mathbf{H}}_c = \mathbf{H}_c \mathbf{P}_{\mathbf{vu}}^{-1}, \tag{4.26}$$

$$\mathbf{K_1^c} = 2\mathbf{S_u^T} \bar{\mathbf{C}}^T \mathbf{Q_1} \bar{\mathbf{C}} \begin{bmatrix} \mathbf{I}_{N_w} \\ \mathbf{0}_{N_z \times N_w} \end{bmatrix}, \qquad (4.27)$$

$$\mathbf{K}_{2}^{\mathbf{c}} = 2\mathbf{S}_{\mathbf{u}}^{T} \overline{\mathbf{C}}^{T} \mathbf{Q}_{1} \overline{\mathbf{C}} \begin{bmatrix} \mathbf{0}_{N_{w} \times N_{z}} \\ I_{N_{z}} \end{bmatrix}, \qquad (4.28)$$

$$\mathbf{K}_{\mathbf{3}}^{\mathbf{c}} = 2\mathbf{S}_{\mathbf{u}}^{T} \bar{\mathbf{C}}^{T} \mathbf{Q}_{\mathbf{1}} \mathbf{S}_{\mathbf{dw}}, \qquad (4.29)$$

$$\mathbf{K_4^c} = 2\mathbf{S_u^T} \bar{\mathbf{C}}^T \mathbf{Q_1} \mathbf{S_{dz}},\tag{4.30}$$

$$\mathbf{K}_{\mathbf{5}}^{\mathbf{c}} = 2\mathbf{S}_{\mathbf{u}}^{T} \bar{\mathbf{C}}^{T} \mathbf{Q}_{\mathbf{1}} \mathbf{S}_{\mathbf{a}}, \qquad (4.31)$$

$$\mathbf{K}_{\mathbf{6}}^{\mathbf{c}} = 2\mathbf{S}_{\mathbf{u}}^{T} \overline{\mathbf{C}}^{T} \mathbf{Q}_{\mathbf{1}} \mathbf{S}_{\mathbf{g}}, \tag{4.32}$$

$$\mathbf{K}_{\mathbf{7}}^{\mathbf{c}} = -2\mathbf{S}_{\mathbf{u}}^{T} \overline{\mathbf{C}}^{T} \mathbf{Q}_{\mathbf{1}}.$$
(4.33)

4.3.3 Extraction matrices for decentralized MPC

Let $\mathbf{x} \in \mathbb{R}^{N_z}$ be a vector, the l^{th} component, \mathbf{x}^l of which is the value of an appropriate physical quantity associated with the l^{th} zone in the building, where $l \in \{1, 2, ..., N_z\}$. In the present context, \mathbf{x} could represent a vector of temperatures, control inputs or unmodeled thermal loads for the various zones. Next, consider a general p-partition of the building consisting of clusters S_{zi} , where $i \in \{1, 2, ..., p\}$. For each $i \in \{1, 2, ..., p\}$, let $\mathbf{x}_i \in \mathbb{R}^{N_{zi}}$ be a vector such that its r^{th} component, \mathbf{x}_i^r , where $r \in \{1, 2, ..., N_{zi}\}$ is the value of the afore-mentioned physical quantity for the r^{th} zone in the cluster S_{zi} . Therefore, the set of elements of \mathbf{x}_i is a subset of the elements of \mathbf{x} . However, a sequential stack of \mathbf{x}_i given by $(\mathbf{x_1}^T \mathbf{x_2}^T \dots \mathbf{x_p}^T)^T$ does not necessarily produce \mathbf{x} . This is also true in the context of vectors $\mathbf{y} \in \mathbb{R}^{N_w}$ and $\mathbf{y}_i \in \mathbb{R}^{N_{wi}}$, which are analogues of \mathbf{x} and \mathbf{x}_i respectively, but are defined for walls instead of zones. Here, we introduce the concept of *extraction matrices* to enable an accurate representation of the mathematical relationship between the cluster-level vectors and the overall system level vectors, and extend it to include the case when these vectors are lifted in time. This concept is required for the optimality analysis of decentralized MPC that follows in the next section.

Definition 4.1 (Cluster extraction matrix): Let $\mathbf{s}_z \in \mathbb{N}^{N_z}$ and $\mathbf{s}_{zi} \in \mathbb{N}^{N_{zi}}$ be vector representations of the elements of the sets S_z and S_{zi} respectively, where $i \in \{1, 2, ..., p\}$. The *cluster extraction matrix*, $\mathbf{P}_i \in \mathbb{Z}^{N_{zi} \times N_z}$ for the *i*th cluster, S_{zi} in the p-partition is defined as the Boolean matrix with exactly one 1 in each row which satisfies

$$\mathbf{s}_{\mathbf{z}i} = \mathbf{P}_i \mathbf{s}_{\mathbf{z}}.\tag{4.34}$$

Definition 4.2 (Lifted cluster extraction matrix): The *lifted cluster extraction matrix*, $\overline{\mathbf{P}}_i \in \mathbb{Z}^{N.N_{zi} \times N.N_z}$ for the *i*th cluster, S_{zi} in the p-partitionis defined as the boolean matrix which is obtained from \mathbf{P}_i by replacing all scalar ones with \mathbf{I}_N and all scalar zeros with $\mathbf{0}_{N \times N}$.

Definition 4.3 (Overall extraction matrix): The *overall extraction matrix*, $\mathbf{P} \in \mathbb{Z}^{N_z \times N_z}$ is defined in (4.35) by stacking the matrices \mathbf{P}_i , $i \in \{1, 2, ..., p\}$ along their columns.

$$\mathbf{P} = \left(\mathbf{P}_1^T \ \mathbf{P}_2^T \ \dots \ \mathbf{P}_p^T\right)^T. \tag{4.35}$$

Definition 4.4 (Overall lifted extraction matrix): The overall lifted extraction matrix, $\overline{\mathbf{P}} \in$

 $\mathbb{Z}^{N.N_Z \times N.N_Z}$ is defined in (4.36) by stacking the matrices $\overline{\mathbf{P}}_i$, $i \in \{1, 2, ..., p\}$ along their columns.

$$\overline{\mathbf{P}} = \left(\overline{\mathbf{P}}_1^T \ \overline{\mathbf{P}}_2^T \ \dots \ \overline{\mathbf{P}}_p^T\right)^T.$$
(4.36)

Definition 4.5 (Wall extraction matrix): Let $\mathbf{s}_{\mathbf{w}} \in \mathbb{N}^{N_w}$ be a vector representing the set of walls in the building. Similarly, let $\mathbf{s}_{wi} \in \mathbb{N}^{N_{wi}}$ be a vector representing the set of walls in the cluster S_{zi} . The *wall extraction matrix*, $\mathbf{R}_i \in \mathbb{Z}^{N_{wi} \times N_w}$ for the *i*th cluster, S_{zi} in the p-partition is defined as the Boolean matrix with exactly one 1 in each row which satisfies

$$\mathbf{s}_{\mathbf{w},i} = \mathbf{R}_i \mathbf{s}_{\mathbf{w}}.\tag{4.37}$$

Definition 4.6. (Overall wall extraction matrix): The overall wall extraction matrix, $\mathbf{R} \in \mathbb{Z}^{(\sum_{i=1}^{p} N_{w,i}) \times N_{w}}$ is defined in (4.38) by stacking the matrices \mathbf{R}_{i} , $i \in \{1, 2, ..., p\}$ along their columns.

$$\mathbf{R} = \left(\mathbf{R}_1^T \ \mathbf{R}_2^T \ \dots \ \mathbf{R}_p^T\right)^T. \tag{4.38}$$

Definition 4.7 (Lifted wall extraction matrix): The *lifted wall extraction matrix*, $\overline{\mathbf{R}}_i \in \mathbb{Z}^{N.N_{wi} \times N.N_w}$ for the *i*th cluster, S_{zi} in the p-partition is defined as the Boolean matrix which is obtained from \mathbf{R}_i by replacing all scalar ones with \mathbf{I}_N and all scalar zeros with $\mathbf{O}_{N \times N}$.

Definition 4.8 (Overall lifted wall extraction matrix): The *overall lifted wall extraction matrix*, $\overline{\mathbf{R}} \in \mathbb{Z}^{(\sum_{i=1}^{p} N.N_{w,i}) \times N.N_{w}}$ is defined in (4.39) by stacking the matrices $\overline{\mathbf{R}}_{i}$, $i \in \{1, 2, ..., p\}$ along their columns.

$$\overline{\mathbf{R}} = \left(\overline{\mathbf{R}}_1^T \ \overline{\mathbf{R}}_2^T \ \dots \ \overline{\mathbf{R}}_p^T\right)^T. \tag{4.39}$$

As an example, consider the 3-zone building in Figure 3.3. We demonstrate the above definitions using the 2-partition of this building $\{\{1,3\},\{2\}\}$. Here, $S_z = \{1,2,3\}$, $S_{z1} = \{1,3\}$ and $S_{z2} = \{2\}$. Therefore, $\mathbf{s_z} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T$, $\mathbf{s_{z1}} = \begin{bmatrix} 1 & 3 \end{bmatrix}^T$ and $\mathbf{s_{z2}} = 2$.

The cluster extraction matrices are given by:

$$\mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \mathbf{P}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

The lifted cluster extraction matrices are given by:

$$\overline{\mathbf{P}}_1 = \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} & \mathbf{I}_N \end{bmatrix}, \ \overline{\mathbf{P}}_2 = \begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{I}_N & \mathbf{0}_{N \times N} \end{bmatrix}.$$

The overall extraction matrix and overall lifted extraction matrix are then given by:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \overline{\mathbf{P}} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} & \mathbf{I}_N \\ \mathbf{0}_{N \times N} & \mathbf{I}_N & \mathbf{0}_{N \times N} \end{bmatrix}.$$

Using the graph shown in Figure 3.6, the vector of walls in the building is given by $\mathbf{s_w} = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10]^T$, $\mathbf{s_{w1}} = [1 \ 2 \ 3 \ 4 \ 7 \ 8 \ 9 \ 10]^T$ and $\mathbf{s_{w2}} = [4 \ 5 \ 6 \ 7]^T$.

Hence, the wall extraction matrices are given by:

R ₁ =	$ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0$	0 1 0 0 0 0 0 0	0 0 1 0 0 0 0	0 0 1 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 1 0 0	0 0 0 0 0 1 0	0 0 0 0 0 0 1	0 0 0 0 0 0 0	, R ₂ =	$\begin{bmatrix} 0\\0\\0\\0\\0 \end{bmatrix}$	0 0 0 0	0 0 0 0	1 0 0 0	0 1 0 0	0 0 1 0	0 0 0 1	0 0 0 0	0 0 0 0	$\begin{pmatrix} 0\\0\\0\\0 \end{bmatrix}$
	$\begin{bmatrix} 0\\0 \end{bmatrix}$	0	0	0	0	0	0	0	0	$\begin{bmatrix} 0\\1 \end{bmatrix}$											

The overall wall extraction matrix is given by:

	г1	0	0	0	0	0	0	0	0	ך0	
	0	1	0	0	0	0	0	0	0	0	
	0	0	1	0	0	0	0	0	0	0	
	0	0	0	1	0	0	0	0	0	0	
	0	0	0	0	0	0	1	0	0	0	
R =	0	0	0	0	0	0	0	1	0	0	
	0	0	0	0	0	0	0	0	1	0	·
	0	0	0	0	0	0	0	0	0	1	
	0	0	0	1	0	0	0	0	0	0	
	0	0	0	0	1	0	0	0	0	0	
	0	0	0	0	0	1	0	0	0	0	
	LO	0	0	0	0	0	1	0	0	01	

The *lifted wall extraction matrices* $\overline{\mathbf{R}}_1$ and $\overline{\mathbf{R}}_2$ are obtained from \mathbf{R}_1 and \mathbf{R}_2 respectively by replacing all scalar ones with \mathbf{I}_N and all scalar zeros with $\mathbf{0}_{N \times N}$. The *overall lifted wall extraction matrix* $\overline{\mathbf{R}}$ is obtained from \mathbf{R} in a similar manner.

We state some easily verifiable properties of the extraction matrices in terms of \mathbf{x} , \mathbf{x}_i , \mathbf{y}

and \mathbf{y}_i defined earlier, and their time-lifted analogues³ $\mathbf{\bar{x}}$, $\mathbf{\bar{x}}_i$, $\mathbf{\bar{y}}$ and $\mathbf{\bar{y}}_i$ defined in (4.40) to (4.43).

$$\bar{\mathbf{x}} = (\bar{\mathbf{x}}^1 \ \bar{\mathbf{x}}^2 \ \dots \ \bar{\mathbf{x}}^{N_z})^T \text{ where, } \bar{\mathbf{x}}^l = [\mathbf{x}^l(k|k) \ \mathbf{x}^l(k+1|k) \ \dots \ \mathbf{x}^l(k+N-1|k)], \quad (4.40)$$

$$\bar{\mathbf{x}}_i = \left(\bar{\mathbf{x}}_i^1 \ \bar{\mathbf{x}}_i^2 \ \dots \ \bar{\mathbf{x}}_i^{N_{zi}}\right)^r \text{ where, } \bar{\mathbf{x}}_i^r = \left[\mathbf{x}_i^r(k|k) \ \mathbf{x}_i^r(k+1|k) \ \dots \ \mathbf{x}_i^r(k+N-1|k)\right], (4.41)$$

$$\overline{\mathbf{y}} = (\overline{\mathbf{y}}^1 \ \overline{\mathbf{y}}^2 \ \dots \ \overline{\mathbf{y}}^{N_w})^T \text{ where, } \overline{\mathbf{y}}^l = [\mathbf{y}^l(k|k) \ \mathbf{y}^l(k+1|k) \ \dots \ \mathbf{y}^l(k+N-1|k)], \quad (4.42)$$

$$\bar{\mathbf{y}}_i = \left(\bar{\mathbf{y}}_i^1 \ \bar{\mathbf{y}}_i^2 \ \dots \ \bar{\mathbf{y}}_i^{N_{zi}}\right)^T \text{ where, } \bar{\mathbf{y}}_i^r = \left[\mathbf{y}_i^r(k|k) \ \mathbf{y}_i^r(k+1|k) \ \dots \ \mathbf{y}_i^r(k+N-1|k)\right]. (4.43)$$

Properties of extraction matrices

1. $\mathbf{x}_i = \mathbf{P}_i \mathbf{x}$. 2. $\mathbf{\bar{x}}_i = \mathbf{\overline{P}}_i \mathbf{\bar{x}}$. 3. $(\mathbf{x}_1^T \ \mathbf{x}_2^T \ \dots \ \mathbf{x}_p^T)^T = \mathbf{P}\mathbf{x}$. 4. $(\mathbf{\bar{x}}_1^T \ \mathbf{\bar{x}}_2^T \ \dots \ \mathbf{\bar{x}}_p^T)^T = \mathbf{\overline{P}}\mathbf{\bar{x}}$. 5. Both \mathbf{P} and $\mathbf{\overline{P}}$ are invertible permutation matrices. 6. $\mathbf{y}_i = \mathbf{R}_i \mathbf{y}$.

7.
$$\overline{\mathbf{y}}_i = \overline{\mathbf{R}}_i \overline{\mathbf{y}}.$$

8. $(\mathbf{y}_1^T \ \mathbf{y}_2^T \ \dots \ \mathbf{y}_p^T)^T = \mathbf{R}\mathbf{y}.$

9.
$$(\bar{\mathbf{y}}_1^T \ \bar{\mathbf{y}}_2^T \dots \bar{\mathbf{y}}_p^T)^T = \overline{\mathbf{R}}\bar{\mathbf{y}}$$
.

4.3.4 Optimality analysis for decentralized MPC

The components of the vectors $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ in (3.22) and (3.23) are positive if $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in (3.2) and (3.3) are component-wise positive. Therefore, as observed for the centralized MPC optimization problem, the optimization problem for the *i*th control agent in the decentralized control framework (3.34) is also strictly convex. The closed form expression for the unique minimizer $\bar{\mathbf{u}}_{dc,i}^*$ of $g_{dc,i}(.)$ in (3.33) is given by

$$-2\mathbf{H}_{\mathbf{dc},\mathbf{i}}\overline{\mathbf{u}}_{\mathbf{dc},\mathbf{i}}^{*} = \mathbf{f}_{\mathbf{dc},\mathbf{i}}$$
$$= 2\mathbf{S}_{\mathbf{u}i}^{T}\overline{\mathbf{C}}_{\mathbf{i}}^{T}\mathbf{Q}_{\mathbf{1},\mathbf{i}} \Big[\overline{\mathbf{C}}_{\mathbf{i}}\Big(\mathbf{M}_{\mathbf{i}}\mathbf{x}_{\mathbf{i}}(\mathbf{k}) + \mathbf{S}_{\mathbf{a}i}\overline{\mathbf{T}}_{\mathbf{a}} + \mathbf{S}_{\mathbf{g}i}\overline{\mathbf{T}}_{\mathbf{g}} + \mathbf{S}_{\mathbf{dw},\mathbf{i}}\overline{\mathbf{d}}_{\mathbf{w}\mathbf{i}} + \mathbf{S}_{\mathbf{dz},\mathbf{i}}\overline{\mathbf{d}}_{\mathbf{z}} + \sum_{j\neq i} \mathbf{S}_{\mathbf{i}j}\overline{\mathbf{T}}_{\mathbf{z}j}\Big) - \overline{\mathbf{T}}_{\mathbf{z}\mathbf{i},\mathbf{ref}}\Big] (4.44)$$

³If x is used to represent the zone temperatures, $\overline{\mathbf{x}}^l$ in (32) should instead be defined as

 $[\]overline{\mathbf{x}}^{l} = (\mathbf{x}^{l}(k+1|k) \mathbf{x}^{l}(k+2|k) \dots \mathbf{x}^{l}(k+N|k))$. The definition of $\overline{\mathbf{x}}_{i}^{r}$ in ((3.26)) should be similarly modified.

The vector $\overline{u}_{dc,i}^*$ is a lifted vector given by

$$\overline{\mathbf{u}}_{\mathbf{dc},\mathbf{i}}^{*} = \begin{bmatrix} \mathbf{u}_{\mathbf{dc},\mathbf{i}}^{*}(k|k) \\ \mathbf{u}_{\mathbf{dc},\mathbf{i}}^{*}(k+1|k) \\ \vdots \\ \mathbf{u}_{\mathbf{dc},\mathbf{i}}^{*}(k+N-1|k) \end{bmatrix}, \text{ where } \mathbf{u}_{\mathbf{dc},\mathbf{i}}^{*}(k+l|k) \in \mathbb{R}^{N_{zi}} \text{ for all } l \in \{0,1,\dots,N-1\}. (4.45)$$

We now introduce a vector $\bar{v}_{dc,i}^*$ whose elements are permutations of the elements of $\bar{u}_{dc,i}^*$ as shown

$$\bar{\mathbf{v}}_{\mathbf{dc},\mathbf{i}}^{*} = \begin{bmatrix} \bar{\mathbf{v}}_{\mathbf{dc},\mathbf{i}}^{*1} \\ \bar{\mathbf{v}}_{\mathbf{dc},\mathbf{i}}^{*2} \\ \vdots \\ \bar{\mathbf{v}}_{\mathbf{dc},\mathbf{i}}^{*N_{zi}} \end{bmatrix}, \qquad (4.46)$$

where,

 $\bar{\mathbf{v}}_{\mathbf{dc},\mathbf{i}}^{*r} = \begin{bmatrix} u_{dc,i}^{*r}(k|k) \\ u_{dc,i}^{*r}(k+1|k) \\ \vdots \\ u_{dc,i}^{*r}(k+N-1|k) \end{bmatrix} \text{ for } r \in \{1, 2, \dots, N_{zi}\}. \quad (4.47)$ Here, $u_{dc,i}^{*r}$ denotes the r^{th} component of the vector of control inputs $\mathbf{u}_{dc,i}^{*}$. Therefore, the elements of $\bar{v}_{dc,i}^*$ and $\overline{u}_{dc,i}^*$ are permutations of each other. Let $P_{vu,i}$ be a permutation matrix such that

$$\overline{\mathbf{v}}_{\mathbf{dc},\mathbf{i}}^* = \mathbf{P}_{\mathbf{vu},\mathbf{i}}\overline{\mathbf{u}}_{\mathbf{dc},\mathbf{i}}^* . \tag{4.48}$$

Substituting (4.48) in (4.44), we obtain

$$-2\mathbf{H}_{\mathbf{dc},\mathbf{i}}\mathbf{P}_{\mathbf{vu},\mathbf{i}}^{-1}\overline{\mathbf{v}}_{\mathbf{dc},\mathbf{i}}^{*} = \\2\mathbf{S}_{\mathbf{u}\mathbf{i}}^{T}\overline{\mathbf{C}}_{\mathbf{i}}^{T}\mathbf{Q}_{\mathbf{1},\mathbf{i}}\left[\overline{\mathbf{C}}_{\mathbf{i}}\left(\mathbf{M}_{\mathbf{i}}\mathbf{x}_{\mathbf{i}}(\mathbf{k}) + \mathbf{S}_{a\mathbf{i}}\overline{\mathbf{T}}_{a} + \mathbf{S}_{g\mathbf{i}}\overline{\mathbf{T}}_{g} + \mathbf{S}_{d\mathbf{w},\mathbf{i}}\overline{\mathbf{d}}_{w\mathbf{i}} + \mathbf{S}_{d\mathbf{z},\mathbf{i}}\overline{\mathbf{d}}_{\mathbf{z}} + \sum_{j\neq i}\mathbf{S}_{ij}\overline{\mathbf{T}}_{zj}\right) - \overline{\mathbf{T}}_{z\mathbf{i},ref}\right] (4.49)$$

Equation (4.49) can be written as

$$-2\overline{\mathbf{H}}_{\mathbf{d}c,i}\overline{\mathbf{v}}_{\mathbf{d}c,i}^* = \mathbf{K}_1^{\mathbf{d}c,i}\mathbf{T}_{\mathbf{w}i}(k) + \mathbf{K}_2^{\mathbf{d}c,i}\mathbf{T}_{\mathbf{z}i}(k) + \mathbf{K}_3^{\mathbf{d}c,i}\overline{\mathbf{d}}_{\mathbf{w}i} + \mathbf{K}_4^{\mathbf{d}c,i}\overline{\mathbf{d}}_{\mathbf{z}i} + \mathbf{K}_5^{\mathbf{d}c,i}\overline{\mathbf{T}}_{\mathbf{a}} + \mathbf{K}_6^{\mathbf{d}c,i}\overline{\mathbf{T}}_{\mathbf{g}} + \mathbf{K}_7^{\mathbf{d}c,i}\overline{\mathbf{T}}_{\mathbf{z}i,\mathrm{ref}} + \sum_{j\neq i} \mathbf{Q}_{ij}\overline{\mathbf{T}}_{\mathbf{z}j}$$
(4.50)

where,

$$\overline{\mathbf{H}}_{\mathbf{dc},\mathbf{i}} = \mathbf{H}_{\mathbf{dc},\mathbf{i}} \mathbf{P}_{\mathbf{vu},\mathbf{i}}^{-1},\tag{4.51}$$

$$\mathbf{K_{1}^{dc,i}} = 2\mathbf{S_{ui}^{T}} \mathbf{\bar{C}_{i}}^{T} \mathbf{Q_{1,i}} \mathbf{\bar{C}_{i}} \begin{bmatrix} \mathbf{I}_{N_{wi}} \\ \mathbf{0}_{N_{zi} \times N_{wi}} \end{bmatrix}, \qquad (4.52)$$

$$\mathbf{K_{2}^{dc,i}} = 2\mathbf{S_{ui}^{T}} \mathbf{\bar{C}_{i}}^{T} \mathbf{Q_{1,i}} \mathbf{\bar{C}_{i}} \begin{bmatrix} \mathbf{0}_{N_{wi} \times N_{zi}} \\ \mathbf{I}_{N_{zi}} \end{bmatrix}, \qquad (4.53)$$

$$\mathbf{K}_{3}^{\mathbf{dc},\mathbf{i}} = 2\mathbf{S}_{\mathbf{u}\mathbf{i}}^{T} \bar{\mathbf{C}}_{\mathbf{i}}^{T} \mathbf{Q}_{\mathbf{1},\mathbf{i}} \mathbf{S}_{\mathbf{dw},\mathbf{i}}, \qquad (4.54)$$

$$\mathbf{K}_{4}^{\mathbf{dc},\mathbf{i}} = 2\mathbf{S}_{\mathbf{u}\mathbf{i}}^{T} \bar{\mathbf{C}}_{\mathbf{i}}^{T} \mathbf{Q}_{\mathbf{1},\mathbf{i}} \mathbf{S}_{\mathbf{dz},\mathbf{i}}, \qquad (4.55)$$

$$\mathbf{K}_{5}^{\mathbf{dc},\mathbf{i}} = 2\mathbf{S}_{\mathbf{u}\mathbf{i}}^{T} \bar{\mathbf{C}}_{\mathbf{i}}^{T} \mathbf{Q}_{\mathbf{1},\mathbf{i}} \mathbf{S}_{\mathbf{a}\mathbf{i}}, \qquad (4.56)$$

$$\mathbf{K}_{6}^{\mathbf{dc},\mathbf{i}} = 2\mathbf{S}_{\mathbf{u}\mathbf{i}}^{T} \bar{\mathbf{C}}_{\mathbf{i}}^{T} \mathbf{Q}_{\mathbf{1},\mathbf{i}} \mathbf{S}_{\mathbf{g}\mathbf{i}}, \qquad (4.57)$$

$$\mathbf{K}_{7}^{\mathbf{dc},\mathbf{i}} = -2\mathbf{S}_{\mathbf{u}i}^{T} \bar{\mathbf{C}}_{\mathbf{i}}^{T} \mathbf{Q}_{\mathbf{1},\mathbf{i}}, \qquad (4.58)$$

$$\mathbf{Q}_{ij} = 2\mathbf{S}_{ui}^T \bar{\mathbf{C}}_i^T \mathbf{Q}_{1,i} \mathbf{S}_{ij}. \tag{4.59}$$

4.3.5 Comparison of centralized and decentralized MPC

We now present a quantitative comparison of the centralized and decentralized MPC control methodologies which forms the basis of the OLF metric presented in the next section. Let $\overline{\mathbf{\hat{T}}}_{\mathbf{z}}^{\mathbf{r}}$ be a vector of temperature estimates for zone *r* lifted in time, defined as

$$\overline{\mathbf{\hat{T}}}_{\mathbf{z}}^{\mathbf{r}} = \left(\mathbf{\widehat{T}}_{\mathbf{z}}^{\mathbf{r}}(k|k) \ \mathbf{\widehat{T}}_{\mathbf{z}}^{\mathbf{r}}(k+1|k) \dots \ \mathbf{\widehat{T}}_{\mathbf{z}}^{\mathbf{r}}(k+N-1|k)\right)^{T}.$$
(4.60)

Here, $\widehat{\mathbf{T}}_{\mathbf{z}}^{\mathbf{r}}(k+l|k)$ at any time instant k denotes the projected estimate of the temperature of zone r, l time steps ahead in future where $l \in \{0, 1, ..., N-1\}$. The overall lifted vector of estimates $\overline{\widehat{\mathbf{T}}}_{\mathbf{z}}$ is then defined as the sequential stack of $\overline{\widehat{\mathbf{T}}}_{\mathbf{z}}^{\mathbf{r}}$ given by (35)

$$\overline{\mathbf{\hat{T}}}_{\mathbf{z}} = \left[\left(\overline{\mathbf{\hat{T}}}_{\mathbf{z}}^{1} \right)^{T} \left(\overline{\mathbf{\hat{T}}}_{\mathbf{z}}^{2} \right)^{T} \dots \left(\overline{\mathbf{\hat{T}}}_{\mathbf{z}}^{N_{z}} \right)^{T} \right]^{T}$$
(4.61)

The vector $\overline{\mathbf{\hat{T}}}_{zj}$ appearing in (4.50) and defined in Section 3.4.4 can be generated by selecting appropriate components of $\overline{\mathbf{\hat{T}}}_{z}$. It can then be used to compute $\overline{\mathbf{v}}_{dc,i}^*$ using (4.50). Using the definition (4.46) of $\overline{\mathbf{v}}_{dc,i}^*$, the values $\{u_{dc,i}^{*r}(k+l|k)\}_{l=0}^{l=N-1}$ in (4.47) can be obtained for each zone *r* in the cluster S_{zi} . We observe that each pair (i, r) maps to a zone *m* in the building where

 $i \in \{1, 2, ..., p\}, r \in \{1, 2, ..., N_{zi}\}$ and $m \in \{1, 2, ..., N_z\}$. For each pair (i, r) we denote $u_{dc,i}^{*r}$ by u_{dc}^{*m} . In this way, repeating this procedure for each cluster S_{zi} , where $i \in \{1, 2, ..., p\}$, we obtain the set of values $\{u_{dc}^{*n}(k + l|k)\}_{l=0}^{l=N-1}$ for each zone $n \in \{1, 2, ..., N_z\}$ in the building. This is because the union of clusters S_{zi} decomposes the entire building (see definition 3.2)

For each $n \in \{1, 2, ..., N_z\}$, we represent the values $\{u_{dc}^{*n}(k + l|k)\}_{l=0}^{l=N-1}$ in a succint form by defining the vector

$$\bar{\mathbf{v}}_{dc}^{*n} = \begin{bmatrix} u_{dc}^{*n}(k|k) \\ u_{dc}^{*n}(k+1|k) \\ \vdots \\ u_{dc}^{*n}(k+N-1|k) \end{bmatrix}.$$
(4.62)

Next, an overall lifted vector of control inputs, \bar{v}_{dc}^* for the decentralized case is constructed by the sequential stack of \bar{v}_{dc}^{*n} given by

$$\bar{\mathbf{v}}_{\mathbf{dc}}^* = \left[(\bar{\mathbf{v}}_{\mathbf{dc}}^{*1})^T \ (\bar{\mathbf{v}}_{\mathbf{dc}}^{*2})^T \ \dots \ (\bar{\mathbf{v}}_{\mathbf{dc}}^{*N_z})^T \right]^T$$
(4.63)

Note that $\bar{\mathbf{v}}_{dc}^*$ constructed above and $\bar{\mathbf{v}}_c^*$ defined in (4.23) are comparable vectors in the sense that entries in identical locations in these vectors are associated with the same zone in the building and the same time instant in the prediction horizon. This fact is easily verifiable.

Theorem 4.1 (Centralized-Decentralized Equivalence) Let $\overline{\mathbf{\hat{T}}}_{\mathbf{z}}^* \in \mathbb{R}^{N.N_z}$ be a solution of the linear equation (4.64) in y. If the overall lifted vector of zone temperature estimates $\overline{\mathbf{\hat{T}}}_{\mathbf{z}}$ (defined in (4.61)) is set to $\overline{\mathbf{\hat{T}}}_{\mathbf{z}}^*$, then the ensuing overall lifted vector of control inputs, $\overline{\mathbf{v}}_{\mathbf{dc}}^*$ (defined in (4.63)) computed by the decentralized multi-agent MPC control architecture satisfies $\overline{\mathbf{v}}_{\mathbf{dc}}^* = \overline{\mathbf{v}}_{\mathbf{c}}^*$.

$$(\mathbf{Q}_{dc}\overline{\mathbf{P}})\mathbf{y} = \mathbf{K}_{1}\mathbf{T}_{w}(k) + \mathbf{K}_{2}\mathbf{T}_{z}(k) + \mathbf{K}_{3}\overline{\mathbf{d}}_{w} + \mathbf{K}_{4}\overline{\mathbf{d}}_{z} + \mathbf{K}_{5}\overline{\mathbf{T}}_{a} + \mathbf{K}_{6}\overline{\mathbf{T}}_{6} + \mathbf{K}_{7}\overline{\mathbf{T}}_{z,ref}$$
(4.64)

where, $\mathbf{K}_{1} = \overline{\mathbf{H}}_{dc} \overline{\mathbf{P}} \overline{\mathbf{H}}_{c}^{-1} \mathbf{K}_{1}^{c} - \mathbf{K}_{1}^{dc} \mathbf{R}, \qquad (4.65)$

$$\mathbf{K}_2 = \overline{\mathbf{H}}_{dc} \overline{\mathbf{P}} \overline{\mathbf{H}}_c^{-1} \mathbf{K}_2^c - \mathbf{K}_2^{dc} \mathbf{P}, \qquad (4.66)$$

$$\mathbf{K}_3 = \overline{\mathbf{H}}_{dc} \overline{\mathbf{P}} \mathbf{H}_c^{-1} \mathbf{K}_3^c - \mathbf{K}_3^{dc} \overline{\mathbf{R}}, \qquad (4.67)$$

$$\mathbf{K}_4 = \overline{\mathbf{H}}_{dc} \overline{\mathbf{P}} \mathbf{H}_c^{-1} \mathbf{K}_4^c - \mathbf{K}_4^{dc} \overline{\mathbf{P}}, \qquad (4.68)$$

$$\mathbf{K}_5 = \mathbf{H}_{\mathbf{dc}} \overline{\mathbf{P}} \mathbf{H}_{\mathbf{c}}^{-1} \mathbf{K}_5^{\mathbf{c}} - \mathbf{K}_5^{\mathbf{dc}}, \qquad (4.69)$$

$$\mathbf{K}_6 = \mathbf{H}_{dc} \overline{\mathbf{P}} \mathbf{H}_c^{-1} \mathbf{K}_6^c - \mathbf{K}_6^{dc}, \qquad (4.70)$$

$$\mathbf{K}_7 = \mathbf{H}_{\mathbf{dc}} \overline{\mathbf{P}} \mathbf{H}_{\mathbf{c}}^{-1} \mathbf{K}_7^{\mathbf{c}} - \mathbf{K}_7^{\mathbf{dc}} \overline{\mathbf{P}}, \qquad (4.71)$$

$$\mathbf{Q}_{dc} = \begin{pmatrix} \mathbf{0} & \mathbf{Q}_{1,2} & \cdots & \mathbf{Q}_{1,p} \\ \mathbf{Q}_{2,1} & \mathbf{0} & \cdots & \mathbf{Q}_{2,p} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{Q}_{p,1} & \mathbf{Q}_{p,2} & \cdots & \mathbf{0} \end{pmatrix}.$$
(4.72)

In the above equations,

$$\overline{\mathbf{H}}_{dc} = \begin{pmatrix} \overline{\mathbf{H}}_{dc,1} & & & \\ & \overline{\mathbf{H}}_{dc,2} & & \\ & & \ddots & \\ & & & \overline{\mathbf{H}}_{dc,p} \end{pmatrix}, \qquad (4.73)$$

$$\mathbf{K}_{m}^{dc} = \begin{pmatrix} \mathbf{K}_{m}^{dc,1} & & & \\ & \mathbf{K}_{m}^{dc,2} & & \\ & & \ddots & \\ & & & \mathbf{K}_{m}^{dc,p} \end{pmatrix} \text{ for } m \in \{1,2,3,4,7\}, \qquad (4.74)$$

$$\mathbf{K}_{m}^{\mathbf{dc}} = \left[\left(\mathbf{K}_{m}^{\mathbf{dc},\mathbf{1}} \right)^{T} \quad \left(\mathbf{K}_{m}^{\mathbf{dc},\mathbf{2}} \right)^{T} \dots \left(\mathbf{K}_{m}^{\mathbf{dc},\mathbf{p}} \right)^{T} \right]^{T} \text{ for } m \in \{5,6\}. \quad (4.75)$$

Proof: Using properties 1, 2, 6 and 7 of the extraction matrices (Section 4.3.3), (4.50) can equivalently be restated in the form shown below

$$-2\overline{\mathbf{H}}_{\mathbf{d}c,i}\overline{\mathbf{P}}_{i}\overline{\mathbf{v}}_{\mathbf{d}c}^{*} = \mathbf{K}_{1}^{\mathbf{d}c,i}\mathbf{R}_{i}\mathbf{T}_{w}(k) + \mathbf{K}_{2}^{\mathbf{d}c,i}\mathbf{P}_{i}\mathbf{T}_{z}(k) + \mathbf{K}_{3}^{\mathbf{d}c,i}\overline{\mathbf{R}}_{i}\overline{\mathbf{d}}_{w} + \mathbf{K}_{4}^{\mathbf{d}c,i}\mathbf{P}_{i}\overline{\mathbf{d}}_{z} + \mathbf{K}_{5}^{\mathbf{d}c,i}\overline{\mathbf{T}}_{a} + \mathbf{K}_{6}^{\mathbf{d}c,i}\overline{\mathbf{T}}_{g} + \mathbf{K}_{7}^{\mathbf{d}c,i}\overline{\mathbf{P}}_{i}\overline{\mathbf{T}}_{z,ref} + \sum_{j\neq i}\mathbf{Q}_{ij}\overline{\mathbf{P}}_{j}\overline{\mathbf{T}}_{z}$$
(4.76)

Using the definitions of **P**, $\overline{\mathbf{P}}$, **R** and $\overline{\mathbf{R}}$ from (4.35), (4.36), (4.38) and (4.39) respectively, the combined form of (4.76) resulting from the concatenation over all clusters (i = 1, 2, ..., p), is expressed as (4.76), where \mathbf{Q}_{dc} , \mathbf{H}_{dc} and \mathbf{K}_m^{dc} ($m \in \{1, 2, ..., 7\}$) are as defined in (4.73) to (4.75).

$$-2\overline{H}_{dc}\overline{P}\overline{v}_{dc}^* = K_1^{dc}RT_w(k) + K_2^{dc}PT_z(k) + K_3^{dc}\overline{R}\overline{d}_w + K_4^{dc}P\overline{d}_z + K_5^{dc}\overline{T}_a + K_6^{dc}\overline{T}_g + K_7^{dc}\overline{P}\overline{T}_{z,ref}$$

$$+ \mathbf{Q}_{\mathbf{dc}} \overline{\mathbf{P}} \widehat{\mathbf{T}}_{\mathbf{z}} \tag{4.76}$$

Comparing (4.25) and (4.76), $\bar{\mathbf{v}}_{dc}^* = \bar{\mathbf{v}}_c^*$ if $\overline{\mathbf{\hat{T}}}_z$ is such that $(\mathbf{Q}_{dc}\mathbf{\bar{P}})\mathbf{\hat{T}}_z = \mathbf{K}_1\mathbf{T}_w(k) + \mathbf{K}_2\mathbf{T}_z(k) + \mathbf{K}_3\mathbf{d}_w + \mathbf{K}_4\mathbf{d}_z + \mathbf{K}_5\mathbf{\bar{T}}_a + \mathbf{K}_6\mathbf{\bar{T}}_6 + \mathbf{K}_7\mathbf{\bar{T}}_{z,ref}$ with $\mathbf{K}_m, m \in \{1, 2, ..., 7\}$ defined in (4.65) to (4.71). This completes the proof of Theorem 4.1.

Remarks. It should be noted that the centralized-decentralized equivalence condition stated above was derived based on the assumption that, apart from the underlying dynamical models, no other constraints are imposed on the centralized and decentralized optimization problems.

Corollary 4.1 If the overall lifted vector of zone temperature estimates, $\overline{\mathbf{\hat{T}}}_{\mathbf{z}}$ is chosen to be different from $\overline{\mathbf{\hat{T}}}_{\mathbf{z}}^*$, the overall lifted vector of optimal control inputs for the corresponding decentralized controller differs from that for centralized control by an amount that is linearly dependent on the deviation of $\overline{\mathbf{\hat{T}}}_{\mathbf{z}}$ from $\overline{\mathbf{\hat{T}}}_{\mathbf{z}}^*$. More precisely,

$$\bar{\mathbf{v}}_{\mathbf{c}}^* - \mathbf{v}_{\mathbf{dc}}^* = -\frac{1}{2} \bar{\mathbf{P}}^{-1} \bar{\mathbf{H}}_{\mathbf{dc}}^{-1} \mathbf{Q}_{\mathbf{dc}} \bar{\mathbf{P}} \left(\bar{\mathbf{T}}_{\mathbf{z}}^* - \bar{\mathbf{T}}_{\mathbf{z}} \right)$$
(4.77)

Proof: The application of Theorem 4.1 to (4.76) provides the following alternative expression for $\bar{\mathbf{v}}_{\mathbf{c}}^*$, which was originally given by (4.25):

$$-2\overline{H}_{dc}\overline{P}\overline{v}_{c}^{*} = K_{1}^{dc}RT_{w}(k) + K_{2}^{dc}PT_{z}(k) + K_{3}^{dc}\overline{R}\overline{d}_{w} + K_{4}^{dc}P\overline{d}_{z} + K_{5}^{dc}\overline{T}_{a} + K_{6}^{dc}\overline{T}_{g} + K_{7}^{dc}\overline{P}\overline{T}_{z,ref}$$
$$+ Q_{dc}\overline{P}\overline{T}_{z}^{*}. \qquad (4.78)$$

Subtraction of (4.76) from (4.78) leads to (4.77).

Remarks.

1. The family of solutions to (4.64) can be described by the set $\{\mathbf{y}_0 + \overline{\mathbf{P}}^{-1}\mathbf{x}: \mathbf{x} \in Ker(\mathbf{Q}_{dc})\}$, where \mathbf{x} is any particular solution of (4.64). To satisfy the conditions of Theorem 4.1, $\overline{\mathbf{T}}_z^*$ can be chosen as any element from this set. In the particular case where \mathbf{Q}_{dc} is invertible, $\overline{\mathbf{T}}_z^*$ has a unique closed form expression given by $\overline{\mathbf{T}}_z^* = \overline{\mathbf{P}}^{-1}\mathbf{Q}_{dc}^{-1}(\mathbf{K}_1\mathbf{T}_w(k) + \mathbf{K}_2\mathbf{T}_z(k) + \mathbf{K}_3\mathbf{d}_w + \mathbf{K}_4\mathbf{d}_z + \mathbf{K}_5\mathbf{T}_a + \mathbf{K}_6\mathbf{T}_6 + \mathbf{K}_7\mathbf{T}_{z,ref}).$

2. Since $\overline{\mathbf{H}}_{\mathbf{dc},\mathbf{i}}$, $i \in \{1,2,\ldots,p\}$ is full rank due to strict convexity of the decentralized MPC optimization problem, the matrix $\overline{\mathbf{H}}_{\mathbf{dc}}$ defined in (4.73) is also full rank. Hence, $\mathbf{H}_{\mathbf{dc}}^{-1}$ exists

for use in (4.77).

3. The centralized-decentralized equivalence (Theorem 4.1) provides a sufficient condition for which the solutions of the centralized and decentralized (corresponding to any p-partition) MPC problems match. The temperature estimates \overline{T}_{z}^{*} , that the multi-agent decentralized controllers would require in such a situation, depend on system wide sensory data $T_{z}(k)$ and $T_{w}(k)$, as expressed by (4.64). Since each decentralized controller has access to only certain temperature measurements $T_{zi}(k)$ and $T_{wi}(k)$, the centralized-decentralized equivalence of Theorem 4.1, cannot be achieved in practice. However, this condition can be used to quantify the difference between the centralized and decentralized solutions as stated in Corollary 4.1.

4.3.6 Optimality metric

For any p-partition of the building, where $p \in \{1, 2, ..., N_z\}$, we define an appropriate scalar metric to quantify the optimality associated with a multi-agent MPC controller that is decentralized with respect to the clusters constituting the partition. The overall lifted vector of temperature estimates \overline{T}_z defined in (4.61) which is required for implementing the decentralized MPC controller is, at best, chosen heuristically. The centralized-decentralized equivalence (Theorem 4.1) establishes a theoretical best value of this estimate, \overline{T}_z^* , which if used, results in matching of the centralized and decentralized control inputs. However, an arbitrary choice of this estimate results in a deviation of the decentralized control inputs \mathbf{v}_{dc}^* from the centralized control inputs $\overline{\mathbf{v}}_c^*$ which is quantified by (4.77). This deviation translates into a deviation of the centralized objective function from its optimal value. To quantify it we proceed as follows.

Using the transformation 4.4, the centralized objective function $g_c(\bar{\mathbf{u}})$ given by (3.19) was expressed as a function $g'_c(\bar{\mathbf{v}})$ shown in (4.5). An alternative expression for $g'_c(\bar{\mathbf{v}})$ is

$$g_c'(\bar{\mathbf{v}}) = g_c'(\bar{\mathbf{v}}_c^*) + (\bar{\mathbf{v}} - \bar{\mathbf{v}}_c^*)^T \overline{\mathbf{H}}(\bar{\mathbf{v}} - \bar{\mathbf{v}}_c^*).$$
(4.79)

Using Corollary 4.1, we obtain

$$g_{c}'(\bar{\mathbf{v}}_{c}^{*}) - g_{c}'(\bar{\mathbf{v}}_{dc}^{*}) = (\bar{\mathbf{v}}_{dc}^{*} - \bar{\mathbf{v}}_{c}^{*})^{T} \overline{\mathbf{H}}(\bar{\mathbf{v}}_{dc}^{*} - \bar{\mathbf{v}}_{c}^{*})$$
$$= \frac{1}{4} \left(\overline{\mathbf{T}}_{z} - \overline{\mathbf{T}}_{z}^{*}\right)^{T} \overline{\mathbf{P}}^{T} \mathbf{Q}_{dc}^{T} \overline{\mathbf{H}}_{dc}^{-1} \overline{\mathbf{P}} \overline{\mathbf{H}} \overline{\mathbf{P}}^{T} \overline{\mathbf{H}}_{dc}^{-1} \mathbf{Q}_{dc} \overline{\mathbf{P}}\left(\overline{\mathbf{T}}_{z} - \overline{\mathbf{T}}_{z}^{*}\right).$$
(4.80)

Here, we have used the fact that \overline{H}_{dc} is a symmetric matrix and \overline{P} is a permutation matrix.

It is desired that the above deviation be small, so that the decentralized controller can provide performance which is close to that provided by centralized control. An illustration of this deviation is shown in Figure 4.13 for a hypothetical scalar case.

Using the inequality shown in (4.81), it can be concluded that the quantity $\lambda_{max}(\overline{\mathbf{P}}^T \mathbf{Q}_{dc}^T \overline{\mathbf{H}}_{dc}^{-1} \overline{\mathbf{P}} \overline{\mathbf{H}} \overline{\mathbf{P}}^T \overline{\mathbf{H}}_{dc}^{-1} \mathbf{Q}_{dc} \overline{\mathbf{P}})$ characterizes an upper bound on the above deviation in the objective function which is independent of the deviation of $\overline{\mathbf{T}}_z$ from $\overline{\mathbf{T}}_z^*$. Therefore, it represents the 'loss' in optimality in going from a centralized control architecture to a decentralized control architecture. We denote this quantity using the term **Optimality Loss Factor** (OLF) as shown in (4.82).

$$\left(\overline{\widehat{\mathbf{T}}}_{\mathbf{z}} - \overline{\widehat{\mathbf{T}}}_{\mathbf{z}}^{*} \right)^{T} \overline{\mathbf{P}}^{T} \mathbf{Q}_{\mathbf{dc}}^{T} \overline{\mathbf{H}}_{\mathbf{dc}}^{-1} \overline{\mathbf{P}} \overline{\mathbf{H}} \overline{\mathbf{P}}^{T} \overline{\mathbf{H}}_{\mathbf{dc}}^{-1} \mathbf{Q}_{\mathbf{dc}} \overline{\mathbf{P}} \left(\overline{\widehat{\mathbf{T}}}_{\mathbf{z}} - \overline{\widehat{\mathbf{T}}}_{\mathbf{z}}^{*} \right) \leq \lambda_{max} \left(\overline{\mathbf{P}}^{T} \mathbf{Q}_{\mathbf{dc}}^{T} \overline{\mathbf{H}}_{\mathbf{dc}}^{-1} \overline{\mathbf{P}} \overline{\mathbf{H}} \overline{\mathbf{P}}^{T} \overline{\mathbf{H}}_{\mathbf{dc}}^{-1} \mathbf{Q}_{\mathbf{dc}} \overline{\mathbf{P}} \right) \left\| \overline{\widehat{\mathbf{T}}}_{\mathbf{z}} - \overline{\widehat{\mathbf{T}}}_{\mathbf{z}}^{*} \right\|_{2}^{2}.$$
(4.81)

$$OLF = \lambda_{max}(\overline{\mathbf{P}}^T \mathbf{Q}_{dc}^T \overline{\mathbf{H}}_{dc}^{-1} \overline{\mathbf{P}} \overline{\mathbf{H}} \overline{\mathbf{P}}^T \overline{\mathbf{H}}_{dc}^{-1} \mathbf{Q}_{dc} \overline{\mathbf{P}}) = \sigma_{max}^2(\overline{\mathbf{H}}^{-1} \overline{\mathbf{P}}^T \overline{\mathbf{H}}_{dc}^{-1} \mathbf{Q}_{dc} \overline{\mathbf{P}}).$$
(4.82)

The OLF is used as the appropriate optimality metric which must be minimized in the choice of partitions for decentralized control.



Figure 4.13 Scalar illustration of deviation in performance between centralized and decentralized MPC when $\overline{\widehat{T}}_z^* \neq \overline{\widehat{T}}_z$

4.3.7 Robustness metric

For any *p*-partition of the building, where $p \in \{1, 2, ..., N_z\}$, we define an appropriate scalar metric to quantify the robustness associated with a multi-agent MPC controller that is decentralized with respect to the clusters constituting the partition. In such an architecture, the effect of a sensor or communication related fault on the resulting control inputs, $\bar{\mathbf{v}}_{dc}^*$, will be confined to the cluster where the fault originates because the control agents do not communicate. Therefore, the average number of affected zones in the event of a failure is an indicator of robustness in the sense that a small value ensures that the effect of failures is less widespread.

To quantify this concept, we consider the event that a failure has occurred in one of the zones in the building. For simplicity, we assume that the probability of failure occurring in any particular zone of the building is uniform across all zones and therefore equals $1/N_z$. We define the **Fault Propagation Metric** (FPM) $\in (0,1]$ as the expected value of the aggregated thermal capacity of all affected zones in case of above failure event, normalized with respect to the total thermal capacity of zones in the building. An expression for the FPM is derived as shown

$$FPM = \frac{Expected value of aggregated thermal capacity of all affected zones}{Total thermal capacity of all zones in the building}
$$\sum_{m=1}^{N_z} (Probability of failure in zone m \times \\ = \frac{Net thermal capacity of zones in cluster containing zone m)}{Total thermal capacity of all zones in the building} \\ \frac{\sum_{i=1}^{p} (Probability of failure in each zone \times Number of zones in cluster i \times \\ = \frac{Net thermal capacity of zones in cluster i)}{Total thermal capacity of all zones in the building}$$$$

$$=\frac{\sum_{i=1}^{p}\frac{1}{N_{z}}N_{zi}C_{zi}}{N_{z}}=\frac{1}{N_{z}C_{z}}\sum_{i=1}^{p}N_{zi}C_{zi}$$
(4.83)

As an example, FPM for all possible partitions of the 3-zone building shown in Figure 3.3 are calculated in Table 4.5. For simplicity, the thermal capacity of each zone is set to unity.

p	p-partitions	FPM
1	{1,2,3}	1
2	{1,2},{3} OR {1,3},{2} OR {3,2},{1}	$\frac{(2\times2)+(1\times1)}{(3\times3)} = \frac{5}{9}$
3	{1},{2},{3}	$\frac{(1\times1)+(1\times1)+(1\times1)}{3\times3} = \frac{3}{9}$

 Table 4.5 FPM computation for 3-zone building in Figure 3.3

It should be noted that in the case of a building, which exhibits heterogeneity in the sense that its thermal zones are sized differently, the total volume of space affected is a more appropriate characterization of the effect of the failure event than just the number of affected thermal zones. Since the thermal capacity of a zone is closely related to its volume, the FPM defined above is an indicator of the effect of failure in terms of the volume of affected space. Therefore, the FPM is used as an appropriate robustness metric which must be minimized in the choice of partitions for decentralized control.

4.3.8 Optimal partitioning problem and complexity analysis

This section presents a formal definition of the partitioning problem for the decentralized thermal control of a building and analyzes the underlying computational complexity.

Definition 4.9 (Optimal *p*-*partition*): An *optimal p*-*partition* is defined as one with the smallest OLF among all possible *p*-*partitions* of S_z , for a fixed $p \in \{1, 2, ..., N_z\}$.

Definition 4.10. (Optimal partitioning problem): The *optimal partitioning problem* is to determine a family of *optimal p-partitions*, one for each $p \in \{1, 2, ..., N_z\}$.

For a given p, the number of *p*-partitions of the set S_z is given by the Stirling number of the second kind, $S(N_z, p)$ [84], as expressed in (4.84). Therefore, the total number of partitions to be considered to solve the optimal partitioning problem is given by the sum of the Stirling numbers over p, which is also defined as the Bell number [84], denoted by B_{N_z} and is expressed in (4.85).

$$S(N_z, p) = \frac{1}{N_z!} \sum_{j=1}^p (-1)^j {p \choose j} (p-j)^{N_z}$$
(4.84)

$$B_{N_z} = \sum_{p=1}^{N_z} S(N_z, p)$$
 (4.85)

The Bell number grows exponentially⁴ with N_z . This implies that if all possible partitions were to be considered, the optimal partitioning problem becomes intractable as the number of zones in the building increases. This motivates the development of a less computationally complex methodology for optimal partitioning using only a small subset of all partitions of S_z .

4.3.9 Agglomerative clustering

Agglomerative or 'bottom-up' clustering [86] is a hierarchical methodology used in a variety of applications such as data-mining and bio-informatics [87, 88] to form clusters of objects. It starts with individual objects that are progressively grouped together into larger clusters until the root cluster containing all the objects is reached. This is typically done using a greedy approach which groups the two 'closest' clusters together at each step, based on a suitable distance function metric between clusters.

We adopt the agglomerative clustering approach using the OLF as a distance function, to address the optimal partitioning problem in the context of decentralized building thermal control.

Algorithm 4.1. Agglomerative Clustering for Partitioning a Building (See Figure 4.14 for illustration)

STEP 1: Define the initial *parent partition* as the unique N_z -partition of S_z , which consists of N_z clusters, each having exactly one zone.

STEP 2: Agglomerate any two clusters in the parent partition to create a *child partition*. In this way, find all child partitions of the parent partition. Compute the OLF for each such child partition.

STEP 3: Among all child partitions found above, determine one with smallest OLF. Set the new parent partition to be this child partition. In case of multiple child partitions having the smallest OLF, select any one of them.

STEP 4: Repeat steps 2 and 3, until the parent partition becomes the unique 1-partition of Sz,

⁴The first few Bell numbers are given by 1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975,...

which consists of exactly one cluster that contains all zones.



Figure 4.14 Illustration of agglomerative clustering for a building with 5 zones

Remarks.

1. The *parent partition* has N_z clusters initially. At the end of each iteration, the number of clusters decreases exactly by 1. Therefore the *parent partition* obtained at the end of the *i*th iteration is a *p*-partition, with $p = N_z - i + 1$.

2. The *p*-partitions generated via these iterations are not necessarily optimal *p*-partitions (definition 4.9). They represent a guess for an optimal *p*-partition, obtained from the previous iteration in a greedy manner.

3. If the size of the parent partition in any iteration of Algorithm 4.1 is N^{parent} , the number of child partitions N^{child} that are created using the agglomeration described in Step 2 of the algorithm is given by:

$$N^{child} = \binom{N^{parent}}{2} \tag{4.86}$$

Noting that N^{parent} starts from N_z and decreases by 1 in each iteration, the total number of child partitions, N_{total}^{child} considered in one run of the algorithm is $O(N_z^3)$ as computed below.

$$N_{total}^{child} = \sum_{1}^{N_z} {N^{parent} \choose 2} = \sum_{1}^{N_z} \frac{N^{parent} (N^{parent} - 1)}{2} = \frac{N_z (N_z + 1)(N_z - 1)}{6} \quad (4.87)$$

Hence a significant computational benefit is achieved when compared to the complexity associated with considering all possible partitions (Figure 4.15).



Figure 4.15 Computational complexity comparison of partitioning approaches

Similar to the CLF-MCS method, the results of Algorithm 4.1 can be presented on an *optimality-robustness trade-off curve*, as notionally illustrated in Figure 4.16, where the OLF and the FPM values of the resulting parent partitions from each iteration are plotted. This represents a multi-objective optimization framework, where the goal is to simultaneously minimize both the OLF and the FPM. The rightmost and leftmost points on this curve correspond to the two extremes of a completely centralized architecture (a single cluster) and a completely decentralized architecture (N_z clusters) respectively. This curve serves as a useful design tool. It can be used to compare various partitions and make a decision on the appropriate intermediate architecture between these two extremes that results in a satisfactory trade-off between optimality and robustness objectives. Some heuristic guidelines, similar to the CLF-MCS approach, are presented below based on visual inspection.



Figure 4.16 Illustration of optimality-robustness trade-off curve for OLF-FPM approach

Starting from the rightmost point, and proceeding left on the optimality-robustness tradeoff curve, the partitions which correspond to a 'knee point' on this curve should be explored. At such points, navigation in either direction would result in a large increase in one metric but only a relatively small decrease in the other metric. Therefore, these points reflect the attainment of a satisfactory balance between the optimality and robustness objectives, and the corresponding partition of the building should preferably be used for decentralized control. In the event of multiple knee points in the curve, the sharpest among them may be considered. Also, those knee points which are more centrally located should be preferred over others. The following sections provide some examples to further explain this process.

4.3.10 Nine zone building example

We revisit the 9-zone building described in section 4.2.9. Its layout is shown in Figure 4.9, and the construction properties, wall resistances and capacitances are shown in Table 4.1 and Table 4.2. Similar to Section 4.2.9, the capacitance for each zone was assumed to be 250 kJ/kg. The contribution of occupants and objects to the zone capacitances was ignored for simplicity. With the above modeling assumptions, the building is rendered thermally symmetric, meaning that the resistances offered by all internal walls are the same (Table 4.2). To introduce anisotropy, we artificially decrease the thermal resistances associated with the vertical internal walls by a factor of 3 from their originally computed values. Furthermore, we multiply each wall resistance by a factor of 0.06 to increase the coupling among zones. An overall system model of

the form (3.4) was obtained by constructing an RC network using these details, and then applying Algorithm 3.1 where discretization was performed using the zero-order-hold method with step size of 10 minutes.

To partition the building, the exact solution to the optimal partitioning problem was obtained for this example by considering all possible p-partitions, for each $p \in \{1, 2, ..., 9\}$ as described in Section 4.3.8. Next, the agglomerative clustering approach (Algorithm 4.1) was applied, which provided suitable guesses for optimal p-partitions. OLF computations were based on the MPC parameters N = 5 samples, $\alpha = 0.1\mathbf{e}_9$ and $\beta = \mathbf{e}_9$. All relevant codes are provided in Appendix C. The resulting partitions from the agglomerative approach and the exact solution approach are compared in Figure 4.17 and Table 4.6. The corresponding optimality-robustness trade-off curves are shown in Figure 4.18. The following observations can be made from these results:

1. As seen in Figure 4.17, the p-partitions obtained using the agglomerative approach have OLF values close (in many cases identical) to optimal *p*-partitions, for all $p \in \{1, 2, ..., 9\}$.

2. Visual inspection of Figure 4.18 for knee-points suggests that the partition which offers the 'best' trade-off between the OLF (optimality) and FPM (robustness) objectives is an optimal *3-partition*, which from the Table 4.6 corresponds to $\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\}$. This would partition the building along its floors and is physically consistent with the thermal anisotropy that was introduced by causing horizontal walls (floors and ceilings) to be more insulated than vertical walls.

3. Another knee-point is observed in Figure 4.18 which corresponds to p = 6. This can possibly be explained on the basis of the partitions obtained in Table 3 for the agglomerative approach. Navigation from p = 7 to p = 6 results in fusion of the zones 8 and 9 from separate clusters into a single cluster. However, these zones have a potentially significant dynamical coupling through the relatively less insulated common vertical wall between them. This is likely to cause a large decrease in OLF for a comparatively small increase in FPM while going from p = 7 to p = 6 resulting in the knee point seen in Figure 4.18.

4. Finding the exact solution to the optimal partitioning problem required the consideration of 4140 partitions, whereas only 240 partitions were analyzed by the agglomerative approach.

From the above observations, it can be concluded that agglomeration is able to provide sufficiently accurate results with significantly less computational effort when compared to the exact solution approach involving the analysis of all partitions.

	Optimal p-partitio	n	p-partition from agglo	meration
p	Clusters	OLF (× 10 ⁻³)	Clusters	OLF (× 10 ⁻³)
9	$\{1\} \{2\} \{3\} \{4\} \{5\} \{6\} \{7\} \{8\} \{9\}$	4.273	$\{1\} \{2\} \{3\} \{4\} \{5\} \{6\} \{7\} \{8\} \{9\}$	4.273
8	$\{1\}\{2\}\{3\}\{4,5\}\{6\}\{7\}\{8\}\{9\}$	4.053	$\{1\}\{2\}\{3\}\{4,5\}\{6\}\{7\}\{8\}\{9\}$	4.053
7	$\{1,2\}$ $\{3\}$ $\{4\}$ $\{5\}$ $\{6\}$ $\{7\}$ $\{8,9\}$	3.873	$\{1\}\{2,3\}\{4,5\}\{6\}\{7\}\{8\}\{9\}$	3.977
6	$\{1\}\{2,3\}\{4,5\}\{6\}\{7\}\{8,9\}$	2.088	$\{1\}\{2,3\}\{4,5\}\{6\}\{7\}\{8,9\}$	2.088
5	$\{1\}\{2,3\}\{4,5,6\}\{7,8\}\{9\}$	2.006	$\{1\}\{2,3\}\{4,5\}\{6\}\{7,8,9\}$	2.023
4	$\{1,2,3\}$ $\{4\}$ $\{5,6\}$ $\{7,8,9\}$	1.936	$\{1,2,3\}$ $\{4,5\}$ $\{6\}$ $\{7,8,9\}$	1.936
3	{1,2,3} {4,5,6} {7,8,9}	5.118	{1,2,3} {4,5,6} {7,8,9}	5.118
2	{1,2,3,4,5,6} {7,8,9}	2.559	{1,2,3,4,5,6} {7,8,9}	2.559
1	{1,2,3,4,5,6,7,8,9}	0	{1,2,3,4,5,6,7,8,9}	0

Table 4.6 Optimal partitions vs. partitions using agglomeration for 9-zone building



Figure 4.17 OLF comparison of all p-partitions (solid circles) of the 9-zone building with agglomerative partitions (empty circles). The agglomerative clustering progresses from left to right, starting with the most decentralized partition (p = 9).



Figure 4.18 Optimality robustness trade-off *curves* for 9-zone building using true optimal partitions and agglomerative partitions

4.3.11 Eleven zone circular building

We now consider a single-story circular building with 11 zones including a central atrium (zone 1), as shown in Figure 4.19, which can be thought of as a small office building. It is surrounded by ambient on all sides. Hallways, shown shaded, are provided to facilitate the movement of people inside the building. For the purposes of modeling, the building has 27 external walls and an equal number of internal walls. The walls are assumed to have construction properties as shown in Table 4.1. The R and C parameters shown in Table 4.7 and Table 4.8 were computed in a manner similar to the 9-zone building, with the hallways modeled as resistors with a high value of resistance calculated using the thermal conduction and convection properties of air. For simplicity, the accessibility factors [38] for computation of wall resistances were assumed to be 0.5 each. Similar to the 9-zone building in the previous section, an overall system model of the form (3.4) was obtained by constructing an RC network using these details, and then applying Algorithm 3.1 where discretization was performed using the zero-order-hold method with step size of 10 minutes.



Figure 4.19 Layout of 11-zone circular building (plan view with building height = 15 ft)

Table 4.7 Resistance and capacitance values for walls of 11-zone building (see Figure 3.5)

Type of wall	C _{wall} (kJ/kg)	R ₁ (K/kW)	R ₂ (K/kW)
Ceiling and floor for zone 1	4.27×10^4	9.178	9.178
Ceilings and floors for zones 2-6	2.56×10^4	15.297	15.297
Ceilings and floors for zones 7-11	3.36×10^4	8.344	8.344
Vertical walls separating ambient from each of zones $7-11$	4.17×10^4	9.414	9.414
Vertical walls between zone 1 and each of zones $2-6$	6.53×10^3	9.160	9.160
Vertical walls separating zone pairs (2,3), (3,4), (4,5) and (5,6)	5.20×10^3	11.515	11.515
Vertical wall (hallway separation) between zones 2 and 6	1.04×10^4	1151.50	1151.50
Vertical walls (hallway) separating zone pairs (2,7), (3,8), (4,9), (5,10), (6,11)	2.77×10^4	431.25	431.25
Vertical walls separating zone pairs (7,8), (8,9), (9,10) and (10,11)	5.20×10^3	11.515	11.517
Vertical wall (hallway separation) between zones 7 and 11	1.04×10^4	1151.50	1151.50

Zone number	Capacitance (kJ/kg)
1	643.72
2-6	386.23
7 – 11	708.09

 Table 4.8 Zone capacitances for 11-zone building

An intuitive method to partition the building is to split it along the thermally insulating circular ring of hallway, resulting in two clusters: {1,2,3,4,5,6} and {7,8,9,10,11}. However, it is not obvious how to further partition these clusters into smaller clusters. The partitions resulting from the application of the agglomerative clustering algorithm are shown in Table 4.9, where parameters used to compute the OLF values are N = 5, $\alpha = 0.1e_{11}$ and $\beta = e_{11}$. Relevant codes are provided in Appendix D. The corresponding optimality-robustness trade-off curve is shown in Figure 4.20. A visual inspection of this curve indicates the presence of two knee points, as labeled in the figure. Following observations are made:

1. Knee 1 corresponds to the intuitive partition {{1,2,3,4,5,6}, {7,8,9,10,11}} which was noted above. However, the unacceptably high FPM associated with this partition indicates inferior robustness to faults, motivating further partitioning of these clusters.

2. The OLF values for both knees 1 and 2 are close to zero. However, the FPM associated with knee 2 is only about 60% of the FPM for knee 1. Therefore, knee 2 provides a better trade-off between optimality and robustness than knee 1.

3. Strong dynamic coupling is expected to exist among zones 1 to 6, primarily due to the atrium (zone 1) which is connected to each of the zones 2 to 6. Therefore to ensure small deviation from optimality, the building should be partitioned such that these zones are contained in the same cluster. This is verified from the clusters constituting the partition for knee 2 (p = 4 in Table 4.9).

The above observations can also be explained by considering a scalarized framework, which is a widely used approach for multi-objective optimization [82]. For the multi-objective problem of minimizing OLF and FPM, we define a single scalar objective function $J_{partition}$ as shown in (4.88), where λ is a parameter $\in [0,1]$ which can be adjusted to influence the relative weights on the optimality and robustness objectives. Here OLF _{max} is the value of OLF for the most decentralized partition $\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\},\{10\},\{11\}\}\}$, whereas FPM _{max} is the value of the FPM for the most centralized partition, i.e. $\{1,2,3,4,5,6,7,8,9,10,11\}$.

$$J_{\text{partition}}(p) = \lambda \frac{\text{OLF}(p)}{\text{OLF}_{\text{max}}} + (1 - \lambda) \frac{\text{FPM}(p)}{\text{FPM}_{\text{max}}}$$
(4.88)

Figure 4.21 is a plot of $J_{\text{partition}}$ for the various p-partitions produced in Table 4.9 by the application of the agglomerative clustering approach. For $\lambda = 0.5$, it is observed that of all the partitions, the 4-partition (knee 2 in Figure 4.20) corresponds to the global minimum, resulting in the smallest value of $J_{\text{partition}}$. However, increasing the weight on optimality by changing λ to 0.85 causes the 2-partition (knee 1 in Figure 4.20) to provide the global minimum. This is consistent with the analysis presented above. However since a suitable value of λ is not obvious to decide, we prefer to use the optimality-robustness trade-off curve instead as the appropriate tool for the analysis of the partitions provided by agglomerative clustering.

p	p-partition from agglomeration
11	$\{1\} \{2\} \{3\} \{4\} \{5\} \{6\} \{7\} \{8\} \{9\} \{10\} \{11\}$
10	$\{1,4\}$ $\{2\}$ $\{3\}$ $\{5\}$ $\{6\}$ $\{7\}$ $\{8\}$ $\{9\}$ $\{10\}$ $\{11\}$
9	$\{1,3,4\}$ $\{2\}$ $\{5\}$ $\{6\}$ $\{7\}$ $\{8\}$ $\{9\}$ $\{10\}$ $\{11\}$
8	$\{1,3,4,5\}$ $\{2\}$ $\{6\}$ $\{7\}$ $\{8\}$ $\{9\}$ $\{10\}$ $\{11\}$
7	$\{1,2,3,4,5\}$ $\{6\}$ $\{7\}$ $\{8\}$ $\{9\}$ $\{10\}$ $\{11\}$
6	$\{1,2,3,4,5,6\}$ $\{7\}$ $\{8\}$ $\{9\}$ $\{10\}$ $\{11\}$
5	$\{1,2,3,4,5,6\}$ $\{7\}$ $\{8\}$ $\{9,10\}$ $\{11\}$
4	$\{1,2,3,4,5,6\}$ $\{7,8\}$ $\{9,10\}$ $\{11\}$
3	{1,2,3,4,5,6} {7,8} {9,10,11}
2	{1,2,3,4,5,6} {7,8,9,10,11}
1	{1,2,3,4,5,6,7,8,9,10,11}

Table 4.9 Partitions using agglomeration for 11-zone building



Figure 4.20 Optimality robustness trade-off curve for 11-zone building



Figure 4.21 Plot of $J_{\text{partition}}$ for partitions from agglomeration for 11-zone building

A validation of the above findings is provided by observations from open loop

simulations of the building thermal model. The building was partitioned into clusters corresponding to knee 2 in Figure 4.20. Cluster-level models, as discussed in Section 3.4.3, were obtained from the centralized model by decoupling the system along the boundaries (physical walls) of each cluster. An estimate of 20^o C was used to represent the temperature of the zones outside any cluster, i.e each element of the vector \mathbf{T}_{zj} in (3.24) is 20. Simulation results obtained for a period of 10 days are shown in Figure 4.22, where $T_{c,mean}$ is the capacity-weighted mean temperature of the zones, defined as

$$T_{c,mean}(k) = \frac{1}{c_z} \sum_{i=1}^{N_z} C_{z,i} T_{z,i}(k).$$
(4.89)

The relevant MATLAB codes are provided in Appendix D and the SIMULINK models are included in the media accompanying this thesis. In defining $T_{c,mean}$, the contribution of each zone's temperature is weighted by its thermal capacity. Since the thermal capacity of a zone is closely related to its volume, $T_{c,mean}$ indicates an effective temperature for the building as a fully decentralized corresponding whole. The response to the partition. i.e. $\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\},\{10\},\{11\}\}\$ is also plotted. In these simulations, the ambient temperature T_a and the unmodeled thermal disturbances $d_{z,i}$ (for all $i \in \{1, 2, ..., 11\}$) are assumed to be sinusoids with a 24 hour time-period as shown in Figure 4.23. The thermal disturbance $d_{w,i}$ is assumed to be 0 for all $i \in \{1, 2, ..., 11\}$ for simplicity. From Figure 4.22 it is observed that the error in predicting $T_{c,mean}$ is less than 5% for the knee 2 partition, whereas, the fully decentralized partition results in a maximum error of about 20% over the simulation time window. Here, errors are evaluated with respect to $T_{c,mean}$ predicted by the centralized case. This observation verifies that by partitioning the system using knee 2, the corresponding cluster-level decentralized models do not result in significant loss in inter-cluster thermal coupling, when compared to the centralized system model.

The 11-zone building example considered above demonstrates the benefit of using OLF-FPM partitioning approach, as opposed to physical intuition, which may be absent or can only provide limited insight. For example, the partition {{1,2,3,4,5,6},{7,8,9,10,11}} arrived at using the intuition of separation along the thermally insulating hallways generates a knee in Figure 4.20. However, as the analysis presented above clearly demonstrates, it is not the most appropriate choice.



Figure 4.22 Open loop response analysis for partitions of 11-zone building



(a) Ambient temperature profile

Figure 4.23 (cont. on next page)



(b) Unmodeled thermal load profile for each zone

Figure 4.23 Sinusoidal disturbance profiles used in open loop analysis of 11-zone building

4.3.12 Remarks

The main advantage of the OLF-FPM approach over the CLF-MCS approach is that the optimality and robustness metrics are analytically derived and directly correlated with the notions of optimality and robustness. In particular, OLF is related to an upper bound on the deviation in performance between centralized control and decentralized control. Similarly, FPM by definition represents the expected value of the % volume of building affected in the event of failures. On the other hand, a quantifiable relationship between CLF and optimality or MCS and robustness is not available. However the CLF-MCS approach is useful because it provides an initial framework for partitioning which is improved upon by the OLF-FPM method. In the remainder of this work, only the OLF-FPM method is considered for making control architecture decisions because of its above stated benefits over the CLF-MCS method.

Chapter 5 Control Design with Optimal Architecture

5.1 Introduction

Chapter 4 presented two different approaches for the determination of decentralized control architectures which balance the underlying optimality and robustness requirements in the thermal control of buildings. However, the design of decentralized controllers based on the control architectures determined using these methodologies was not considered. This chapter seeks to address this requirement by focusing on the design aspects of decentralized control for a given architecture. Two key challenges are envisioned in this regard: all system states are not measurable and there are several unmodeled disturbances. These practical concerns are addressed in this chapter by first developing reduced order model representations for the thermal dynamics then using decentralized extended state observers to simultaneously estimate the unknown states and disturbances. This is followed by control design based on the Model Predictive Control (MPC) framework, since MPC has been extensively applied in the building systems control literature because of its proven effectiveness in handling large-scale constrained optimal control problems (see discussion in section 3.1.2.2).

The decentralized control design methodology presented in this chapter differs from Chapter 3 in three aspects. Firstly, the model used for control design is based on a three resistor – two capacitor (3R2C) framework to represent the thermal dynamics of walls, which is a more accurate representation than the two resistor – one capacitor (2R1C) framework that was used in Chapter 3. Secondly, the controllers are output-feedback as opposed to the state-feedback framework presented in chapter 3, considering the fact that typically only zone temperature measurements are available in a building via the thermostats. Lastly, the control design considers certain practical constraints which were ignored in Chapter 3 for simplicity. The nomenclature used in this chapter is shown in Table 5.1.

5.2 Cluster level modeling

In this section, we consider a p-partition of the building (definition 3.2) and develop models to describe the thermal dynamics associated with the clusters which constitute the partition. These models are used for the design of decentralized observers and controllers as described in subsequent sections.

Symbol	Description
N _w	Number of walls surfaces in the building
Nz	Number of zones in the building
N _{zi}	Number of zones in <i>i</i> th cluster
T _w	Vector of wall surface temperatures
T_w^i	Vector of wall surface temperatures in i^{th} cluster
$T^i_{w,k}$	Temperature of k^{th} wall surface in i^{th} cluster
Tz	Vector of zone temperatures
Tzi	Vector of zone temperatures in <i>i</i> th cluster
T ⁱ _{z,ref}	Vector of zone temperature set-points in i^{th} cluster
T ⁱ _{z,meas}	Vector of zone temperature measurements in i^{th} cluster
T _a	Ambient temperature
Tg	Temperature of ground below building
$T_{z,n}^i$	Temperature of n^{th} zone in i^{th} cluster
T _{supp}	Temperature of cold air supplied to each zone
d _w	Vector of unknown thermal loads acting on wall surfaces
dz	Vector of unknown thermal loads acting on zones

Table 5.1 Nomenclature of common symbols in Chapter 5

dw	Vector of unknown thermal loads acting on wall surfaces in i^{th} cluster
dzi	Vector of unknown thermal loads acting on zones in i^{th} cluster
\bar{d}^i_w	Vector of aggregated thermal loads acting on wall surfaces in reduced order
	model for i^{th} cluster
\bar{d}_w^i	Vector of aggregated thermal loads acting on zones in reduced order model
	for <i>ith</i> cluster
$C_{w,k}^i$	Thermal capacitance associated with k^{th} wall surface in i^{th} cluster
$C_{z,n}^i$	Thermal capacitance associated with n^{th} zone in i^{th} cluster
C _{pa}	Specific heat capacity of air
x(k+l k)	Projected value of quantity x after l time steps in future, given $x(k)$. Note
	x(k k) = x(k).
u	Vector of control inputs
u ⁱ	Vector of control inputs in <i>i</i> th cluster
u_n^i	Control input for n^{th} zone in i^{th} cluster
$\dot{m}^i_{max,n}$	Maximum air mass flow rate for n^{th} zone in i^{th} cluster
$\dot{Q}^i_{RH-max,n}$	Maximum reheating available for n^{th} zone in i^{th} cluster
$0_{m imes n}$	Zero matrix of dimension $m \times n$
In	Identity matrix of dimension $m \times n$

5.2.1 3R2C modeling framework

We consider a 3R2C modeling paradigm [40], where each wall is represented by a set of 3 resistances and 2 capacitances (Figure 5.1), whereas, each room is treated as a single capacitance. This is an improvement over the 2R1C paradigm considered in chapter 3 for control architecture selection. In Figure 5.1, $C_{w-in,i}$ and $C_{w-out,i}$ are capacitances for the *i*th wall, the states associated with them being the temperatures of its two surfaces. For internal walls (flanked by zones on both sides), the designation of "in" and "out" for the wall surfaces is arbitrary. However, for external walls (facing the ambient or ground on one side and a zone on the other), by convention "out" refers to the surface which faces the ambient/ground and "in"

refers to the surface facing the zone. $R_{w-in,i}$ and $R_{w-out,i}$ in Figure 5.1 are thermal resistances between the surfaces of wall *i* and the appropriate elements (zones, ambient or ground) which they thermally interact with. $R_{ww,i}$ represents the resistance between inner and outer wall surfaces. Similar to the 2R1C framework, each zone in the building is represented by a lumped capacitance.

The states of the system are the wall surface temperatures and zone temperatures. The control inputs represent the rates of energy transfer – positive for heating, and negative for cooling — provided to the zones by the HVAC system. The ambient and ground temperatures are considered as measured disturbances. Various other factors also affect the thermal dynamics which are treated as unknown disturbances. These include long-wave and short-wave radiation heat transfers affecting the walls, and thermal loads from occupants, appliances and lighting which affect the zones. In this work, these factors are not modeled separately and only their lumped thermal contribution to each zone and wall is represented using disturbance vectors d_z and d_w .



Figure 5.1 Schematic of 3R2C modeling paradigm for wall *i*. Un-modeled thermal loads acting on the wall's surfaces are also shown.

The ensuing linear state space model is of the form (5.1), with reference to the notation described in the nomenclature (Table 5.1). It is obtained by applying Algorithm 5.1 presented below on a RC network for the building constructed using the framework presented above. Note

that Algorithm 5.1 is an extension of Algorithm 3.1 to the 3R2C case.

$$\frac{d}{dt} \begin{bmatrix} \mathbf{T}_{\mathbf{w}} \\ \mathbf{T}_{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathbf{w}\mathbf{w}} & \mathbf{A}_{\mathbf{w}\mathbf{z}} \\ \mathbf{A}_{\mathbf{z}\mathbf{w}} & \mathbf{A}_{\mathbf{z}\mathbf{z}} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{\mathbf{w}} \\ \mathbf{T}_{\mathbf{z}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{\mathbf{z}} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{B}_{\mathbf{a}} & \mathbf{B}_{\mathbf{g}} & \mathbf{B}_{\mathbf{d}\mathbf{w}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{\mathbf{d}\mathbf{z}} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{a} \\ \mathbf{T}_{g} \\ \mathbf{d}_{w} \\ \mathbf{d}_{z} \end{bmatrix}.$$
(5.1)

Algorithm 5.1. Generation of state space model for building thermal dynamics using 3R2C framework

STEP 1: A weighted graph is created with nodes for each of the wall surfaces, the zones, the ambient and ground. While numbering the nodes, those representing wall surfaces are numbered first, followed by the zones, the ambient and lastly the ground. Each wall surface node is connected by undirected edges to the two nodes (zones, ambient or ground) to which it is connected via resistances in the RC network. Additionally, the inside and outside surfaces of each wall are also connected by an edge. The weight of each edge is set to be the inverse of the corresponding thermal resistance between the two nodes it connects. The resulting weighted graph is denoted by G = (V, E) along with the weight function, $w: E \mapsto \mathbb{R}^+$. We also define capacitance matrices, C_w and C_z which are diagonal matrices of the thermal capacitances associated with the wall surfaces and the zones respectively. The diagonal entries in these matrices are entered in the order of the corresponding node numbers in G.

STEP 2: The Laplacian matrix of G, denoted by L_G is then obtained as:

$$\mathbf{L}_{\mathbf{G}} = \mathbf{D}_{\mathbf{G}} - \mathbf{A}_{\mathbf{G}}.\tag{5.2}$$

where,

$$\mathbf{A}_{\mathbf{G}}(i,j) = \begin{cases} w(i,j) & \text{if}(i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$
$$\mathbf{D}_{\mathbf{G}}(i,i) = \sum_{j} A_{G}(i,j) .$$

Note that D_G is a diagonal matrix. We extract the following sub-matrices from L_G :

- 1. \mathbf{L}_{ww} is the square sub-matrix of $\mathbf{L}_{\mathbf{G}}$ which corresponds to its first N_w rows and first N_w columns.
- 2. L_{wz} is the sub-matrix of L_G which corresponds to its first N_w rows and the columns $N_w + 1$, $N_w + 2$, ..., $N_w + N_z$.
- 3. L_{zw} is the sub-matrix of L_G which corresponds to the rows $N_w + 1$, $N_w + 2$, ..., $N_w + N_z$ and the first N_w columns.
- 4. L_{zz} is the sub-matrix of L_G which corresponds to the rows $N_w + 1$, $N_w + 2$, ..., $N_w + N_z$ and the columns $N_w + 1$, $N_w + 2$, ..., $N_w + N_z$.
- 5. $\mathbf{L}_{\mathbf{a}}$ is a column vector which corresponds to the first N_w rows and the $(N_w + N_z + 1)^{th}$ column of $\mathbf{L}_{\mathbf{G}}$
- 6. $\mathbf{L}_{\mathbf{g}}$ is a column vector which corresponds to the first N_w rows and the $(N_w + N_z + 2)^{th}$ column of $\mathbf{L}_{\mathbf{G}}$

STEP 3: The matrices appearing in (5.1) are then obtained as follows:

$$\mathbf{A}_{\mathbf{w}\mathbf{w}} = -\mathbf{C}_{\mathbf{w}}^{-1}\mathbf{L}_{\mathbf{w}\mathbf{w}},\tag{5.3}$$

$$\mathbf{A}_{\mathbf{w}\mathbf{z}} = -\mathbf{C}_{\mathbf{w}}^{-1}\mathbf{L}_{\mathbf{w}\mathbf{z}},\tag{5.4}$$

$$\mathbf{A}_{\mathbf{zw}} = -\mathbf{C}_{\mathbf{z}}^{-1}\mathbf{L}_{\mathbf{zw'}} \tag{5.5}$$

$$\mathbf{A}_{\mathbf{z}\mathbf{z}} = -\mathbf{C}_{\mathbf{z}}^{-1}\mathbf{L}_{\mathbf{z}\mathbf{z}},\tag{5.6}$$

$$\mathbf{B}_{\mathbf{a}} = -\mathbf{C}_{\mathbf{w}}^{-1}\mathbf{L}_{\mathbf{a}},\tag{5.7}$$

$$\mathbf{B}_{\mathbf{g}} = -\mathbf{C}_{\mathbf{w}}^{-1}\mathbf{L}_{\mathbf{g}},\tag{5.8}$$

$$\mathbf{B}_{\mathbf{dw}} = \mathbf{C}_{\mathbf{w}}^{-1}, \tag{5.9}$$

$$B_{dz} = C_z^{-1}$$
. (5.10)

5.2.2 Full order cluster level model

We consider a p-partition of the building. The model for the i^{th} ($i \in \{1, 2, ..., p\}$) cluster is obtained from (5.1) by extracting the dynamics of the walls and the zones constituting the cluster as shown below (see nomenclature)
$$\frac{d}{dt} \begin{bmatrix} \mathbf{T}_{\mathbf{w}}^{\mathbf{i}} \\ \mathbf{T}_{\mathbf{z}}^{\mathbf{i}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathbf{ww}}^{\mathbf{i}} & \mathbf{A}_{\mathbf{wz}}^{\mathbf{i}} \\ \mathbf{A}_{\mathbf{zw}}^{\mathbf{i}} & \mathbf{A}_{\mathbf{zz}}^{\mathbf{i}} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{\mathbf{w}}^{\mathbf{i}} \\ \mathbf{T}_{\mathbf{z}}^{\mathbf{i}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{\mathbf{z}}^{\mathbf{i}} \end{bmatrix} \mathbf{u}^{\mathbf{i}} + \begin{bmatrix} \mathbf{B}_{\mathbf{a}}^{\mathbf{i}} & \mathbf{B}_{\mathbf{g}}^{\mathbf{i}} & \mathbf{B}_{\mathbf{dw}}^{\mathbf{i}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{\mathbf{dz}}^{\mathbf{i}} \end{bmatrix} \begin{bmatrix} T_{a} \\ T_{g} \\ \mathbf{d}_{\mathbf{w}}^{\mathbf{i}} \\ \mathbf{d}_{\mathbf{z}}^{\mathbf{i}} \end{bmatrix} + \sum_{j \neq i} \begin{bmatrix} \mathbf{B}_{\mathbf{w},\mathbf{z}}^{\mathbf{i}j} \\ \mathbf{0} \end{bmatrix} \widehat{\mathbf{T}}_{\mathbf{z}}^{\mathbf{j}} . (5.11)$$

The dynamics of the *i*th cluster depends on the zone temperatures in the other clusters \mathbf{T}_{z}^{j} , where $j \neq i$. However, imposition of a control architecture that is decentralized with respect to the clusters implies that the *i*th control agent does not have access to the sensory data from the other p-1 clusters. Therefore, as mentioned in chapter 3, \mathbf{T}_{z}^{j} for each $j \neq i$ must be replaced with an appropriate guess or estimate $\mathbf{\widehat{T}}_{z}^{j}$ as shown in (5.11).

The states $\mathbf{T}_{\mathbf{w}}^{\mathbf{i}}$, and the disturbances $\mathbf{d}_{\mathbf{w}}^{\mathbf{i}}$ and $\mathbf{d}_{\mathbf{z}}^{\mathbf{i}}$ are unknown and therefore (5.11) cannot be directly used for control design. This concern is addressed in the remainder of this section.

5.2.3 Model order reduction

For a building with N_z zones, the number of states in the model (5.1) is of the order of $7N_z$ [29]. This motivates the development of reduced order models, which can be used for control design at the overall building level (centralized architecture) or the cluster level (decentralized architecture). An aggregation based methodology was proposed in [29] for the development of reduced order models of building thermal dynamics. A particular advantage is that the reduced order models can also be represented by RC networks. In this paper, we adopt the methodology in [29] to seek an observable reduced order model representation which allows extended state observers to estimate the unmeasured states and disturbances.

Algorithm 5.2, presented below, outlines steps for model reduction via aggregation of states, which eventually lead to an observable representation in Section 5.2.4. The resulting state space model for the *i*th cluster obtained from the algorithm consists of the states $\overline{T}_{w-in,m}^{i}$, $\overline{T}_{w-out,m}^{i}$ and $T_{z,m}^{i}$, where $m \in \{1, 2, ..., N_{zi}\}$ and is given by

$$\frac{d}{dt}\underbrace{\begin{bmatrix}\overline{T}_{w-in}^{i}\\ \overline{T}_{w-out}^{i}\\ \underline{T_{z}^{i}}^{i}\end{bmatrix}}_{x^{i}} = \underbrace{\begin{bmatrix}\overline{A}_{w-in,w-in}^{i} & \overline{A}_{w-in,w-out}^{i} & \overline{A}_{w-in,z}^{i}\\ \overline{A}_{w-out,w-in}^{i} & \overline{A}_{w-out,w-out}^{i} & \mathbf{0}\\ \overline{A}_{z,w-in}^{i} & \mathbf{0} & \overline{A}_{z,z}^{i}\end{bmatrix}}_{\overline{A}^{i}} \begin{bmatrix}\overline{T}_{w-in}^{i}\\ \overline{T}_{w-out}^{i}\\ T_{z}^{i}\end{bmatrix}} + \underbrace{\begin{bmatrix}\mathbf{0}\\\mathbf{0}\\ \overline{B}_{z,u}^{i}\\ \overline{B}_{u}^{i}\end{bmatrix}}_{\overline{B}_{u}^{i}} u^{i}$$

$$+ \left[\underbrace{\begin{matrix} \mathbf{0} & \mathbf{0} & \overline{\mathbf{B}}_{w-\mathrm{in},\mathrm{dw}}^{i} & \mathbf{0} \\ \underline{\mathbf{0}} & \underline{\mathbf{0}} & \overline{\mathbf{B}}_{w-\mathrm{out},\mathrm{g}}^{i} & \overline{\mathbf{B}}_{w-\mathrm{out},\mathrm{dw}}^{i} & \mathbf{0} \\ \underline{\mathbf{0}} & \underline{\mathbf{0}} & \underline{\mathbf{0}} & \overline{\mathbf{B}}_{w-\mathrm{out},\mathrm{dw}}^{i} & \overline{\mathbf{B}}_{w-\mathrm{out},\mathrm{dw}}^{i} \\ \underline{\mathbf{0}} & \underline{\mathbf{0}} & \underline{\mathbf{0}} & \overline{\mathbf{B}}_{w-\mathrm{out},\mathrm{dw}}^{i} \\ \overline{\mathbf{B}}_{\mathrm{dw}}^{i} & \overline{\mathbf{B}}_{\mathrm{dz}}^{i} \\ \end{bmatrix} \left[\underbrace{\begin{matrix} T_{a} \\ T_{g} \\ \overline{\mathbf{d}}_{w}^{i} \\ \overline{\mathbf{d}}_{w}^{i} \\ \overline{\mathbf{d}}_{z}^{i} \\ \end{bmatrix} + \sum_{j \neq i} \underbrace{\begin{bmatrix} \mathbf{0} \\ \overline{\mathbf{B}}_{w-\mathrm{out},z}^{i} \\ \underline{\mathbf{0}} \\ \overline{\mathbf{B}}_{z}^{i} \\ \overline{\mathbf{B}}_{z}^{i} \\ \end{bmatrix}}_{\overline{\mathbf{B}}_{z}^{ij}} \widehat{\mathbf{T}}_{z}^{i} (5.12)$$

Here,

$$\overline{\mathbf{T}}_{\mathbf{w}-\mathbf{in}}^{\mathbf{i}} = \begin{bmatrix} \overline{T}_{w-in,1}^{i} & \overline{T}_{w-in,2}^{i} & \dots & \overline{T}_{w-in,n_{w-in}}^{i} \end{bmatrix}^{T}, \qquad (5.13)$$

$$\overline{\mathbf{T}}_{\mathbf{w}-\mathbf{out}}^{i} = \begin{bmatrix} \overline{T}_{w-out,1}^{i} & \overline{T}_{w-out,2}^{i} & \dots & \overline{T}_{w-out,n_{w-out}}^{i} \end{bmatrix}^{T}.$$
 (5.14)

In the above equations, n_{w-in}^{i} and n_{w-out}^{i} are defined in step 1 of Algorithm 5.2.

Algorithm 5.2. Model reduction for the *i*th cluster via aggregation of states

STEP 1: For each zone $m \in \{1, 2, ..., N_{zi}\}$ in the cluster, all wall surfaces belonging to the walls that enclose it are identified (note that a wall has two surfaces, each represented by a capacitance in the 3R2C framework). Of these, we denote those surfaces that directly face the zone by the set \mathcal{N}_m^i . Among the remaining surfaces which do not face the zone *m* directly, those which also do not face another zone in the cluster directly are identified. The set of these surfaces is denoted by $\overline{\mathcal{N}}_m^i$. Note that $\overline{\mathcal{N}}_m^i$ can be an empty set. We define n_{w-in}^i as the number of non-empty sets \mathcal{N}_m^i , where $m \in \{1, 2, ..., N_{zi}\}$. Similarly, n_{w-in}^i is defined as the number of non-empty sets $\overline{\mathcal{N}}_m^i$, where $m \in \{1, 2, ..., N_{zi}\}$. Note that $n_{w-in}^i = N_{zi}$ because each zone is surrounded by at least one wall surface.

STEP 2: For each zone $m \in \{1, 2, ..., N_{zi}\}$, the temperatures of all wall surfaces in the set \mathcal{N}_m^i and $\overline{\mathcal{N}}_m^i$ are aggregated into single states $\overline{T}_{w-in,m}^i$ and $\overline{T}_{w-out,n}^i$ respectively as shown (refer to nomenclature)

$$\bar{T}_{w-in,m}^{i} = \frac{\sum_{k \in \mathcal{N}_{m}^{i}} C_{w,k}^{i} T_{w,k}^{i}}{\sum_{k \in \mathcal{N}_{m}^{i}} C_{w,k}^{i}}, \qquad (5.15)$$

$$\bar{T}_{w-out,m}^{i} = \frac{\sum_{k \in \bar{N}_{m}^{i}} c_{w,k}^{i} T_{w,k}^{i}}{\sum_{k \in \bar{N}_{m}^{i}} c_{w,k}^{i}}.$$
(5.16)

STEP 3: Equivalent capacitances $\bar{C}_{w-in,m}^i$ and $\bar{C}_{w-out,m}^i$ associated with the states $\bar{T}_{w-in,m}^i$ and $\bar{T}_{w-out,m}^i$ respectively are computed as shown

$$\bar{C}^i_{w-in,m} = \sum_{k \in \mathcal{N}^i_m} C^i_{w,k} , \qquad (5.17)$$

$$\bar{C}^{i}_{w-out,m} = \sum_{k \in \bar{\mathcal{N}}^{i}_{m}} C^{i}_{w,k} .$$
(5.18)

STEP 4: For each $m \in \{1, 2, ..., N_{zi}\}, \overline{R}^{i}_{w-in,w-out,m}$ is defined as the parallel equivalent of all resistances which connect an element in \mathcal{N}^{i}_{m} with an element in $\overline{\mathcal{N}}^{i}_{m}$. The capacitors $\overline{C}^{i}_{w-in,m}$ and $\overline{C}^{i}_{w-out,m}$ are then connected using $\overline{R}^{i}_{w-in,w-out,m}$.

STEP 5: For each $m, n \in \{1, 2, ..., N_{zi}\}, \overline{R}_{w-in, z, m, n}^{i}$ is defined as the parallel equivalent of all resistances which connect an element in \mathcal{N}_{m}^{i} with zone *n*. The capacitors $\overline{C}_{w-in,m}^{i}$ and $C_{z,n}^{i}$ (see nomenclature) are connected using $\overline{R}_{w-in, z, m, n}^{i}$.

STEP 6: For each $m, n \in \{1, 2, ..., N_{zi}\}, \overline{R}_{w-in,w-in,m,n}^{i}$ is defined as the parallel equivalent of all resistances which connect an element in \mathcal{N}_{m}^{i} with an element in \mathcal{N}_{n}^{i} . The capacitors $\overline{C}_{w-in,m}^{i}$ and $\overline{C}_{w-in,n}^{i}$ are connected using $\overline{R}_{w-in,w-in,m,n}^{i}$.

STEP 7: For each $m \in \{1, 2, ..., N_{zi}\}$, $\overline{R}_{w-out,a,m}^{i}$ is defined as the parallel equivalent of all resistances which connect an element in $\overline{\mathcal{N}}_{m}^{i}$ with the ambient. The capacitor $\overline{C}_{w-out,m}^{i}$ and ambient are connected with $\overline{R}_{w-out,a,m}^{i}$. Similarly, $\overline{R}_{w-out,g,m}^{i}$ is defined for the ground instead of ambient.

STEP 8: For each cluster $j \in \{1, 2, ..., p\}$, $j \neq i$ and for each zone $m \in \{1, 2, ..., N_{zi}\}$ in cluster *i* and zone $n \in \{1, 2, ..., N_{zj}\}$ in cluster *j*, we define the resistance $\overline{R}_{w-out,z,m,n}^{i,j}$ as the parallel

equivalent of all resistances which connect an element in $\overline{\mathcal{N}}_m^i$ with the zone *n*. The capacitor $\overline{C}_{w-out,m}^i$ is connected to the external zone *n* (lying outside cluster *i*) using $\overline{R}_{w-out,z,m,n}^{i,j}$.

STEP 9: The reduced order RC network is constituted by the capacitances and resistances created in steps 3 - 8.

As an example, consider the 3-zone building shown in Figure 5.2 which is similar to the example in Figure 3.3, except that it is modeled using the 3R2C framework. The zones are marked as z1 - z3. The building has 20 wall surfaces which are marked as w1 - w20. The resistances representing heat transfer paths between the wall surfaces, zones, ambient and ground are also shown in the figure. The capacitances (not shown in the figure) corresponding to the wall surfaces and the zones are given by $C_{w1} - C_{w20}$ and $C_{z1} - C_{z3}$ respectively. Let the temperatures associated with these capacitances be represented by $T_{w1} - T_{w20}$ and $T_{z1} - T_{z3}$.



Figure 5.2 Schematic of 3-zone building used for illustration of Algorithm 5.2

Consider a 2-partition $\{\{1,2\},\{3\}\}\$ of the building. Application of each step of Algorithm 5.2 to the first cluster $\{1,2\}$ is shown below.

STEP 1:
$$\mathcal{N}_1^1 = \{w1, w2, w3, w4\}, \overline{\mathcal{N}}_1^1 = \{w13, w14, w15\}, \mathcal{N}_2^1 = \{w5, w6, w7, w8\}$$

and $\overline{\mathcal{N}}_2^1 = \{w16, w9, w17\}.$

STEP 2:
$$\bar{T}_{w-in,1}^{1} = \frac{c_{w1}T_{w1} + c_{w2}T_{w2} + c_{w3}T_{w3} + c_{w4}T_{w4}}{c_{w1} + c_{w2} + c_{w3} + c_{w4}}, \ \bar{T}_{w-out,1}^{1} = \frac{c_{w13}T_{w13} + c_{w14}T_{w14} + c_{w15}T_{w15}}{c_{w13} + c_{w14} + c_{w15}},$$

$$\bar{T}_{w-in,2}^{1} = \frac{C_{w5}T_{w5} + C_{w6}T_{w6} + C_{w7}T_{w7} + C_{w8}T_{w8}}{C_{w5} + C_{w6} + C_{w7} + C_{w8}} \text{ and } \bar{T}_{w-out,2}^{1} = \frac{C_{w16}T_{w16} + C_{w9}T_{w9} + C_{w17}T_{w17}}{C_{w16} + C_{w9} + C_{w17}}$$

STEP 3: $\bar{C}_{w-in,1}^1 = C_{w1} + C_{w2} + C_{w3} + C_{w4}$, $\bar{C}_{w-out,1}^1 = C_{w13} + C_{w14} + C_{w15}$, $\bar{C}_{w-in,2}^1 = C_{w5} + C_{w6} + C_{w7} + C_{w8}$ and $\bar{C}_{w-out,2}^1 = C_{w16} + C_{w9} + C_{w17}$.

STEP 4:
$$\frac{1}{\bar{R}^1_{w-in,w-out,1}} = \frac{1}{R_{w2,w14}} + \frac{1}{R_{w13,w1}} + \frac{1}{R_{w4,w15}} \text{ and } \frac{1}{\bar{R}^1_{w-in,w-out,2}} = \frac{1}{R_{w6,w16}} + \frac{1}{R_{w7,w9}} + \frac{1}{R_{w8,w17}}.$$

STEP 5:
$$\frac{1}{\bar{R}^1_{w-in,z,1,1}} = \frac{1}{R_{w1,z1}} + \frac{1}{R_{w2,z1}} + \frac{1}{R_{w3,z1}} + \frac{1}{R_{w4,z1}}, \frac{1}{\bar{R}^1_{w-in,z,1,2}} = 0, \frac{1}{\bar{R}^1_{w-in,z,2,1}} = 0$$
, and

$$\frac{1}{\bar{R}^{1}_{w-in,z,2,2}} = \frac{1}{R_{w6,z2}} + \frac{1}{R_{w7,z2}} + \frac{1}{R_{w8,z2}} + \frac{1}{R_{w5,z2}} \,.$$

STEP 6:
$$\frac{1}{\bar{R}^1_{w-in,w-in,1,2}} = \frac{1}{R_{w3,w5}}$$
.

STEP 7:
$$\frac{1}{\bar{R}^1_{w-out,a,1}} = \frac{1}{R_{w13,a}} + \frac{1}{R_{w14,a}}$$
, $\frac{1}{\bar{R}^1_{w-out,g,1}} = \frac{1}{R_{w15,g}}$, $\frac{1}{\bar{R}^1_{w-out,a,2}} = \frac{1}{R_{w16,a}}$ and $\frac{1}{\bar{R}^1_{w-out,g,2}} = \frac{1}{R_{w17,g}}$

STEP 8:
$$\frac{1}{\bar{R}_{w-out,z,1,1}^{1,2}} = 0$$
 and $\frac{1}{\bar{R}_{w-out,z,2,1}^{1,2}} = \frac{1}{R_{w9,z3}}$.

STEP 9: The reduced order RC network for the 3-zone building is shown in Figure 5.3. It has

only 6 states compared to 23 states in the full order model.



Figure 5.3 Reduced order RC network obtained via Algorithm 5.2 for the cluster {1,2} of the 3-zone building in Figure 5.2

5.2.4 State transformation

The states $\overline{\mathbf{T}}_{w-in}^{i}$ and $\overline{\mathbf{T}}_{w-out}^{i}$, and disturbances $\overline{\mathbf{d}}_{w}^{i}$ and $\overline{\mathbf{d}}_{z}^{i}$ in (5.12) are unknown. An extended state observer (ESO) [89, 90] can be designed to estimate them based on assumptions on the dynamics of the unknown disturbances. In this paper, we assume that the unknown

disturbances are slowly time-varying quantities. Therefore, if the time window of interest for control design – such as prediction horizon in case of MPC – is sufficiently small, e.g. 30 minutes to an hour, we can assume that the disturbances are constant, that is $\dot{\mathbf{d}}_{w}^{i} \approx \mathbf{0}$ and $\dot{\mathbf{d}}_{z}^{i} \approx \mathbf{0}$. This is used to augment the dynamics in (5.12) as shown

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x}^{i} \\ \mathbf{\bar{d}}^{i}_{w} \\ \mathbf{\bar{d}}^{i}_{z} \end{bmatrix} = \begin{bmatrix} \mathbf{\bar{A}}^{i} & \mathbf{\bar{B}}^{i}_{dw} & \mathbf{\bar{B}}^{i}_{dz} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{i} \\ \mathbf{\bar{d}}^{i}_{w} \\ \mathbf{\bar{d}}^{i}_{z} \end{bmatrix} + \begin{bmatrix} \mathbf{\bar{B}}^{i}_{u} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^{i} + \begin{bmatrix} \mathbf{\bar{B}}^{i}_{a} & \mathbf{\bar{B}}^{i}_{g} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} T_{a} \\ T_{g} \end{bmatrix} + \sum_{j \neq i} \begin{bmatrix} \mathbf{\bar{B}}^{ij}_{z} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{\widehat{T}}^{j}_{z} . \quad (5.19)$$

A limitation of representation (5.19) is that it does not guarantee observability when measurements are only available for the zone temperatures T_z^i in the cluster. To address this limitation, we define new states

$$\eta_1^i = \overline{\mathbf{T}}_w^i + \left(\overline{\mathbf{A}}_{w,w}^i\right)^{-1} \overline{\mathbf{B}}_{w,dw}^i \overline{\mathbf{d}}_w^i, \qquad (5.20)$$

$$\eta_2^i = \overline{B}_{z,dz}^i \overline{d}_z^i - \overline{A}_{z,w}^i \left(\overline{A}_{w,w}^i\right)^{-1} \overline{B}_{w,dw}^i \overline{d}_w^i , \qquad (5.21)$$

where,

$$\begin{split} \overline{T}_{w}^{i} &= \begin{bmatrix} \overline{T}_{w-in}^{i} \\ \overline{T}_{w-out}^{i} \end{bmatrix}, \overline{A}_{w,w}^{i} = \begin{bmatrix} \overline{A}_{w-in,w-in}^{i} & \overline{A}_{w-in,w-out}^{i} \\ \overline{A}_{w-out,w-in}^{i} & \overline{A}_{w-out,w-out}^{i} \end{bmatrix}, \overline{B}_{w,dw}^{i} = \begin{bmatrix} \overline{B}_{w-in,dw}^{i} \\ \overline{B}_{w-out,dw}^{i} \end{bmatrix}, \\ \text{and} \quad \overline{A}_{z,w}^{i} &= \begin{bmatrix} \overline{A}_{z,w-in}^{i} & \mathbf{0}_{N_{zi} \times n_{w-out}^{i}} \end{bmatrix}. \end{split}$$

The transformed state space model using the new states η_1^i and η_2^i is given by

$$\frac{d}{dt} \begin{bmatrix} \boldsymbol{\eta}_{1}^{i} \\ \boldsymbol{T}_{z}^{i} \\ \boldsymbol{\eta}_{2}^{i} \end{bmatrix} = \underbrace{\begin{bmatrix} \overline{A}_{w,w}^{i} & \overline{A}_{w,z}^{i} & \mathbf{0} \\ \overline{A}_{z,w}^{i} & \overline{A}_{z,z}^{i} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}_{\overline{A}_{tfo}^{i}} \begin{bmatrix} \boldsymbol{\eta}_{1}^{i} \\ \boldsymbol{T}_{z}^{i} \\ \boldsymbol{\eta}_{2}^{i} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \overline{B}_{z,u}^{i} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^{i} + \begin{bmatrix} \overline{B}_{w,a}^{i} & \overline{B}_{w,g}^{i} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} T_{a} \\ T_{g} \end{bmatrix} + \sum_{j \neq i} \begin{bmatrix} \overline{B}_{w,z}^{ij} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \widehat{\mathbf{T}}_{z}^{j}, \quad (5.22)$$

where,

$$\overline{\mathbf{A}}_{\mathbf{w},\mathbf{z}}^{\mathbf{i}} = \begin{bmatrix} \overline{\mathbf{A}}_{\mathbf{w}-\mathbf{in},\mathbf{z}}^{\mathbf{i}} \\ \mathbf{0}_{n_{w-out}^{i} \times N_{z}}^{\mathbf{i}} \end{bmatrix}, \ \overline{\mathbf{B}}_{\mathbf{w},\mathbf{a}}^{\mathbf{i}} = \begin{bmatrix} \mathbf{0}_{n_{w-in}^{i} \times 1}^{\mathbf{i}} \\ \overline{\mathbf{B}}_{\mathbf{w},\mathbf{g}}^{\mathbf{i}} = \begin{bmatrix} \mathbf{0}_{n_{w-in}^{i} \times 1}^{\mathbf{i}} \\ \overline{\mathbf{B}}_{\mathbf{w}-\mathbf{out},\mathbf{g}}^{\mathbf{i}} \end{bmatrix}, \ \text{and} \ \overline{\mathbf{B}}_{\mathbf{w},\mathbf{z}}^{\mathbf{ij}} = \begin{bmatrix} \mathbf{0}_{n_{w-in}^{i} \times N_{zj}}^{\mathbf{i}} \\ \overline{\mathbf{B}}_{\mathbf{w}-\mathbf{out},\mathbf{z}}^{\mathbf{ij}} \end{bmatrix}$$

The measurement model corresponding to (5.22) is given by

$$\mathbf{T}_{z,\text{meas}}^{i} = \underbrace{\begin{bmatrix} \mathbf{0}_{N_{zi} \times (n_{w-out}^{i} + n_{w-in}^{i})} & \mathbf{I}_{N_{zi}} & \mathbf{0}_{N_{zi} \times N_{zi}} \end{bmatrix}}_{\mathbf{C}_{\text{tfo}}^{i}} \begin{bmatrix} \mathbf{\eta}_{1}^{i} \\ \mathbf{T}_{z}^{i} \\ \mathbf{\eta}_{2}^{i} \end{bmatrix}}$$
(5.23)

Theorem 5.1: *The pair* $(\overline{A}_{tfo}^{i}, C_{tfo}^{i})$ *is observable.*

Proof: Using the definitions of $\overline{A}_{w,w}^i$, $\overline{A}_{w,z}^i$, $\overline{A}_{z,w}^i$ and $\overline{A}_{z,z}^i$ above, \overline{A}_{tfo}^i can be written as

$$\overline{\mathbf{A}}_{tfo}^{i} = \begin{bmatrix} \overline{\mathbf{A}}_{w-in,w-in}^{i} & \overline{\mathbf{A}}_{w-in,w-out}^{i} & \overline{\mathbf{A}}_{w-in,z}^{i} & \mathbf{0} \\ \overline{\mathbf{A}}_{w-out,w-in}^{i} & \overline{\mathbf{A}}_{w-out,w-out}^{i} & \mathbf{0} & \mathbf{0} \\ \overline{\mathbf{A}}_{z,w-in}^{i} & \mathbf{0} & \overline{\mathbf{A}}_{z,z}^{i} & \mathbf{I}_{N_{zi}} \\ \mathbf{0}_{N_{zi}\times n_{w-in}^{i}}^{i} & \mathbf{0}_{N_{zi}\times n_{w-out}^{i}}^{i} & \mathbf{0}_{N_{zi}\times N_{zi}}^{i} & \mathbf{0}_{N_{zi}\times N_{zi}} \end{bmatrix}.$$
(5.24)

For simplicity of notation, we define $A_{11} = \overline{A}^{i}_{w-in,w-in}$, $A_{12} = \overline{A}^{i}_{w-in,w-out}$, $A_{13} = \overline{A}^{i}_{w-in,z}$, $A_{21} = \overline{A}^{i}_{w-out,w-in}$, $A_{22} = \overline{A}^{i}_{w-out,w-out}$, $A_{31} = \overline{A}^{i}_{z,w-in}$ and $A_{33} = \overline{A}^{i}_{z,z}$. Furthermore, we define the matrix

$$\mathbf{X} = -\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}.$$
 (5.25)

A weighted graph is created with nodes corresponding to each of the aggregated wall temperature states in cluster i ($\overline{\mathbf{T}}_{\mathbf{w}-\mathbf{in}}^{\mathbf{i}}$ and $\overline{\mathbf{T}}_{\mathbf{w}-\mathbf{out}}^{\mathbf{i}}$ in (5.12)), zone temperatures ($\mathbf{T}_{\mathbf{z}}^{\mathbf{j}}$), ambient temperature (T_a), ground temperatures (T_g), and zone temperature estimates ($\widehat{\mathbf{T}}_{\mathbf{z}}^{\mathbf{j}}$) $j \neq i$) in all other clusters. These nodes are numbered in the order in which they are mentioned above. An edge is used to connect each pair of nodes which are connected by a resistance in the reduced order RC network (obtained from Algorithm 5.2). The weight such edges are set to be the inverse of the corresponding thermal resistances. The resulting weighted graph is denoted by $G^i =$ (V^i, E^i) along with the weight function $w: E^i \to \mathbb{R}^+$. Note that this procedure is an extension of the Algorithm 5.1 to a cluster in the building. We also define capacitance matrices $\overline{\mathbf{C}}_{\mathbf{w}-\mathbf{in}}^i, \overline{\mathbf{C}}_{\mathbf{w}-\mathbf{out}}^i$ and $\mathbf{C}_{\mathbf{z}}^i$, which are diagonal matrices of the thermal capacitances associated with the state vectors $\overline{\mathbf{T}}_{\mathbf{w}-\mathbf{out}}^i$, and $\mathbf{T}_{\mathbf{z}}^i$ respectively. The diagonal entries in these matrices are entered in the order of the corresponding node numbers in G^i . Let $\mathbf{L}_{\mathbf{G}}^i$ denote the Laplacian matrix of G^i and recall that n_{w-in}^{i} and n_{w-out}^{i} denote the lengths of the vectors \overline{T}_{w-in}^{i} and \overline{T}_{w-out}^{i} respectively. We now define the following sub-matrices of L_{G}^{i} .

1. L₁₁ is the square sub-matrix of $\mathbf{L}_{\mathbf{G}}^{i}$ which corresponds to its first n_{w-in}^{i} rows and columns.

2. $\mathbf{L}_{\mathbf{X}}$ is the square sub-matrix of $\mathbf{L}_{\mathbf{G}}^{\mathbf{i}}$ which corresponds to its first $n_{w-in}^{\mathbf{i}} + n_{w-out}^{\mathbf{i}}$ rows and columns.

3. **L**₃₁ is the square sub-matrix of $\mathbf{L}_{\mathbf{G}}^{i}$ which corresponds to rows $n_{w-in}^{i} + n_{w-out}^{i} + 1, ..., n_{w-in}^{i} + n_{w-out}^{i} + N_{zi}$ and the first n_{w-in}^{i} columns.

4. L₁₂ is defined as the sub-matrix of L_G^i which corresponds to its first n_{w-in}^i rows and columns $n_{w-in}^i + 1, ..., n_{w-in}^i + n_{w-out}^i$.

An illustration of the above matrix definitions is presented below. Consider a 2-partition $\{\{1,2\},\{3\}\}\)$ of the building in Figure 5.2. The Laplacian matrix L_G^1 for the first cluster $\{1,2\}\)$ in this partition can be expressed as follows using its reduced order RC network representation in Figure 5.3.

$$\mathbf{L}_{\mathbf{G}}^{\mathbf{1}} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & 0 & a_{1,5} & 0 & 0 & 0 & 0 \\ a_{2,1} & a_{2,2} & 0 & a_{2,4} & 0 & a_{2,6} & 0 & 0 & 0 \\ a_{3,1} & 0 & a_{3,3} & 0 & 0 & 0 & a_{3,7} & a_{3,8} & 0 \\ 0 & a_{4,2} & 0 & a_{4,4} & 0 & 0 & a_{4,7} & a_{4,8} & a_{4,9} \\ a_{5,1} & 0 & 0 & 0 & a_{5,5} & 0 & 0 & 0 & 0 \\ 0 & a_{6,2} & 0 & 0 & 0 & a_{6,6} & 0 & 0 & 0 \\ 0 & 0 & a_{7,3} & a_{7,4} & 0 & 0 & a_{7,7} & 0 & 0 \\ 0 & 0 & a_{8,3} & a_{8,4} & 0 & 0 & 0 & a_{8,8} & 0 \\ 0 & 0 & 0 & a_{9,4} & 0 & 0 & 0 & 0 & a_{9,9} \end{bmatrix}$$

where,

$$a_{1,2} = a_{2,1} = \frac{-1}{\bar{R}_{w-in,w-in,1,2}^{1}}$$
$$a_{1,3} = a_{3,1} = \frac{-1}{\bar{R}_{w-in,w-out,1}^{1}}$$
$$a_{1,5} = a_{5,1} = \frac{-1}{\bar{R}_{-1}^{1}}$$

$$a_{1,5} = a_{5,1} = \frac{-1}{\bar{R}_{w-in,z,1,1}^1}$$

$$a_{2,4} = a_{4,2} = \frac{-1}{\bar{R}_{w-in,w-out,2}^{1}}$$

$$a_{2,6} = a_{6,2} = \frac{-1}{\bar{R}_{w-in,z,2,2}^{1}}$$

$$a_{3,7} = a_{7,3} = \frac{-1}{\bar{R}_{w-out,a,1}^{1}}$$

$$a_{3,8} = a_{8,3} = \frac{-1}{\bar{R}_{w-out,a,2}^{1}}$$

$$a_{4,7} = a_{7,4} = \frac{-1}{\bar{R}_{w-out,a,2}^{1}}$$

$$a_{4,8} = a_{8,4} = \frac{-1}{\bar{R}_{w-out,z,2,1}^{1}}$$

$$a_{4,9} = a_{9,4} = \frac{-1}{\bar{R}_{w-out,z,2,1}^{1}}$$

$$a_{1,1} = -a_{1,2} - a_{1,3} - a_{1,5}$$

$$a_{2,2} = -a_{2,1} - a_{2,4} - a_{2,6}$$

$$a_{3,3} = -a_{3,1} - a_{3,7} - a_{3,8}$$

$$a_{4,4} = -a_{4,2} - a_{4,7} - a_{4,8} - a_{4,9}$$

$$a_{5,5} = -a_{5,1}$$

$$a_{6,6} = -a_{6,2}$$

$$a_{7,7} = -a_{7,3} - a_{7,4}$$

$$a_{8,8} = -a_{8,3} - a_{8,4}$$

$$a_{9,9} = -a_{9,4}$$

The matrices $L_{11},\,L_X,\,L_{31}$ and L_{12} are then obtained as

$$\mathbf{L_{11}} = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix},$$

$$\mathbf{L_X} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & 0\\ a_{2,1} & a_{2,2} & 0 & a_{2,4}\\ a_{3,1} & 0 & a_{3,3} & 0\\ 0 & a_{4,2} & 0 & a_{4,4} \end{bmatrix}$$
$$\mathbf{L_{31}} = \begin{bmatrix} a_{5,1} & 0\\ 0 & a_{6,2} \end{bmatrix},$$
$$\mathbf{L_{12}} = \begin{bmatrix} a_{1,3} & 0\\ 0 & a_{2,4} \end{bmatrix}.$$

The following properties can be easily verified from the structure of graph G^i :

1. L_{11} and L_X are Hermitian, irreducible, diagonally dominant matrices with positive diagonal entries. Furthermore, at least one row of both L_{11} and L_X is strictly diagonally dominant.

2. L_{31} is a diagonal matrix with negative entries on diagonal.

3. L_{12} is a sparse matrix of nonpositive elements such that each column has exactly one nonzero entry. Furthermore, the nonzero entries in two different columns of L_{12} are in different row locations.

Now we state and prove the following Lemmas.

Lemma 5.1: A₃₁ is a full rank matrix.

Proof: It can be shown that $A_{31} = -(C_z^i)^{-1}L_{31}$. Using property 2 of L_{31} above, we conclude that A_{31} is a diagonal matrix with positive entries on diagonal. Hence, A_{31} is full rank.

Lemma 5.2: A₁₁ is an invertible matrix.

Proof: $\mathbf{A_{11}}$ can be expressed as $-(\overline{\mathbf{C}_{w-in}^{i}})^{-1}\mathbf{L_{11}}$. Since $\mathbf{L_{11}}$ is an *irreducibly diagonally dominant* matrix from property 1 above, it is non-singular as a result of the *Levy-Desplanques* theorem [91]. Hence, $\mathbf{L_{11}^{-1}}$ exists. Therefore, $\mathbf{A_{11}^{-1}}$ exists and is given by $-\mathbf{L_{11}^{-1}}\overline{\mathbf{C}_{w-in}^{i}}$.

Lemma 5.3: X is a positive definite matrix.

Proof: **X** can be expressed as shown

$$\mathbf{X} = \begin{bmatrix} \bar{\mathbf{C}}_{\mathbf{w}-\mathbf{in}}^{\mathbf{i}} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{C}}_{\mathbf{w}-\mathbf{out}}^{\mathbf{i}} \end{bmatrix}^{-1} \mathbf{L}_{\mathbf{X}}.$$
 (5.26)

Since $\mathbf{L}_{\mathbf{X}}$ is an *irreducibly diagonally dominant* matrix from property 1 above, it is non-singular as a result of the *Levy-Desplanques* theorem [91]. Also, since $\mathbf{L}_{\mathbf{X}}$ is a Hermitian, diagonally dominant matrix with real non-negative entries on the diagonal, it is positive semi-definite. Since a non-singular positive semi-definite matrix can only be positive definite, we conclude that $\mathbf{L}_{\mathbf{X}}$ is positive definite. Using the fact that $\mathbf{\bar{C}}_{\mathbf{w}-\mathbf{in}}^{\mathbf{i}}$ and $\mathbf{\bar{C}}_{\mathbf{w}-\mathbf{out}}^{\mathbf{i}}$ are diagonal matrices with positive entries, we establish using (5.26) that \mathbf{X} is positive definite.

Lemma 5.4: $\mathbf{A}_{12}^{\mathrm{T}} \mathbf{A}_{12}$ is a positive definite matrix.

Proof: A_{12} can be written as $-(\bar{C}_{w-in}^{i})^{-1}L_{12}$. Since \bar{C}_{w-in}^{i} is a diagonal matrix with positive entries, A_{12} share the properties of L_{12} above (property 3), except that its entries are nonnegative instead of nonpositive. Next,

$$(\mathbf{A}_{12}^{\mathrm{T}}\mathbf{A}_{12})_{p,q} = \sum_{r} (\mathbf{A}_{12}^{\mathrm{T}})_{p,r} (\mathbf{A}_{12})_{r,q} = \sum_{r} (\mathbf{A}_{12})_{r,p} (\mathbf{A}_{12})_{r,q}$$

Since $(A_{12})_{r,p}$ and $(A_{12})_{r,q}$ are both nonzero simultaneously for some *r* only if p = q, we conclude that:

$$\left(\mathbf{A}_{12}^{\mathrm{T}} \mathbf{A}_{12} \right)_{p,q} \begin{cases} > 0 & \text{if } p = q \\ = 0 & \text{otherwise.} \end{cases}$$

Hence, $\mathbf{A_{12}^T}\mathbf{A_{12}}$ is a diagonal matrix with positive entries and is therefore positive definite. \Box

We rewrite C_{tfo}^{i} in the form (5.27) which is aligned with the expanded form of \overline{A}_{tfo}^{i} in (5.24).

$$\mathbf{C}_{\mathbf{tfo}}^{i} = \begin{bmatrix} \mathbf{0}_{N_{zi} \times n_{w-in}^{i}} & \mathbf{0}_{N_{zi} \times n_{w-out}^{i}} & \mathbf{I}_{N_{zi}} & \mathbf{0}_{N_{zi}} \end{bmatrix}.$$
(5.27)

The resulting observability matrix for $(\overline{A}^{i}_{tfo}, C^{i}_{tfo})$ is obtained as shown

$$\mathcal{O} = \begin{bmatrix} \mathbf{C}_{tfo}^{l} \\ \mathbf{C}_{tfo}^{i} \overline{A}_{tfo}^{i} \\ \mathbf{C}_{tfo}^{i} (\overline{A}_{tfo}^{i})^{2} \\ \vdots \\ \mathbf{C}_{tfo}^{i} (\overline{A}_{tfo}^{i})^{n_{w-in}^{i} + n_{w-out}^{i} + 2N_{zi}^{i}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{A}_{31} & \mathbf{0} & \mathbf{A}_{33} & \mathbf{I} \\ \mathbf{A}_{31} \mathbf{A}_{11} + \mathbf{A}_{33} \mathbf{A}_{31} & \mathbf{A}_{31} \mathbf{A}_{12} & \mathbf{A}_{31} \mathbf{A}_{13} + (\mathbf{A}_{33})^{2} & \mathbf{A}_{33} \\ \mathbf{G}_{1} & \mathbf{G}_{2} & \mathbf{G}_{3} & \mathbf{G}_{4} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \end{bmatrix}$$
(5.28)

Here, $\mathbf{G}_1 = (\mathbf{A}_{31}\mathbf{A}_{11} + \mathbf{A}_{33}\mathbf{A}_{31})\mathbf{A}_{11} + \mathbf{A}_{31}\mathbf{A}_{12}\mathbf{A}_{21} [\mathbf{A}_{31}\mathbf{A}_{13} + (\mathbf{A}_{33})^2]\mathbf{A}_{31}$,

$$\begin{split} \mathbf{G}_2 &= (\mathbf{A}_{31}\mathbf{A}_{11} + \mathbf{A}_{33}\mathbf{A}_{31})\mathbf{A}_{12} + \mathbf{A}_{31}\mathbf{A}_{12}\mathbf{A}_{22}, \\ \mathbf{G}_3 &= (\mathbf{A}_{31}\mathbf{A}_{11} + \mathbf{A}_{33}\mathbf{A}_{31})\mathbf{A}_{13} + [\mathbf{A}_{31}\mathbf{A}_{13} + (\mathbf{A}_{33})^2]\mathbf{A}_{33}, \\ \mathbf{G}_4 &= \mathbf{A}_{31}\mathbf{A}_{13} + (\mathbf{A}_{33})^2. \end{split}$$

Let us assume that the matrix O does not have full rank. Hence, there exists a vector **v** whose elements are not all zero such that,

$$\boldsymbol{\mathcal{O}}\mathbf{v} = \mathbf{0} \tag{5.29}$$

We rewrite **v** in the form shown in below which is aligned with the expanded form of \boldsymbol{O} in (5.28)

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 \end{bmatrix}^T. \tag{5.30}$$

Using (5.28) and (5.30), (5.29) results in

$$\mathbf{v}_3 = \mathbf{0},\tag{5.31}$$

$$\mathbf{A_{31}v_1} + \mathbf{A_{33}v_3} + \mathbf{v_4} = \mathbf{0}, \tag{5.32}$$

$$(\mathbf{A}_{31}\mathbf{A}_{11} + \mathbf{A}_{33}\mathbf{A}_{31})\mathbf{v}_1 + \mathbf{A}_{31}\mathbf{A}_{12}\mathbf{v}_2 + [\mathbf{A}_{31}\mathbf{A}_{13} + (\mathbf{A}_{33})^2]\mathbf{v}_3 + \mathbf{A}_{33}\mathbf{v}_4 = \mathbf{0}, \quad (5.33)$$

$$\mathbf{G_1}\mathbf{v_1} + \mathbf{G_2}\mathbf{v_2} + \mathbf{G_3}\mathbf{v_3} + \mathbf{G_4}\mathbf{v_4} = \mathbf{0} \tag{5.34}$$

Using (5.31) in (5.32), we obtain

$$A_{31}v_1 + v_4 = 0. (5.35)$$

Using (5.31) and (5.35) in (5.33), we obtain

$$\mathbf{A_{31}}(\mathbf{A_{11}}\mathbf{v_1} + \mathbf{A_{12}}\mathbf{v_2}) = \mathbf{0}.$$
 (5.36)

Since A_{31} is full rank (Lemma 5.1), we obtain

$$\mathbf{A_{11}v_1} + \mathbf{A_{12}v_2} = \mathbf{0}. \tag{5.37}$$

Using (5.31), (5.35) and (5.37) in (5.34), and using the definitions of $\mathbf{G_1}$ to $\mathbf{G_4}$ results in

$$\mathbf{A_{31}A_{12}}[-\mathbf{A_{22}} - \mathbf{A_{21}}(-\mathbf{A_{11}})^{-1}\mathbf{A_{12}}]\mathbf{v_2} = \mathbf{0}.$$
 (5.38)

Note that $(-A_{11})^{-1}$ exists from Lemma 5.2. Since A_{31} is full rank (Lemma 5.1), we obtain

$$\mathbf{A}_{12}[-\mathbf{A}_{22} - \mathbf{A}_{21}(-\mathbf{A}_{11})^{-1}\mathbf{A}_{12}]\mathbf{v}_2 = \mathbf{0}.$$
 (5.39)

We define $\mathbf{S} = -\mathbf{A}_{22} - \mathbf{A}_{21}(-\mathbf{A}_{11})^{-1}\mathbf{A}_{12}$. Note that **S** is the Schur complement of **X**. Since **X** is positive definite (Lemma 5.3), using the Schur complement condition for positive definiteness, we conclude that **S** is positive definite. Next, (5.39) leads to

$$\mathbf{v_2^T} \mathbf{S}^{\mathrm{T}} \mathbf{A_{12}^{\mathrm{T}}} \mathbf{A_{12}} \mathbf{S} \mathbf{v_2} = \mathbf{0}. \tag{5.40}$$

Since **S** and $\mathbf{A}_{12}^{T}\mathbf{A}_{12}$ are positive definite matrices (Lemma 5.4), we conclude from (5.40) that $\mathbf{v}_{2} = \mathbf{0}$. Therefore, from (5.37), $\mathbf{v}_{1} = \mathbf{0}$ since \mathbf{A}_{11} is full rank (Lemma 5.2). Using (5.35), this implies $\mathbf{v}_{4} = \mathbf{0}$. Hence, $\mathbf{v}_{1} = \mathbf{v}_{2} = \mathbf{v}_{3} = \mathbf{v}_{4} = \mathbf{0}$ implying from (5.30) that $\mathbf{v} = \mathbf{0}$. This contradicts the assumption that at least one element of \mathbf{v} is nonzero. Therefore, \mathcal{O} is full rank and hence $(\overline{\mathbf{A}}_{tfo}^{i}, \mathbf{C}_{tfo}^{i})$ is observable.

5.3 Observer and controller design

The reduced order, transformed representation shown in (5.22) provides an observable state space representation of the cluster level dynamics. In this section, we use this property to design an observer at the cluster level to estimate the unknown states. A cluster level controller is then designed which utilizes these estimates to provide optimal control decisions that minimize an appropriate objective function subject to constraints.

5.3.1 Observer design

For any p-partition of a building, a family of discrete-time observers – one for each cluster i – can be designed to estimate the states η_1^i and η_2^i in the model (5.22) using the zone temperature measurements obtained in (5.23). From an implementation perspective, it is desired that the sampling rate of the observers should match that of the controllers. The temperature estimates \widehat{T}_z^j for the zones in the clusters other than *i* appearing in (5.22) are treated as design parameters. For example, they can be set to the corresponding set-point temperatures $T_{z,ref}^j$, based on the assumption that the controllers are able to accurately regulate the zone temperatures around the set-point.

5.3.2 Controller design

Here, we consider the design of output-feedback model predictive controllers which are

decentralized at the cluster-level and use the state estimates provided by the observer in Section 5.3.1. The objective function to be minimized at any time step k for the i^{th} cluster in a p-partition consists of a sum of appropriate performance and cost objectives (see (3.21)).

$$J_i = J_i^{perf} + J_i^{cost}, (5.41)$$

where,

$$J_i^{perf} = \sum_{l=1}^{N_p} [\mathbf{e}_i(k+l)]^T \, diag(\mathbf{\beta}_i) \mathbf{e}_i(k+l), \qquad (5.42)$$

$$\mathbf{e}_{i}(k+l) = \mathbf{T}_{z}^{i}(k+l|k) - \mathbf{T}_{z,ref}^{i}(k), \qquad (5.43)$$

$$J_i^{cost} = \sum_{l=0}^{N_u-1} [\mathbf{u}^{\mathbf{i}}(k+l|k)]^T diag(\mathbf{\alpha}_{\mathbf{i}}) \, \mathbf{u}^{\mathbf{i}}(k+l|k).$$
(5.44)

Here, $\alpha_i, \beta_i \in \mathbb{R}^{N_{zi}}$ are vector valued, component-wise positive weights on the performance (set-point tracking of zone temperatures) and cost (energy consumption) objectives; N_p and N_u represent the prediction and control horizon lengths respectively. A discrete-time version of the model (5.22) has the form

$$\begin{bmatrix} \boldsymbol{\eta}_{1}^{i} \\ \mathbf{T}_{z}^{i} \\ \boldsymbol{\eta}_{2}^{i} \end{bmatrix} (k+1) = \overline{\mathbf{A}}_{tfo,D}^{i} \begin{bmatrix} \boldsymbol{\eta}_{1}^{i} \\ \mathbf{T}_{z}^{i} \\ \boldsymbol{\eta}_{2}^{i} \end{bmatrix} (k) + \overline{\mathbf{B}}_{u,tfo,D}^{i} \mathbf{u}^{i}(k) + \begin{bmatrix} \overline{\mathbf{B}}_{a,tfo,D}^{i} & \overline{\mathbf{B}}_{g,tfo,D}^{i} \end{bmatrix} \begin{bmatrix} T_{a} \\ T_{g} \end{bmatrix} (k) + \sum_{j \neq i} \overline{\mathbf{B}}_{z,tfo,D}^{ij} \widehat{\mathbf{T}}_{z}^{j}(k).$$

$$(5.45)$$

The model (5.45) imposes the following constraints on the optimization:

1. For each $l \in \{1, 2, ..., N_u\}$,

$$\begin{bmatrix} \boldsymbol{\eta}_{1}^{i} \\ \mathbf{T}_{z}^{i} \\ \boldsymbol{\eta}_{2}^{i} \end{bmatrix} (k+l|k) = \overline{\mathbf{A}}_{tfo,D}^{i} \begin{bmatrix} \boldsymbol{\eta}_{1}^{i} \\ \mathbf{T}_{z}^{i} \\ \boldsymbol{\eta}_{2}^{i} \end{bmatrix} (k+l-1|k) + \overline{\mathbf{B}}_{u,tfo,D}^{i} \mathbf{u}^{i}(k+l-1|k) + \left[\overline{\mathbf{B}}_{a,tfo,D}^{i} - \overline{\mathbf{B}}_{g,tfo,D}^{i} \right] \begin{bmatrix} T_{a} \\ T_{g} \end{bmatrix} (k) + \sum_{j \neq i} \overline{\mathbf{B}}_{z,tfo,D}^{ij} \widehat{\mathbf{T}}_{z}^{j}(k), \quad (5.46)$$

2. For each
$$l \in \{N_u + 1, ..., N_u + N_p\}$$
,

$$\begin{bmatrix} \mathbf{\eta}_1^i \\ \mathbf{T}_z^i \\ \mathbf{\eta}_2^i \end{bmatrix} (k+l|k) = \overline{\mathbf{A}}_{\mathsf{tfo},\mathsf{D}}^i \begin{bmatrix} \mathbf{\eta}_1^i \\ \mathbf{T}_z^i \\ \mathbf{\eta}_2^i \end{bmatrix} (k+l-1|k) + \overline{\mathbf{B}}_{\mathsf{u},\mathsf{tfo},\mathsf{D}}^i \mathbf{u}^i (k+N_u-1|k)$$

+
$$\begin{bmatrix} \overline{\mathbf{B}}_{a,\text{tfo},\mathbf{D}}^{i} & \overline{\mathbf{B}}_{g,\text{tfo},\mathbf{D}}^{i} \end{bmatrix} \begin{bmatrix} T_{a} \\ T_{g} \end{bmatrix} (k) + \sum_{j \neq i} \overline{\mathbf{B}}_{z,\text{tfo},\mathbf{D}}^{ij} \widehat{\mathbf{T}}_{z}^{j}(k).$$
 (5.47)

Additional constraints, such as heating and cooling capacity bounds are represented by

$$\mathbf{u}^{i}(k+l-1|k) \in \mathcal{U}^{i}$$
, for all $l \in \{1, 2, ..., N_{u}\}.$ (5.48)

Here, \mathcal{U}^i represents the set of feasible values of $\mathbf{u}^i(k+l-1|k)$. At each time instant k, the controller uses: (a) the state estimates from the observer $\hat{\mathbf{\eta}}_1^i(k)$ and $\hat{\mathbf{\eta}}_2^i(k)$, (b) the zone temperature measurements from thermostats $\mathbf{T}_{z,meas}^i(k)$, and (c) the signals $T_a(k)$, $T_g(k)$ and $\{\hat{\mathbf{T}}_z^j(k)\}_{j\neq i}$ to determine optimal values $\{\mathbf{u}_*^i(k+l)\}_{l=0}^{N_u-1}$ of the control inputs $\{\mathbf{u}^i(k+l-1|k)\}_{l=1}^{N_u-1}$ which minimize the objective function (5.41) subject to the constraints (5.46) – (5.48). In accordance with the MPC methodology, the control input $\mathbf{u}_*^i(k)$ corresponding to the current time instant is then applied to the plant. The observer and controller for the *i*th cluster are illustrated in Figure 5.4.



Figure 5.4 Illustration of controller and observer for i^{th} cluster.

It should be noted that the design of cluster level decentralized controllers and observers discussed in this section leads to an output feedback control framework which only uses zone temperature measurements that are provided by thermostats. It also incorporates physical constraints such as heating or cooling capacity bounds associated with the HVAC system. Therefore, it builds upon the decentralized control framework that was presented in chapter 3 (Section 3.4) to make it practically implementable.

5.4 Optimization

Through a procedure analogous to that presented in Section 3.4.4, the optimization of the cost function J_i in (5.41) for the i^{th} control agent, in the presence of heating/cooling capacity constraints can be converted to a Quadratic Program (QP). The underlying procedure is shown in this section.

5.4.1 Re-statement of objective function

We augment the state space model in (5.45) by defining a new state x_u^i as shown below

$$\mathbf{x}_{\mathbf{u}}^{\mathbf{i}}(k) = \mathbf{u}^{\mathbf{i}}(k-1).$$
 (5.49)

Therefore,

$$\mathbf{x}_{\mathbf{u}}^{\mathbf{i}}(k+1) = \mathbf{x}_{\mathbf{u}}^{\mathbf{i}}(k) + \Delta \mathbf{u}^{\mathbf{i}}(k), \qquad (5.50)$$

Where $\Delta u^{i}(k)$ is defined as

$$\Delta \mathbf{u}^{\mathbf{i}}(k) = \mathbf{u}^{\mathbf{i}}(k) - \mathbf{u}^{\mathbf{i}}(k-1) . \qquad (5.51)$$

Using (5.50) and (5.51), the augmented form of (5.45) is given by

$$\mathbf{x}_{mpc}^{i}(k+1) = \mathbf{A}_{mpc}^{i}\mathbf{x}_{mpc}^{i}(k) + \mathbf{B}_{u,mpc}^{i}\Delta\mathbf{u}^{i}(k) + \mathbf{B}_{a,mpc}^{i}T_{a}(k) + \mathbf{B}_{g,mpc}^{i}T_{g}(k) + \sum_{j\neq i} \overline{\mathbf{B}}_{mpc}^{ij} \widehat{\mathbf{T}}_{z}^{j}(k), \qquad (5.52)$$

where,

$$\mathbf{x}_{mpc}^{i} = \begin{bmatrix} \eta_{1}^{i} \\ \mathbf{T}_{z}^{i} \\ \eta_{2}^{i} \\ \mathbf{x}_{u}^{i} \end{bmatrix}, \mathbf{A}_{mpc}^{i} = \begin{bmatrix} \overline{\mathbf{A}}_{tfo,D}^{i} & \overline{\mathbf{B}}_{u,tfo,D}^{i} \\ \mathbf{0} & \mathbf{I}_{N_{zi}} \end{bmatrix}, \mathbf{B}_{u,mpc}^{i} = \begin{bmatrix} \overline{\mathbf{B}}_{u,tfo,D}^{i} \\ \mathbf{I}_{N_{zi}} \end{bmatrix}, \mathbf{B}_{a,mpc}^{i} = \begin{bmatrix} \overline{\mathbf{B}}_{a,tfo,D}^{i} \\ \mathbf{0}_{N_{zi} \times 1} \end{bmatrix},$$

$$\mathbf{B}_{g,mpc}^{i} = \begin{bmatrix} \overline{\mathbf{B}}_{g,tfo,D}^{i} \\ \mathbf{0}_{N_{zi} \times 1} \end{bmatrix}, \ \overline{\mathbf{B}}_{mpc}^{ij} = \begin{bmatrix} \overline{\mathbf{B}}_{z,tfo,D}^{ij} \\ \mathbf{0}_{N_{zi} \times N_{zj}} \end{bmatrix}.$$

Based on (5.52), the constraints (5.46) and (5.47) can be written as the following.

- 1. For each $l \in \{1, 2, ..., N_u\}$: $\mathbf{x_{mpc}^{i}}(k+l|k) = \mathbf{A_{mpc}^{i}}\mathbf{x_{mpc}^{i}}(k+l-1|k) + \mathbf{B_{u,mpc}^{i}}\Delta \mathbf{u}^{i}(k+l-1|k)$ $+ \mathbf{B_{a,mpc}^{i}}T_{a}(k+l-1|k) + \mathbf{B_{g,mpc}^{i}}T_{g}(k+l-1|k)$ $+ \sum_{j \neq i} \overline{\mathbf{B}_{mpc}^{ij}} \widehat{\mathbf{T}_{z}^{j}}(k+l-1|k),$ (5.53)
- 2. For each $l \in \{N_u + 1, 2, ..., N_p\}$:

$$\mathbf{x}_{mpc}^{\mathbf{i}}(k+l|k) = \mathbf{A}_{mpc}^{\mathbf{i}}\mathbf{x}_{mpc}^{\mathbf{i}}(k+l-1|k) + \mathbf{B}_{a,mpc}^{\mathbf{i}}T_{a}(k+l-1|k) + \mathbf{B}_{g,mpc}^{\mathbf{i}}T_{g}(k+l-1|k) + \sum_{j\neq i} \overline{\mathbf{B}}_{mpc}^{\mathbf{ij}} \widehat{\mathbf{T}}_{\mathbf{z}}^{\mathbf{j}}(k+l-1|k), \qquad (5.54)$$

The derivation of (5.54) uses the fact that from definition (5.51), $\Delta \mathbf{u}^{\mathbf{i}}(k+l-1|k) = 0$ outside the control horizon, i.e. when $l \in \{N_u + 1, 2, ..., N_p\}$. Using (5.53) and (5.54), we obtain the following equations via iterations for each time step in the prediction horizon.

$$\mathbf{x}_{mpc}^{i}(k+1|k) = \mathbf{A}_{mpc}^{i}\mathbf{x}_{mpc}^{i}(k) + \mathbf{B}_{u,mpc}^{i}\Delta\mathbf{u}^{i}(k) + \mathbf{B}_{a,mpc}^{i}T_{a}(k) + \mathbf{B}_{g,mpc}^{i}T_{g}(k)$$
$$+ \sum_{j\neq i} \overline{\mathbf{B}}_{mpc}^{ij} \widehat{\mathbf{T}}_{z}^{j}(k).$$

 $\mathbf{x}_{mpc}^{i}(k+2|k) = \mathbf{A}_{mpc}^{i}\mathbf{x}_{mpc}^{i}(k+1|k) + \mathbf{B}_{u,mpc}^{i}\Delta\mathbf{u}^{i}(k+1|k) + \mathbf{B}_{a,mpc}^{i}T_{a}(k+1|k)$

$$+\mathbf{B}_{g,mpc}^{i}T_{g}(k+1|k) + \sum_{j\neq i} \overline{\mathbf{B}}_{mpc}^{ij} \widehat{\mathbf{T}}_{z}^{j}(k+1|k)$$

$$= \left(\mathbf{A}_{mpc}^{i}\right)^{2} \mathbf{x}_{mpc}^{i}(k) + \mathbf{A}_{mpc}^{i} \mathbf{B}_{u,mpc}^{i} \Delta \mathbf{u}^{i}(k) + \mathbf{B}_{u,mpc}^{i} \Delta \mathbf{u}^{i}(k+1|k)$$

$$+ \mathbf{A}_{mpc}^{i} \mathbf{B}_{a,mpc}^{i} T_{a}(k) + \mathbf{B}_{a,mpc}^{i} T_{a}(k+1|k) + \mathbf{A}_{mpc}^{i} \mathbf{B}_{g,mpc}^{i} T_{g}(k)$$

$$+ \mathbf{B}_{g,mpc}^{i} T_{g}(k+1|k) + \sum_{j\neq i} \left(\mathbf{A}_{mpc}^{i} \overline{\mathbf{B}}_{mpc}^{ij} \widehat{\mathbf{T}}_{z}^{j}(k) + \overline{\mathbf{B}}_{mpc}^{ij} \widehat{\mathbf{T}}_{z}^{j}(k+1|k)\right)$$

÷

$$\mathbf{x}_{mpc}^{\mathbf{i}}(k+N_{u}|k) = \left(\mathbf{A}_{mpc}^{\mathbf{i}}\right)^{N_{u}} \mathbf{x}_{mpc}^{\mathbf{i}}(k) + \sum_{m=0}^{N_{u}-1} \left(\mathbf{A}_{mpc}^{\mathbf{i}}\right)^{N_{u}-1-m} \left(\mathbf{B}_{u,mpc}^{\mathbf{i}} \Delta \mathbf{u}^{\mathbf{i}}(k+m|k)\right)$$
$$+ \mathbf{B}_{a,mpc}^{\mathbf{i}} T_{a}(k+m|k) + \mathbf{B}_{g,mpc}^{\mathbf{i}} T_{g}(k+m|k)\right) + \sum_{m=0}^{N_{u}-1} \sum_{j\neq i} \left(\mathbf{A}_{mpc}^{\mathbf{i}}\right)^{N_{u}-1-m} \left(\overline{\mathbf{B}}_{mpc}^{\mathbf{i}} \widehat{\mathbf{T}}_{z}^{\mathbf{j}}(k+m|k)\right)$$

$$\mathbf{x_{mpc}^{i}}(k+N_{u}+1|k) = \left(\mathbf{A_{mpc}^{i}}\right)^{N_{u}+1} \mathbf{x_{mpc}^{i}}(k) + \sum_{m=0}^{N_{u}-1} \left(\mathbf{A_{mpc}^{i}}\right)^{N_{u}-m} \mathbf{B_{u,mpc}^{i}} \Delta \mathbf{u}^{i}(k+m|k)$$
$$+ \sum_{m=0}^{N_{u}} \left(\mathbf{A_{mpc}^{i}}\right)^{N_{u}-m} \left(\mathbf{B_{a,mpc}^{i}}T_{a}(k+m|k) + \mathbf{B_{g,mpc}^{i}}T_{g}(k+m|k)\right)$$
$$+ \sum_{m=0}^{N_{u}} \sum_{j \neq i} \left(\mathbf{A_{mpc}^{i}}\right)^{N_{u}-m} \left(\mathbf{\overline{B}_{mpc}^{ij}} \mathbf{\widehat{T}_{z}^{j}}(k+m|k)\right)$$
$$\vdots$$

$$\mathbf{x}_{mpc}^{\mathbf{i}}(k+N_p|k) = \left(\mathbf{A}_{mpc}^{\mathbf{i}}\right)^{N_p} \mathbf{x}_{mpc}^{\mathbf{i}}(k) + \sum_{m=0}^{N_u-1} \left(\mathbf{A}_{mpc}^{\mathbf{i}}\right)^{N_p-1} \mathbf{B}_{\mathbf{u},mpc}^{\mathbf{i}} \Delta \mathbf{u}^{\mathbf{i}}(k+m|k)$$
$$+ \sum_{m=0}^{N_p-1} \left(\mathbf{A}_{mpc}^{\mathbf{i}}\right)^{N_p-1} \left(\mathbf{B}_{\mathbf{a},mpc}^{\mathbf{i}} T_a(k+m|k) + \mathbf{B}_{\mathbf{g},mpc}^{\mathbf{i}} T_g(k+m|k)\right)$$
$$+ \sum_{m=0}^{N_p-1} \sum_{j\neq i} \left(\mathbf{A}_{mpc}^{\mathbf{i}}\right)^{N_p-1} \left(\overline{\mathbf{B}}_{mpc}^{\mathbf{ij}} \widehat{\mathbf{T}}_{\mathbf{z}}^{\mathbf{j}}(k+m|k)\right)$$

The above equations can be succinctly written as

$$\bar{\mathbf{x}}_{mpc}^{i} = \mathbf{M}_{i} \mathbf{x}_{mpc}^{i}(k) + \mathbf{S}_{ui} \overline{\Delta u}^{i} + \mathbf{S}_{ai} \overline{\mathbf{T}}_{a} + \mathbf{S}_{gi} \overline{\mathbf{T}}_{g} + \sum_{j \neq i} \mathbf{S}_{ij} \overline{\mathbf{T}}_{z}^{j}.$$
(5.55)

Here,

$$\bar{\mathbf{x}}_{mpc}^{i} = \begin{bmatrix} \mathbf{x}_{mpc}^{i}(k+1|k) \\ \mathbf{x}_{mpc}^{i}(k+2|k) \\ \vdots \\ \mathbf{x}_{mpc}^{i}(k+N_{p}|k) \end{bmatrix}, \quad \overline{\Delta \mathbf{u}}^{i} = \begin{bmatrix} \Delta \mathbf{u}^{i}(k|k) \\ \Delta \mathbf{u}^{i}(k+1|k) \\ \vdots \\ \Delta \mathbf{u}^{i}(k+N_{u}-1|k) \end{bmatrix}, \quad \overline{\mathbf{T}}_{a} = \begin{bmatrix} T_{a}(k) \\ T_{a}(k+1|k) \\ \vdots \\ T_{a}(k+N_{p}-1|k) \end{bmatrix},$$

$$\overline{\mathbf{T}}_{\mathbf{g}} = \begin{bmatrix} T_g(k) \\ T_g(k+1|k) \\ \vdots \\ T_g(k+N_p-1|k) \end{bmatrix}, \ \overline{\widehat{\mathbf{T}}}_{\mathbf{z}}^{\mathbf{j}} = \begin{bmatrix} \widehat{\mathbf{T}}_{\mathbf{z}}^{\mathbf{j}}(k) \\ \widehat{\mathbf{T}}_{\mathbf{z}}^{\mathbf{j}}(k+1|k) \\ \vdots \\ \widehat{\mathbf{T}}_{\mathbf{z}}^{\mathbf{j}}(k+N_p-1|k) \end{bmatrix}, \ \mathbf{M}_{\mathbf{i}} = \begin{bmatrix} \mathbf{A}_{\mathbf{mpc}}^{\mathbf{i}} \\ \left(\mathbf{A}_{\mathbf{mpc}}^{\mathbf{i}}\right)^2 \\ \vdots \\ \left(\mathbf{A}_{\mathbf{mpc}}^{\mathbf{i}}\right)^{N_p} \end{bmatrix},$$

$$S_{ui} = \begin{bmatrix} B_{u,mpc}^{i} & 0 & ... & 0 \\ A_{mpc}^{i}B_{u,mpc}^{i} & \ddots & \ddots & \vdots \\ \vdots & \ddots & B_{u,mpc}^{i} & 0 \\ (A_{mpc}^{i})^{N_{u}-1}B_{u,mpc}^{i} & ... & A_{mpc}^{i}B_{u,mpc}^{i} & B_{u,mpc}^{i} \\ (A_{mpc}^{i})^{N_{u}}B_{u,mpc}^{i} & ... & (A_{mpc}^{i})^{2}B_{u,mpc}^{i} & A_{mpc}^{i}B_{u,mpc}^{i} \\ \vdots & \ddots & \vdots & \vdots \\ (A_{mpc}^{i})^{N_{p}-1}B_{u,mpc}^{i} & ... & (A_{mpc}^{i})^{N_{p}-N_{u}-1}B_{u,mpc}^{i} & (A_{mpc}^{i})^{N_{p}-N_{u}}B_{u,mpc}^{i} \end{bmatrix}$$

$$\mathbf{S}_{ai} = \begin{bmatrix} \mathbf{B}_{a,mpc}^{i} & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{A}_{mpc}^{i} \mathbf{B}_{a,mpc}^{i} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_{a,mpc}^{i} & \mathbf{0} \\ \left(\mathbf{A}_{mpc}^{i}\right)^{N_{p}-1} \mathbf{B}_{a,mpc}^{i} & ... & \mathbf{A}_{i} \mathbf{B}_{a,mpc}^{i} & \mathbf{B}_{a,mpc}^{i} \end{bmatrix},$$

$$\mathbf{S_{gi}} = \begin{bmatrix} \mathbf{B_{g,mpc}^{i}} & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{A_{mpc}^{i} B_{g,mpc}^{i}} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B_{g,mpc}^{i}} & \mathbf{0} \\ \left(\mathbf{A_{mpc}^{i}}\right)^{N_{p}-1} \mathbf{B_{g,mpc}^{i}} & ... & \mathbf{A_{i} B_{g,mpc}^{i}} & \mathbf{B_{g,mpc}^{i}} \end{bmatrix},$$

$$\mathbf{S_{ij}} = \begin{bmatrix} \mathbf{\overline{B}_{mpc}^{ij}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A_{mpc}^{i} \overline{B}_{mpc}^{ij}} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{\overline{B}_{mpc}^{ij}} & \mathbf{0} \\ (\mathbf{A_{mpc}^{i}})^{N_p - 1} \mathbf{\overline{B}_{mpc}^{ij}} & \dots & \mathbf{A_i} \mathbf{\overline{B}_{mpc}^{ij}} & \mathbf{\overline{B}_{mpc}^{ij}} \end{bmatrix}.$$

From (5.41), J_i can be re-stated as

$$J_{i} = \left(\overline{\mathbf{T}}_{z}^{i} - \overline{\mathbf{T}}_{z,ref}^{i}\right)^{T} \mathbf{Q}_{1,i} \left(\overline{\mathbf{T}}_{z}^{i} - \overline{\mathbf{T}}_{z,ref}^{i}\right) + \overline{\mathbf{u}}^{i^{T}} \mathbf{Q}_{2,i} \overline{\mathbf{u}}^{i}$$
$$= \left(\begin{bmatrix} \overline{\mathbf{T}}_{z}^{i} \\ \overline{\mathbf{u}}^{i} \end{bmatrix} - \begin{bmatrix} \overline{\mathbf{T}}_{z,ref}^{i} \\ \mathbf{0} \end{bmatrix} \right)^{T} \begin{bmatrix} \mathbf{Q}_{1,i} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{2,i} \end{bmatrix} \left(\begin{bmatrix} \overline{\mathbf{T}}_{z}^{i} \\ \overline{\mathbf{u}}^{i} \end{bmatrix} - \begin{bmatrix} \overline{\mathbf{T}}_{z,ref}^{i} \\ \mathbf{0} \end{bmatrix} \right), \quad (5.56)$$

where,

$$\begin{split} \mathbf{\bar{T}}_{z}^{i} &= \begin{bmatrix} \mathbf{T}_{z}^{i}(k+1|k) \\ \mathbf{T}_{z}^{i}(k+2|k) \\ \vdots \\ \mathbf{T}_{z}^{i}(k+N_{p}|k) \end{bmatrix}, \mathbf{\bar{T}}_{z,ref} = \begin{bmatrix} \mathbf{T}_{z,ref}^{i}(k) \\ \mathbf{T}_{z,ref}^{i}(k) \\ \vdots \\ \mathbf{T}_{z,ref}^{i}(k) \end{bmatrix}_{(N_{p},N_{z}i)\times 1}^{,} \\ \mathbf{Q}_{1,i} &= \begin{bmatrix} diag(\mathbf{\beta}_{i}) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & diag(\mathbf{\beta}_{i}) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & diag(\mathbf{\beta}_{i}) \end{bmatrix}_{(N_{zi},N_{p})\times(N_{zi},N_{p})}^{,} \\ \text{and } \mathbf{Q}_{2,i} &= \begin{bmatrix} diag(\mathbf{\alpha}_{i}) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & diag(\mathbf{\alpha}_{i}) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & diag(\mathbf{\alpha}_{i}) \end{bmatrix}_{(N_{zi},N_{u})\times(N_{zi},N_{u})}^{,} \end{split}$$

Let
$$\mathbf{C}_{\mathbf{mpc}}^{\mathbf{i}} = \begin{bmatrix} \mathbf{0}_{N_{zi} \times N_{x}} & \mathbf{I}_{N_{zi}} & \mathbf{0}_{N_{zi} \times N_{zi}} & \mathbf{0}_{N_{zi} \times N_{zi}} \\ \mathbf{0}_{N_{zi} \times N_{x}} & \mathbf{0}_{N_{zi} \times N_{zi}} & \mathbf{0}_{N_{zi} \times N_{zi}} & \mathbf{I}_{N_{zi}} \end{bmatrix}$$
,
and $\bar{\mathbf{C}}_{\mathbf{mpc}}^{\mathbf{i}} = \begin{bmatrix} \mathbf{C}_{\mathbf{mpc}}^{\mathbf{i}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{mpc}}^{\mathbf{i}} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{\mathbf{mpc}}^{\mathbf{i}} \end{bmatrix}_{(2N_{zi} \cdot N_{p}) \times ((N_{x} + 3N_{zi}) \cdot N_{p})}$.

Here, $N_x = n_{w-in}^i + n_{w-out}^i$.

Clearly,

$$\begin{bmatrix} \overline{\mathbf{T}}_{\mathbf{z}}^{\mathbf{i}} \\ \overline{\mathbf{u}}^{\mathbf{i}} \end{bmatrix} = \overline{\mathbf{C}}_{\mathbf{mpc}}^{\mathbf{i}} \overline{\mathbf{x}}_{\mathbf{mpc}}^{\mathbf{i}}.$$
 (5.57)

Using 5.55), (5.57) can be written as

$$\begin{bmatrix} \overline{\mathbf{T}}_{\mathbf{z}}^{i} \\ \overline{\mathbf{u}}^{i} \end{bmatrix} = \overline{\mathbf{C}}_{\mathrm{mpc}}^{i} \Big(\mathbf{M}_{i} \mathbf{x}_{\mathrm{mpc}}^{i}(k) + \mathbf{S}_{\mathrm{ui}} \overline{\Delta \mathbf{u}}^{i} + \mathbf{S}_{\mathrm{ai}} \overline{\mathbf{T}}_{\mathrm{a}} + \mathbf{S}_{\mathrm{gi}} \overline{\mathbf{T}}_{\mathrm{g}} + \sum_{j \neq i} \mathbf{S}_{ij} \overline{\mathbf{T}}_{\mathrm{g}}^{j} \Big).$$
(5.58)

We define the following:

$$\overline{\mathbf{T}}_{\text{ref}}^{i} = \begin{pmatrix} \overline{\mathbf{T}}_{z,\text{ref}}^{i} \\ \mathbf{0}_{(N_{u},N_{zi})\times 1} \end{pmatrix}, \mathbf{Q}_{i} = \begin{bmatrix} \mathbf{Q}_{1,i} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{2,i} \end{bmatrix}.$$
 (5.59)

Using (5.58) and (5.59) in (5.56), and ignoring terms which are independent of $\overline{\Delta u}^i$ (the optimization variable), we obtain

$$J_i = (\overline{\Delta \mathbf{u}}^i)^T \mathbf{H}^i_{\mathbf{mpc}} \overline{\Delta \mathbf{u}}^i + (\mathbf{f}^i_{\mathbf{mpc}})^T \overline{\Delta \mathbf{u}}^i, \qquad (5.60)$$

where,

$$\mathbf{H}_{\mathbf{mpc}}^{\mathbf{i}} = \mathbf{S}_{\mathbf{ui}}^{T} \left(\bar{\mathbf{C}}_{\mathbf{mpc}}^{\mathbf{i}} \right)^{T} \mathbf{Q}_{\mathbf{i}} \bar{\mathbf{C}}_{\mathbf{mpc}}^{\mathbf{i}} \mathbf{S}_{\mathbf{ui}}^{T}, \qquad (5.61)$$

and,

$$\mathbf{f}_{dc,i} = 2\mathbf{S}_{ui}^{T} \left(\overline{\mathbf{C}}_{mpc}^{i} \right)^{T} \mathbf{Q}_{i} \left[\overline{\mathbf{C}}_{mpc}^{i} \left(\mathbf{M}_{i} \mathbf{x}_{mpc}^{i}(k) + \mathbf{S}_{ui} \overline{\Delta u}^{i} + \mathbf{S}_{ai} \overline{\mathbf{T}}_{a} + \mathbf{S}_{gi} \overline{\mathbf{T}}_{g} + \sum_{j \neq i} \mathbf{S}_{ij} \overline{\mathbf{T}}_{z}^{j} \right) - \overline{\mathbf{T}}_{ref}^{i} \right].$$

$$(5.62)$$

In the above framework, $\mathbf{x}_{mpc}^{i}(k)$ is constructed from the estimated and measured states. Therefore,

$$\mathbf{x}_{\mathrm{mpc}}^{i}(k) = \begin{bmatrix} \widehat{\eta}_{1}^{i}(k) \\ \mathbf{T}_{z,\mathrm{meas}}^{i}(k) \\ \widehat{\eta}_{2}^{i}(k) \end{bmatrix}.$$
(5.63)

5.4.2 Constraints

In this section, we consider constraints representing upper bounds on the heating and cooling provided to the zones by the HVAC system. These constraints translate into upper and lower bound inequality constraints on the control inputs. We assume that the HVAC system is a VAV system with reheating coils described in Chapter 2. For simplicity, we assume 100% recirculation of return air from each zone, i.e. there is no mixing of outside air in the air handling unit. Furthermore, we assume that the temperature of cold supply air is fixed; therefore the heating or cooling provided to the zones is manipulated only by changing the air mass flow rates via dampers in the VAV boxes and/or by varying the amount of heating provided by reheat coils (in case of heating only). In the remainder of this section, ' \leq ' and ' \geq ' when used to compare vectors, refer to inequalities taken component-wise.

5.4.2.1 Cooling constraints

For each zone $n \in \{1, 2, ..., N_{zi}\}$ in the *i*th cluster, the cooling constraint due to the above assumptions is given by (refer to nomenclature):

$$u_n^i(k) \ge \dot{m}_{max,n}^i c_{pa} (T_{supp} - T_{z,n}^i(k))$$
 (5.64)

The combined form of (5.64) for all zones in the cluster is given by

$$\mathbf{u}^{\mathbf{i}}(k) \ge \mathbf{M}^{\mathbf{i}}_{\max} c_{pa} \big(\mathbf{T}_{\mathrm{supp}} - \mathbf{T}^{\mathbf{i}}_{\mathbf{z}}(k) \big)$$
(5.65)

Here,

$$\mathbf{M_{max}^{i}} = \begin{bmatrix} \dot{m}_{max,1}^{i} & 0 & \dots & 0 \\ 0 & \dot{m}_{max,2}^{i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dot{m}_{max,N_{zi}}^{i} \end{bmatrix}_{N_{zi} \times N_{zi}},$$

$$\mathbf{T_{supp}} = \begin{bmatrix} T_{supp} \\ T_{supp} \\ \vdots \\ T_{supp} \end{bmatrix}_{N_{zi} \times 1}.$$

Equation (5.65) results in the following constraints over the control and prediction horizons.

1. For
$$l \in \{1, 2, ..., N_u - 1\}$$
,
 $\mathbf{u}^{\mathbf{i}}(k-1) + \Delta \mathbf{u}^{\mathbf{i}}(k) + \Delta \mathbf{u}^{\mathbf{i}}(k+1|k) + \dots + \Delta \mathbf{u}^{\mathbf{i}}(k+l|k)$
 $\geq \mathbf{M}^{\mathbf{i}}_{\max} c_{pa} \left(\mathbf{T}_{\sup p} - \mathbf{T}^{\mathbf{i}}_{\mathbf{z}}(k+l|k) \right).$ (5.66)

2. For
$$l \in \{N_u, N_u + 1, ..., N_p\}$$
,
 $\mathbf{u}^{\mathbf{i}}(k-1) + \Delta \mathbf{u}^{\mathbf{i}}(k) + \Delta \mathbf{u}^{\mathbf{i}}(k+1|k) + \dots + \Delta \mathbf{u}^{\mathbf{i}}(k+N_u-1|k)$
 $\geq \mathbf{M}^{\mathbf{i}}_{\max} c_{pa} \left(\mathbf{T}_{\mathrm{supp}} - \mathbf{T}^{\mathbf{i}}_{\mathbf{z}}(k+l|k) \right).$ (5.67)

Equations (5.66) and (5.67) can be succinctly written as

$$\mathbf{A}_{1}^{i} \overline{\Delta \mathbf{u}}^{i} \ge c_{pa} \overline{\mathbf{M}}_{\max}^{i} \left(\mathbf{A}_{2}^{i} \mathbf{T}_{\sup p} - \begin{bmatrix} \mathbf{T}_{z}^{i}(k) \\ \overline{\mathbf{T}}_{z}^{i} \end{bmatrix} \right) - \mathbf{A}_{2}^{i} \mathbf{u}^{i}(k-1), \qquad (5.68)$$

where,

$$\mathbf{A_{1}^{i}} = \begin{bmatrix} \mathbf{A_{1,1}^{i}} \\ \mathbf{A_{1,2}^{i}} \end{bmatrix}, \mathbf{A_{1,1}^{i}} = \begin{bmatrix} \mathbf{I}_{N_{zi}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I}_{N_{zi}} & \mathbf{I}_{N_{zi}} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{N_{zi}} & \mathbf{I}_{N_{zi}} & \dots & \mathbf{I}_{N_{zi}} \end{bmatrix}_{(N_{zi}.N_{u}) \times (N_{zi}.N_{u})},$$

$$\mathbf{A_{1,2}^{i}} = \begin{bmatrix} \mathbf{I}_{N_{zi}} & \mathbf{I}_{N_{zi}} & \dots & \mathbf{I}_{N_{zi}} \\ \mathbf{I}_{N_{zi}} & \mathbf{I}_{N_{zi}} & \dots & \mathbf{I}_{N_{zi}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{N_{zi}} & \mathbf{I}_{N_{zi}} & \dots & \mathbf{I}_{N_{zi}} \end{bmatrix}_{\begin{pmatrix} N_{zi} \cdot (N_{p} - N_{u} + 1) \end{pmatrix} \times (N_{zi} \cdot N_{u})}, \mathbf{A_{2}^{i}} = \begin{bmatrix} \mathbf{I}_{N_{zi}} \\ \mathbf{I}_{N_{zi}} \\ \vdots \\ \mathbf{I}_{N_{zi}} \end{bmatrix}_{\begin{pmatrix} N_{zi} \cdot (N_{p} + 1) \end{pmatrix} \times N_{zi}},$$

$$\overline{\mathbf{M}}_{\max}^{i} = \begin{bmatrix} \mathbf{M}_{\max}^{i} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\max}^{i} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{M}_{\max}^{i} \end{bmatrix}_{\left(N_{zi} \cdot (N_{p}+1)\right) \times \left(N_{zi} \cdot (N_{p}+1)\right)}$$

 \overline{T}^{i}_{z} and $\overline{\Delta u}^{i}$ were defined earlier as

$$\overline{\mathbf{T}}_{\mathbf{z}}^{\mathbf{i}} = \begin{bmatrix} \mathbf{T}_{\mathbf{z}}^{\mathbf{i}}(k+1|k) \\ \mathbf{T}_{\mathbf{z}}^{\mathbf{i}}(k+2|k) \\ \vdots \\ \mathbf{T}_{\mathbf{z}}^{\mathbf{i}}(k+N_{p}|k) \end{bmatrix}, \ \overline{\mathbf{\Delta u}}^{\mathbf{i}} = \begin{bmatrix} \mathbf{\Delta u}^{\mathbf{i}}(k|k) \\ \mathbf{\Delta u}^{\mathbf{i}}(k+1|k) \\ \vdots \\ \mathbf{\Delta u}^{\mathbf{i}}(k+N_{u}-1|k) \end{bmatrix}.$$

Let $C^i_{T_z} \text{and } \bar{C}^i_{T_z} \text{be defined as}$

$$\mathbf{C}_{\mathbf{T}_{z}}^{i} = \begin{bmatrix} \mathbf{0}_{N_{zi} \times N_{x}} & \mathbf{I}_{N_{zi}} & \mathbf{0}_{N_{zi} \times N_{zi}} & \mathbf{0}_{N_{zi} \times N_{zi}} \end{bmatrix},$$

$$\mathbf{\bar{C}}_{\mathbf{T}_{z}}^{i} = \begin{bmatrix} \mathbf{C}_{\mathbf{T}_{z}}^{i} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{T}_{z}}^{i} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{\mathbf{T}_{z}}^{i} \end{bmatrix}_{(N_{zi} \cdot N_{p}) \times ((N_{x} + 3N_{zi}) \cdot N_{p})}.$$

Clearly,

$$\overline{\mathbf{T}}_{\mathbf{z}}^{\mathbf{i}} = \overline{\mathbf{C}}_{\mathbf{T}_{\mathbf{z}}}^{\mathbf{i}} \overline{\mathbf{x}}_{\mathbf{mpc}}^{\mathbf{i}} \,. \tag{5.69}$$

•

Using (5.55), (5.69) can be written as

$$\overline{\mathbf{T}}_{z}^{i} = \overline{\mathbf{C}}_{\mathbf{T}_{z}}^{i} \Big(\mathbf{M}_{i} \mathbf{x}_{mpc}^{i}(k) + \mathbf{S}_{ui} \overline{\Delta u}^{i} + \mathbf{S}_{ai} \overline{\mathbf{T}}_{a} + \mathbf{S}_{gi} \overline{\mathbf{T}}_{g} + \sum_{j \neq i} \mathbf{S}_{ij} \overline{\mathbf{T}}_{z}^{j} \Big).$$
(5.70)

Substituting for \overline{T}_{z}^{i} from (5.70) in (5.68) we obtain:

$$\mathbf{A}_{1}^{i}\overline{\Delta \mathbf{u}^{i}} \geq c_{pa}\overline{\mathbf{M}}_{\max}^{i} \left(\mathbf{A}_{2}^{i}\mathbf{T}_{\mathrm{supp}} - \mathbf{A}_{3}^{i}\mathbf{x}_{\mathrm{mpc}}^{i}(k) - \mathbf{A}_{4}^{i}\overline{\Delta \mathbf{u}^{i}} - \mathbf{A}_{5}^{i}\overline{\mathbf{T}}_{a} - \mathbf{A}_{6}^{i}\overline{\mathbf{T}}_{g} - \sum_{j\neq i}\mathbf{A}_{7}^{ij}\overline{\mathbf{T}}_{z}^{j} \right) - \mathbf{A}_{2}^{i}\mathbf{u}^{i}(k-1),$$
(5.71)

where,

$$\mathbf{A_3^i} = \begin{bmatrix} \mathbf{0}_{N_{zi} \times (N_x + 3N_{zi})} \\ \mathbf{\bar{C}}_{T_z}^i \mathbf{M}_i \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{T_z}^i \\ \mathbf{0}_{(N_p \cdot N_{zi}) \times (N_x + 3N_{zi})} \end{bmatrix}, \mathbf{A_4^i} = \begin{bmatrix} \mathbf{0}_{N_{zi} \times (N_u \cdot N_{zi})} \\ \mathbf{\bar{C}}_{T_z}^i \mathbf{S}_{ui} \end{bmatrix}, \mathbf{A_5^i} = \begin{bmatrix} \mathbf{0}_{N_{zi} \times N_p} \\ \mathbf{\bar{C}}_{T_z}^i \mathbf{S}_{ai} \end{bmatrix},$$
$$\mathbf{A_6^i} = \begin{bmatrix} \mathbf{0}_{N_{zi} \times N_p} \\ \mathbf{\bar{C}}_{T_z}^i \mathbf{S}_{gi} \end{bmatrix}, \mathbf{A_7^i} = \begin{bmatrix} \mathbf{0}_{N_{zi} \times (N_u \cdot N_{zj})} \\ \mathbf{\bar{C}}_{T_z}^i \mathbf{S}_{ij} \end{bmatrix}.$$

Note that in deriving (5.71), we used the fact that $\mathbf{T}_{z}^{i}(k) = \mathbf{C}_{T_{z}}^{i} \mathbf{x}_{mpc}^{i}(k)$. The inequality (5.71) can be re-stated as:

$$(\mathbf{A}_{1}^{i} + c_{pa}\overline{\mathbf{M}}_{\max}^{i}\mathbf{A}_{4}^{i})\overline{\Delta \mathbf{u}}^{i} \ge c_{pa}\overline{\mathbf{M}}_{\max}^{i} \left(\mathbf{A}_{2}^{i}\mathbf{T}_{\sup p} - \mathbf{A}_{3}^{i}\mathbf{x}_{mpc}^{i}(k) - \mathbf{A}_{5}^{i}\overline{\mathbf{T}}_{a} - \mathbf{A}_{6}^{i}\overline{\mathbf{T}}_{g} - \sum_{j\neq i}\mathbf{A}_{7}^{ij}\overline{\mathbf{T}}_{z}^{j}\right) - \mathbf{A}_{2}^{i}\mathbf{u}^{i}(k-1).$$

$$(5.72)$$

5.4.2.2 Heating constraints

For each zone $n \in \{1, 2, ..., N_{zi}\}$ in the *i*th cluster, the heating constraint due to the assumptions on the HVAC system mentioned earlier is given by (refer to nomenclature)

$$u_n^i(k) \le Q_{RH-max,n}^i. \tag{5.73}$$

The combined form of (5.73) for all zones in the cluster is given by

$$\mathbf{u}^{\mathbf{i}}(k) \le \mathbf{Q}^{\mathbf{i}}_{\mathbf{RH}-\mathbf{max}}.$$
 (5.74)

Equation (5.74) results in the following constraints for $l \in \{1, 2, ..., N_u - 1\}$ (the control horizon)

$$\mathbf{u}^{\mathbf{i}}(k-1) + \Delta \mathbf{u}^{\mathbf{i}}(k) + \Delta \mathbf{u}^{\mathbf{i}}(k+1|k) + \dots + \Delta \mathbf{u}^{\mathbf{i}}(k+1|k) \le \mathbf{Q}^{\mathbf{i}}_{\mathbf{RH-max}}.$$
 (5.75)

Equation (5.75) can be succinctly written as

$$\mathbf{A}_{1,1}^{i} \overline{\Delta \mathbf{u}}^{i} \le \overline{\mathbf{Q}}_{\mathbf{RH-max}}^{i} - \mathbf{A}_{7}^{i} \mathbf{u}^{i} (k-1), \qquad (5.76)$$

where, $A_{1,1}^{i}$ was defined earlier as

$$\mathbf{A}_{1,1}^{i} = \begin{bmatrix} \mathbf{I}_{N_{zi}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I}_{N_{zi}} & \mathbf{I}_{N_{zi}} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{N_{zi}} & \mathbf{I}_{N_{zi}} & \dots & \mathbf{I}_{N_{zi}} \end{bmatrix}_{(N_{zi}.N_{u}) \times (N_{zi}.N_{u})},$$

and,

$$\overline{\mathbf{Q}}_{\mathbf{R}\mathbf{H}-\mathbf{max}}^{\mathbf{i}} = \begin{bmatrix} \mathbf{Q}_{\mathbf{R}\mathbf{H}-\mathbf{max}}^{\mathbf{i}} \\ \mathbf{Q}_{\mathbf{R}\mathbf{H}-\mathbf{max}}^{\mathbf{i}} \\ \vdots \\ \mathbf{Q}_{\mathbf{R}\mathbf{H}-\mathbf{max}}^{\mathbf{i}} \end{bmatrix}_{(N_{zi}\cdot N_{u})\times 1}, \mathbf{A}_{7}^{\mathbf{i}} = \begin{bmatrix} \mathbf{I}_{N_{zi}} \\ \mathbf{I}_{N_{zi}} \\ \vdots \\ \mathbf{I}_{N_{zi}} \end{bmatrix}_{(N_{zi}\cdot N_{u})\times N_{zi}}$$

5.4.3 Quadratic program formulation

Using the results in Sections 5.4.1 and 5.4.2, the optimization problem corresponding to the decentralized controller for the i^{th} cluster (see section 5.3.2) is expressed as the following quadratic program (QP)

$$\overline{\Delta \mathbf{u}}_{*}^{i} = \arg\min_{\overline{\Delta \mathbf{u}}^{i}} g_{i}(\overline{\Delta \mathbf{u}}^{i}). \qquad (5.77)$$

where,

$$g_i(\overline{\Delta \mathbf{u}}^i) = (\overline{\Delta \mathbf{u}}^i)^T \mathbf{H}^i_{\mathbf{mpc}} \overline{\Delta \mathbf{u}}^i + (\mathbf{f}^i_{\mathbf{mpc}})^T \overline{\Delta \mathbf{u}}^i$$
(5.78)

subject to :

$$(\mathbf{A}_{1}^{i} + c_{pa}\overline{\mathbf{M}}_{\max}^{i}\mathbf{A}_{4}^{i})\overline{\Delta \mathbf{u}}^{i} \ge c_{pa}\overline{\mathbf{M}}_{\max}^{i} \left(\mathbf{A}_{2}^{i}\mathbf{T}_{supp} - \mathbf{A}_{3}^{i}\mathbf{x}_{mpc}^{i}(k) - \mathbf{A}_{5}^{i}\overline{\mathbf{T}}_{a} - \mathbf{A}_{6}^{i}\overline{\mathbf{T}}_{g} - \sum_{j\neq i}\mathbf{A}_{7}^{ij}\overline{\mathbf{T}}_{z}^{j}\right)$$
$$- \mathbf{A}_{2}^{i}\mathbf{u}^{i}(k-1),$$
 (5.79)

$$\mathbf{A}_{1,1}^{\mathbf{i}} \overline{\Delta \mathbf{u}}^{\mathbf{i}} \le \overline{\mathbf{Q}}_{\mathsf{RH-max}}^{\mathbf{i}} - \mathbf{A}_{\mathsf{7}}^{\mathbf{i}} \mathbf{u}^{\mathbf{i}} (k-1). \tag{5.80}$$

The above quadratic program can be solved at each time step
$$k$$
 using an appropriate tool

such as the optimization toolbox in MATLAB [92]. The optimal control input, $\mathbf{u}_*^{i}(k)$ at this

time instant, is given by

$$\mathbf{u}_{*}^{i}(k) = \mathbf{A}_{\mathbf{8}}^{i} \overline{\Delta \mathbf{u}}_{*}^{i} + \mathbf{u}^{i}(k-1), \qquad (5.81)$$

where,

$$\mathbf{A}_{\mathbf{8}}^{\mathbf{i}} = \begin{bmatrix} \mathbf{I}_{N_{zi}} & \mathbf{0}_{N_{zi} \times ((N_u - 1).N_{zi})} \end{bmatrix}.$$
(5.82)

The control input $\mathbf{u}_*^{i}(k)$ is applied to the *i*th cluster in the partition at the *k*th time step, and this procedure is then repeated for all clusters and all time steps.

5.5 Concluding remarks

A framework was presented in this chapter for the design of cluster level decentralized controllers corresponding to any partition of a building. The important elements of this framework were (i) a reduced order representation of the thermal dynamics associated with a cluster, (ii) a cluster level observer for estimation of unknown states and disturbances, and (iii) a cluster level controller based on an MPC framework which utilizes the reduced order model and the estimates provided by the observer.

Chapters 4 and 5 constitute a two-step decentralized control design process for the thermal control of a building. It consists of deciding a control architecture based on the tools developed in chapter 4, followed by design of decentralized controllers using the framework developed in this chapter. To demonstrate the use of these tools, the next chapter presents a real world building example, where this two-step procedure – involving control architecture selection and control design – was applied in simulation.

Chapter 6 Real World Building Simulation Study

6.1 Introduction

This chapter presents simulation studies using a real world building model to demonstrate the applicability of the tools developed in the previous chapters. A detailed model of the building is first developed in EnergyPlus [27] – a state-of-the-art modeling toolbox for buildings developed by the US Department of Energy (DOE). A linear representation of the thermal dynamics for the building is then obtained, which is used to partition it for decentralized control in accordance with the OLF-FPM method presented in Chapter 4. This is followed by the design of decentralized controllers using the methodology described in Chapter 5. Lastly, the optimality and robustness aspects associated with the decentralized controllers are investigated to conclude the chapter. The nomenclature used in this chapter is shown in Table 6.1

Symbol	Description
Nz	Number of zones in a building
$T_{z,j}$	Temperature of zone <i>j</i>
Q _{int,j}	Internal thermal load generated in zone <i>j</i>
C _{p,air}	Specific heat capacity of air
ṁ _{HVAC,j}	Mass flow rate of air supplied to zone <i>j</i> by the HVAC system
C _{zj}	Thermal capacitance of zone <i>j</i>

Table 6.1 Nomenclature of symbols used in Chapter 6

т	Sample time used for EnergyPlus simulations	
IS	(typically 1 minute in this work)	
$\dot{m}_{inf,j}$	Mass flow rate of infiltrated air from ambient to zone <i>j</i>	
T _a	Ambient temperature	
T _{w,i}	Temperature of wall surface <i>i</i>	
E +	- Superscript used to denote data obtained from EnergyPlus simulations	
*	Superscript used to denote optimal values or results obtained from an	
	optimization	
\mathcal{N}_{j}	Set of wall surfaces which enclose zone <i>j</i>	
h _{jl}	Heat transfer coefficient between zone <i>j</i> and wall surface <i>l</i>	
h _j	Heat transfer coefficient between zone <i>j</i> and any enclosing wall surface	
A _i	Area of wall surface <i>i</i>	
N	Length of time window (in samples) used in optimization	
$d_{w-in,i}$	Unmodeled thermal load acting on inside surface of wall <i>i</i>	
$T_{w-in,i}$	Temperature of inside surface of wall <i>i</i>	
Timti	Temperature of interface (zone/ambient/ground) which thermally interacts with	
1 int,i	inside surface of wall <i>i</i>	
T _{w-out,i}	Temperature of outside surface of wall <i>i</i>	
d _{w-out,i}	Unmodeled thermal load acting on outside surface of wall <i>i</i>	
Tauti	Temperature of interface (zone/ambient/ground) which thermally interacts with	
- ext,t	outside surface of wall <i>i</i>	
N _w	Number of wall surfaces in building	
d _w	Vector of unmodeled thermal loads acting on wall surfaces	
d _w	Vector of unmodeled thermal loads acting on zones	
$d_{w,i}$	Thermal load acting on wall surface <i>i</i>	
σ	Stefan-Boltzmann constant	
α _i	Solar absorptance of wall surface <i>i</i>	
q _{swr,i}	Incident solar radiation per unit area on wall surface <i>i</i>	

ϵ_i	Thermal absorptance of wall surface <i>i</i>
T _{gnd}	Temperature of ground outside building used for radiation calculations
T _{sky}	Sky temperature used for radiation calculations
$d_{z,i}$	Thermal load acting on zone <i>i</i>
$\eta_{occ,i}$	Actual occupancy in i^{th} zone as a fraction of nominal occupancy
N _{occ,i}	Nominal occupancy in <i>i</i> th zone
Qocc	Average rate of heat transfer from an occupant
$\eta_{light,i}$	Actual lighting thermal load in <i>i</i> th zone as a fraction of nominal lighting thermal load
$Q_{light,i}$	Nominal lighting thermal load in i^{th} zone
$\eta_{appl,i}$	Actual equipment thermal load in i^{th} zone as a fraction of nominal equipment thermal load
$Q_{appl,i}$	Nominal appliance thermal load in i^{th} zone
$T_{z,i,\mathrm{ref}}$	Set-point temperature for <i>i</i> th zone

6.2 Test building and EnergyPlus model

The test building considered in this chapter is a multi-zone building, modeled based on the layout of the Siemens Corporate Research (SCR) building located in Princeton, New Jersey (NJ). A photograph of the actual building is shown in Figure 6.1. The various features of the building, which are used in the development of an EnergyPlus model, are explained this section.

6.2.1 Building Layout and Geometry

A Google SketchUp [94] illustration of the building is shown in Figure 6.2, whose plan view is shown in Figure 6.3. It consists of 5 blocks marked C, D, E, F and G (Figure 6.4) and has three floors including a basement (Figure 6.5). The building has 9 thermal zones, details of which are provided in Table 6.2. Note that the basement of the building has a section which is not thermally conditioned, indicated by NTCB in Table 6.2and shown in Figure 6.6. Each thermal zone is catered by its own air handling unit, except the zone labeled NTCB. For a

background on thermal zones and air handling units in a building, the reader is directed to Chapter 2. As shown in Figure 6.2, the building also has external windows which are included in the development of an EnergyPlus model.



Figure 6.1 Photograph of the SCR building (source: [93])



Figure 6.2 Google SketchUp illustration of the test office building



Figure 6.3 Top (plan) view of test office building



Figure 6.4 Isometric view of the test office building

13'9	∮ Э 3/8" ↓	1	3' 9 3/8"	
13'9	9 3/8"	1	3' 9 3/8"	
11' 9	∲ 9 3/4" ∳	1	1' 9 3/4"	
_				

Figure 6.5 Side view of test office building viewed from the North-East direction



Figure 6.6 Transverse cut of basement showing conditioned and non-conditioned sections of test office building



Figure 6.7 Transverse cut of D block showing the server room of test office building

Thermal zone number	Description	Alias
1	Entire G block (includes both 1 st and 2 nd floor sections)	G
2	Entire E block (includes both 1 st and 2 nd floor sections)	Е
3	Entire C block (includes both 1 st and 2 nd floor sections)	С
4	Server room (Figure 6.7) located on 1 st floor of D block	SR
5	1 st floor section of F block	F1
6	Entire D block (includes both 1 st and 2 nd floor sections but excludes the server room)	D
7	2 nd floor section of F block	F2
8	Thermally controlled section of the basement (Figure 6.6)	TCB
9	Non-thermally controlled section of the basement (Figure 6.6)	NTCB

Table 6.2 Description of thermal zones in the test office building

6.2.2 Construction data

The construction data for the building was based on a standard 'medium office building' construction template for ASHRAE climate zone 5 (NJ lies in that zone) provided in the OpenStudio tool [95] developed by the National Renewable Energy Laboratory (NREL). This sets appropriate properties for various construction elements in the building such as exterior and internal walls, floors and ceilings, and doors and glass windows. A description of the material layers associated with the various construction elements in the 'medium office building' construction template is shown in Table 6.3.

		Material layers with	
construction element in 'medium	Applicability	thickness indicated in	
office building template		parentheses	
ASHRAE 1891-	Ceilings facing ambient	Roof membrane (0.0095 m),	
2000 ExtPact IEAD ClimateZone 2.5		roof insulation (0.2105 m),	
2009_ExtRool_IEAD_Childte2one 2-5		metal decking (0.0015 m)	
		Gypsum board (0.019 m),	
000_Interior Wall	Interior walls	air gap (0.15 m), gypsum	
		board (0.019 m)	
000_Interior Door	Interior doors	Wood (0.0254 m)	
		Stucco (0.0253 m), heavy-	
ASHRAE_189.1-	Walls facing	weight concrete (0.2033 m),	
2009_ExtWall_Mass_ClimateZone 5	ambient	wall insulation (0.0794 m),	
		gypsum (0.0127 m)	
	Interior floors/ceilings	Acoustic tile (0.0191 m), air	
000_Interior Floor		gap (0.18 m), light-weight	
		concrete (0.1016 m)	
000 ExtSlabCarpet 4 in ClimateZone	Floors facing	Heavy-weight concrete	
1_8	ground	(0.1016 m), carpet pad	
	Siouna	(0.2165 m)	
ASHRAE_189.1-	Windows	Theoretical glass (0.003 m)	
2009_ExtWindow_ClimateZone 4-5	facing ambient		

Table 6.3 Details of construction template used for the test office building

6.2.3 Weather data

Weather information is required as an input for simulating EnergyPlus models. The weather information is provided to EnergyPlus through a weather file which contains information such as dry bulb temperature, wet bulb temperature, relative humidity, wind speed, etc. for any location. A library of weather files for various locations around the world is provided

by the US DOE [96]. For the test building under consideration, a weather file specific to Princeton, NJ where the building is located was not available. Therefore, the weather file for the nearest location for which data is available (Trenton, NJ approximately 10 miles away) was used.

6.2.4 Internal loads

Internal loads are a key constituent of any EnergyPlus model. The calculation of internal thermal loads for each thermal zone in a building is performed in EnergyPlus based on information about occupancy, lighting and equipment corresponding to the zone.

6.2.4.1 Occupancy information

Occupancy information is specified by providing the nominal occupancy for the zones, occupancy schedules which determine what percentage of the nominal number of occupants is present at a given time, and activity schedules which determine the thermal contribution (in W) from each occupant at any time. The nominal occupancy can be entered in the form of total number of people, number of people per floor area or floor area per person. The occupancy and activity schedules can be provided as time-series data sampled hourly for each day in the week. Typically, three schedules – weekday schedule, Saturday schedule and Sunday schedule – are provided for occupancy and activity.

For the test building under consideration, the nominal occupancy values are shown in Table 6.4. Note that the table also shows nominal lighting and equipment loads which are explained later in this Section. The weekday and Saturday occupancy schedules are shown in Figure 6.8 and Figure 6.9 respectively. The Sunday occupancy is zero for the entire day. These occupancy schedules are based on a template called 'Medium_Office_Bldg_Occ' provided in the OpenStudio tool. The activity schedule for all zones at all times corresponds to a thermal contribution of 120 W per occupant. This value is based on the template "Medium_Office_Activity" provided in OpenStudio.

6.2.4.2 Lighting information

Similar to the occupancy information, the thermal load from lighting in each thermal zone in a building is specified by providing the nominal lighting thermal load (W) and lighting schedules which determine what percentage of the nominal lighting load is applicable at any
given time. The nominal lighting thermal load can be entered in the form of an absolute load (W), load per floor area (W/ft²) or load per person (W/person). The lighting schedules can be provided as time-series data sampled hourly for each day in the week.

For the test building under consideration, the nominal lighting thermal load values are shown in Table 6.4. The weekday and Saturday lighting schedules are shown in Figure 6.10 and Figure 6.11 respectively. Sunday's schedule for each zone is set to 5% of the corresponding nominal lighting thermal load at all times in the day. These lighting schedules are based on a template called 'Medium_Office_Bldg_Light' provided in the OpenStudio tool.

Table 6.4 Nominal occupancy, lighting load and equipment load information used inEnergyPlus model of test building

Zone alias	Nominal occupancy	Nominal lighting thermal load (kW)	Nominal equipment thermal load (W per person)
G	50	5.0	400
E	50	5.0	400
С	50	5.0	400
SR	0	5.0	400
F1	75	5.0	400
D	50	5.0	400
F2	75	5.0	400
ТСВ	10	5.0	400
NTCB	0	1.0	400



Figure 6.8 Weekday occupancy schedule in each thermal zone in the test building



Figure 6.9 Saturday's occupancy schedule in each thermal zone of the test building



Figure 6.10 Weekday lighting schedule in each thermal zone in the test building



Figure 6.11 Saturday's lighting schedule in each thermal zone in the test building

6.2.5 Creation of EnergyPlus model

An EnergyPlus model for the test building was created using the OpenStudio tool [95] based on the layout, construction, weather and internal load information described above. The steps used for construction of the model are explained in Appendix E. The description includes a link to online tutorials on OpenStudio. For the interested reader, the Openstudio and EnergyPlus files that were generated for the test building under consideration are included in the media accompanying this thesis.

In the EnergyPlus model constructed above, the 'ideal air loads' option was turned on. This option provides the requisite amount of heating or cooling to each thermally controlled zone in order to achieve specified set-point temperatures, without the need to set up an HVAC system. Consideration of the HVAC system is beyond the scope of this thesis. However, OpenStudio provides the option of specifying an appropriate HVAC system, which can be auto-sized depending on design weather conditions and loads. More details are available in the tutorials referenced in Appendix E.

6.3 Generation of linearized model

In this section, we describe the development of a linearized model, based on the 3R2C framework (see Section 5.2.1) for the test building described in section 6.2. The purpose of the linearized model is to allow the use of the OLF-FPM method described in Chapter 4 to partition the building for decentralized control.

6.3.1 Overview of modeling framework

The identification of resistances and capacitances to construct a RC network model of a building is performed in two steps, which we refer to as 'zone level identification' and 'wall surface level identification'. In this chapter, the term wall is used to represent a general term to represent all surfaces constituting the building, i.e. ceilings, floors, vertical walls (both internal and external), windows and doors. The underlying details of these steps are described in Sections 6.3.2 and 6.3.3 below. The 3R2C modeling paradigm for walls which is used in these steps was illustrated in Figure 5.1 and is reproduced below in Figure 6.12.



Figure 6.12 Schematic of 3R2C modeling paradigm for wall *i*.

The resistances and capacitances obtained through these steps are then used in Section 6.3.4 to create a linear time invariant (LTI) model of the building thermal dynamics.

6.3.2 Zone level identification

The zone level identification corresponds to the estimation of the following parameters

- 1. The thermal capacitance, c_{zj} associated with each zone j in the building.
- 2. The resistances $R_{w-in,i}$ and $R_{w-out,i}$ corresponding to each internal wall *i* (flanked by zones on both sides) in the building as shown in Figure 6.12.
- 3. The resistance $R_{w-in,i}$ corresponding to each external wall *i* (flanked by a zone on one side and ambient/ground on another) in the building as shown in Figure 6.12.

It is easy to verify that the above resistances are given by elements of the set $\{R_{jl} | j \in \{1,2,...,N_z\}, l \in \mathcal{N}_j\}$, where R_{jl} represents the thermal resistance between zone *j* and wall surface *l*, and \mathcal{N}_j represents the set of wall surfaces which enclose zone *j*. EnergyPlus uses the following

difference equation obtained using the Backward Euler method to represent the thermal dynamics for the j^{th} zone (see nomenclature for details on notation)

$$T_{z,j}(k+1)$$

$$=\frac{\dot{q}_{int,j}(k+1)+c_{p,air}\dot{m}_{HVAC,j}(k+1)T_{HVAC,j}(k+1)+\frac{c_{zj}T_{z,j}(k)}{T_{s}}+c_{p,air}\dot{m}_{inf,j}(k)T_{a}(k)+\sum_{l\in\mathcal{N}_{j}}(UA)_{jl}(k)T_{w,l}(k)}{\frac{c_{zj}}{T_{s}}+c_{p,air}\dot{m}_{HVAC,j}(k+1)+c_{p,air}\dot{m}_{inf,j}(k)+\sum_{l\in\mathcal{N}_{j}}(UA)_{jl}(k)}$$
(6.1)

The above equation represents the conservation of energy (first law of Thermodynamics) for the zone. The interested reader is directed to [97] for details its derivation. Here, $(UA)_{jl}$ represents the overall heat transfer coefficient between the zone *j* and wall surface *l* multiplied by the area A_l of surface *l*. This is related to R_{jl} introduced earlier by

$$R_{jl} = \frac{1}{(UA)_{jl}}.$$
 (6.2)

Equation (6.1) can be used to identify the unknown capacitance c_{zj} and the thermal resistances R_{jl} where $l \in \mathcal{N}_j$, based on data obtained from an EnergyPlus simulation. This can be formally expressed as a least-squares identification problem for zone *j* given by the optimization below (see nomenclature for details on notation)

$$\left\{c_{zj}^{*}, \left\{(UA)_{jl}^{*}\right\}_{l \in \mathcal{N}_{j}}\right\} = \arg\min_{\left\{c_{zj}, \left\{(UA)_{jl}\right\}_{l \in \mathcal{N}_{j}}\right\}} \sum_{k=0}^{N-1} \left(T_{z,j}(k+1) - T_{z,j}^{E+}(k+1)\right)^{2}$$
(6.3)

Subject to:

$$T_{z,j}(k+1) = \frac{\dot{Q}_{int,j}^{E+}(k+1) + c_{p,air} \dot{m}_{HVAC,j}^{E+}(k+1) T_{HVAC,j}^{E+}(k+1) + \frac{c_{zj} T_{z,j}(k)}{T_s} + c_{p,air} \dot{m}_{inf,j}^{E+}(k) T_a(k) + \sum_{l \in \mathcal{N}_j} (UA)_{jl} T_{w,l}^{E+}(k)}{\frac{c_{zj}}{T_s} + c_{p,air} \dot{m}_{HVAC,j}^{E+}(k+1) + c_{p,air} \dot{m}_{inf,j}^{E+}(k) + \sum_{l \in \mathcal{N}_j} (UA)_{jl}}, (6.4)$$

$$0 \le c_{zj} \le c_{zj,max}, \qquad (6.5)$$

$$0 \le (UA)_{jl} \le (UA)_{jl,max} \quad \text{for all } l \in \mathcal{N}_j. \qquad (6.6)$$

In the above optimization framework, the superscript "E+" is used to indicate data obtained from an EnergyPlus simulation and the superscript " *" represents optimal values or the result of the optimization. *N* is the length of the time window used for simulation measured

in terms of number of samples, where the sample period is T_s seconds. Equation (6.4) is based on (6.1), whereas the constraints (6.5) and (6.6) are included to prevent the capacitance c_{zj} and coefficients $(UA)_{jl}$ from becoming negative or unbounded. Solution of the above optimization problem for each zone $j \in \{1, 2, ..., N_z\}$ provides the required zone capacitances c_{zj} and resistances R_{jl} (computed from $(UA)_{jl}$ via (6.2)).

Data for use in the above optimization process is obtained by providing persistently excited zone set-point temperature signals in EnergyPlus in time-series format. In the simulation process, the 'fictitious' HVAC system corresponding to the 'ideal air loads' option heats or cools the zones, as needed, to achieve the specified set-point temperatures at each time instant.

The above optimization was performed for zone F2 in the test building using the 'fmincon' command in the MATLAB optimization toolbox [98]. The optimization time window was chosen to be 24 hours, from 12:00 AM on June 3 to 12:00 AM on June 4. The set-point temperature signal used to obtain EnergyPlus simulation data is shown in Figure 6.13, which is a pseudo-random binary signal (PRBS) generated using MATLAB. The relevant codes for performing this optimization are provided in Appendix F.



Figure 6.13 Zone set point temperature signal from 12:00 AM to 1:00 AM used in EnergyPlus simulations for zone level identification

Zone F2 is surrounded by 9 surfaces. The corresponding values of coefficients, $(UA)_{jl}$ in units of K/kW obtained from the optimization are 0, 0, 0, 0, 0, 0, 0, 0.617 and 3.857. A comparison of the zone temperature predicted using the identified model with the zone temperature obtained via EnergyPlus simulation is shown in Figure 6.14. It is observed that the zone temperature predicted using the identified model is very close to that obtained from EnergyPlus simulations all time instants. However, the identified parameter values are unrealistic. In particular, coefficients $(UA)_{jl}$ between zone F2 and seven of the surrounding surfaces are zero, which is physically untenable. A possible reason for this solution is the presence of too many unknown parameters (10 for zone F2 – 9 resistances and 1 capacitance) in the optimization framework. This may lead to a situation where the same total energy transfer to a zone from the surrounding surfaces can be achieved by multiple combinations of surface to zone heat transfer coefficients. Therefore, some of the heat transfer coefficients can be set to zero by augmenting the remaining heat transfer coefficients in such a way that the total energy transfer from the surfaces to the zone is the same as that corresponding to the EnergyPlus simulation.



Figure 6.14 Temperature of zone F2 from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM

In an attempt to address the above problem, potentially arising due over-parameterization in the optimization framework, we propose a simplified framework which is based upon the following physical reasoning. Each coefficient $(UA)_{jl}$ (where $l \in \mathcal{N}_j$) can be written as the product of h_{jl} and A_j , which denote the heat transfer coefficient between zone j and surface l, and the area of surface l respectively. The coefficient h_{jl} corresponds to convection heat transfer which is primarily dependent on the properties of air and temperatures of zone j and surface l. If we ignore the dependence of h_{jl} on the temperatures of zone j and surface l in the range of operation of the building, we can assume that each of the heat transfer coefficients h_{jl} are same for all $l \in \mathcal{N}_j$. This results in a simplified modeling framework which only uses a single heat transfer coefficient denoted by h_j between zone j and all surfaces l enclosing it. Therefore, the following framework for least squares system identification is proposed.

$$\left\{c_{zj}^{*}, h_{j}^{*}\right\} = \underset{\left\{c_{zj}, h_{j}\right\}}{\arg\min} \quad \sum_{k=0}^{N-1} \left(T_{z,j}(k+1) - T_{z,j}^{E+}(k+1)\right)^{2} \tag{6.7}$$

Subject to:

$$T_{z,j}(k+1) = \frac{\dot{q}_{int,j}^{E+}(k+1) + c_{p,air}\dot{m}_{HVAC,j}^{E+}(k+1)T_{HVAC,j}^{E+}(k+1) + \frac{c_{zj}T_{z,j}(k)}{T_s} + c_{p,air}\dot{m}_{inf,j}^{E+}(k)T_a(k) + h_j \sum_{l \in \mathcal{N}_j} A_l T_{w,l}^{E+}(k)}{\frac{c_{zj}}{T_s} + c_{p,air}\dot{m}_{HVAC,j}^{E+}(k+1) + c_{p,air}\dot{m}_{inf,j}^{E+}(k) + h_j \sum_{l \in \mathcal{N}_j} A_l}, \quad (6.8)$$

$$0 \le c_{zj} \le c_{zj,max}, \quad (6.9)$$

$$0 \le h_j \le h_{j,max}. \tag{6.10}$$

Similar to the optimization framework given by (6.3) to (6.6), in the above equations, the superscript "E+" is used to indicate data obtained from an EnergyPlus simulation and the superscript "*" represents optimal values or the result of the optimization.

The above optimization was performed for all zones in the test building. The optimization time window was chosen to be 24 hours, from 12:00 AM on June 3 to 12:00 AM on June 4. The set-point temperature signal used to obtain EnergyPlus simulation data is shown in Figure 6.13 which is the same as that used for the optimization framework described earlier. The result of the optimization for each zone is shown in Table 6.5. A comparison of zone temperatures predicted

using the identified models with zone temperatures obtained via EnergyPlus simulation is shown in Figure 6.15 to Figure 6.23. It is observed that the zone temperatures predicted using the identified models are very close to those obtained from EnergyPlus simulations all time instants. The relevant codes for performing this optimization are provided in Appendix G.

Thermal zone number (j)	Alias	<i>c</i> [*] _{<i>zj</i>} (kJ/K)	h_j^* (K/kW-m ²)
1	G	6.182×10^{3}	2.005×10^{-3}
2	E	7.39 × 10 ³	1.9×10^{-3}
3	C	8.89 × 10 ³	1.795×10^{-3}
4	SR	7.06×10^{2}	1.924×10^{-3}
5	F1	3.709×10^{3}	2.228×10^{-3}
6	D	7.391 × 10 ³	1.921×10^{-3}
7	F2	3.717×10^{3}	2.211×10^{-3}
8	TCB	3.449×10^{3}	1.651×10^{-3}
9	NTCB	6.499×10^{3}	8.85×10^{-4}

Table 6.5 Results of zone level optimization for all zones in test building



Figure 6.15 Temperature of zone G from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM



Figure 6.16 Temperature of zone E from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM



Figure 6.17 Temperature of zone C from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM



Figure 6.18 Temperature of zone SR from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM



Figure 6.19 Temperature of zone F1 from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM



Figure 6.20 Temperature of zone D from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM



Figure 6.21 Temperature of zone F2 from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM



Figure 6.22 Temperature of zone TCB from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM



Figure 6.23 Temperature of zone NTCB from zone level least squares identification methodology compared with EnergyPlus data between 12:00 AM to 1:00 AM

The thermal resistance R_{jl} between zone *j* and its enclosing surfaces *l* can be computed from the solution of the optimization problem given by (6.7) to (6.10) as shown below

$$R_{jl} = \frac{1}{(UA)_{jl}} = \frac{1}{h_j^* A_l}.$$
 (6.11)

As stated earlier, the resistances desired to be computed from the zone level identification are given by elements of the set $\{R_{jl} | j \in \{1, 2, ..., N_z\}, l \in \mathcal{N}_j\}$. Therefore, the use of (6.11) allows the computation of all such resistances. The thermal capacitances associated with the zones appear as optimization variables in (6.7) and therefore are directly provided by the optimization. The calculation of resistances and capacitances in this way completes the zone level identification. For the test building under consideration, the capacitances obtained are shown in Table 6.5. The resistances are computed from the h_j^* values shown in this table in a spreadsheet using (6.11) which is provided in the media accompanying this thesis.

6.3.3 Wall surface level identification

The wall surface level identification involves estimation of the following parameters with reference to Figure 6.12:

- 1. For internal walls (flanked by zones on both side): $R_{ww,i}$, $C_{w-in,i}$ and $C_{w-out,i}$
- For external walls (flanked by a zone on one side and ambient/ground on another): R_{w-out,i}, R_{ww,i}, C_{w-in,i}, C_{w-out,i}

The above parameters, together with the parameters estimated using zone level identification in Section 6.3.2 provide the complete set of resistances and capacitances required to set up a Linear Time Invariant (LTI) model of the building thermal dynamics. Two variants of a least squares identification procedure – treating wall surface thermal loads as known and unknown respectively – were first implemented for the wall surface level identification. However, both these methods were found to have critical limitations and provided unsatisfactory results. Therefore, as an alternative, the wall surface level identification was performed by computing the values of the desired resistances and capacitances directly using material properties of the wall construction layers. The two least squares identification procedures mentioned above are presented below, followed by the procedure for direct computation of parameters.

6.3.3.1 Least squares identification with unknown wall thermal loads

The least squares identification procedure described here treats the wall thermal loads $(d_{w-out,i} \text{ and } d_{w-in,i})$ as unknowns to be determined along with resistances and capacitances. The identification procedure involves solving a pair of optimization problems for each internal and external wall.

The pair of optimization problems corresponding to the i^{th} internal wall is given by:

(i)
$$\left\{ R_{ww,i}^*, C_{w-in,i}^*, \left\{ d_{w-in,i}^*(l) \right\}_{l=0}^{N-1} \right\} =$$

$$\arg \min_{\left\{R_{ww,i},C_{w-in,i},\left\{d_{w-in,i}(l)\right\}_{l=0}^{N-1}\right\}} \sum_{k=0}^{N-1} \left(T_{w-in,i}(k+1) - T_{w-in,i}^{E+}(k+1)\right)^{2} (6.12)$$

Subject to:

$$C_{w-in,i} \frac{T_{w-in,i}(k+1) - T_{w-in,i}(k)}{T_s} = \frac{T_{int,i}^{E+}(k) - T_{w-in,i}(k)}{R_{w-in,i}} + \frac{T_{w-out,i}^{E+}(k) - T_{w-in,i}(k)}{R_{ww,i}} + d_{w-in,i}(k)$$
for all $k \in \{0, 1, ..., N-1\}$, (6.13)

$$0 \le R_{ww,i} \le R_{ww,i,max},\tag{6.14}$$

$$0 \le C_{w-in,i} \le C_{w-in,i,max},\tag{6.15}$$

$$d_{w-in,i,min} \le d_{w-in,i}(k) \le d_{w-in,i,max}$$
 for all $k \in \{0,1,\dots,N-1\}$. (6.16)

(ii)
$$\left\{ C_{w-out,i}^{*}, \left\{ d_{w-out,i}^{*}(l) \right\}_{l=0}^{N-1} \right\} =$$

$$\arg \min_{\left\{ C_{w-out,i}, \left\{ d_{w-out,i}(l) \right\}_{l=0}^{N-1} \right\}} \sum_{k=0}^{N-1} \left(T_{w-out,i}(k+1) - T_{w-out,i}^{E+}(k+1) \right)^{2} (6.17)$$

Subject to:

$$C_{w-out,i} \frac{T_{w-out,i}(k+1) - T_{w-out,i}(k)}{T_s} = \frac{T_{ext,i}^{E+}(k) - T_{w-out,i}(k)}{R_{w-out,i}} + \frac{T_{w-in,i}^{E+}(k) - T_{w-out,i}(k)}{R_{ww,i}} + d_{w-out,i}(k)$$

for all $k \in \{0, 1, \dots, N-1\}, (\ 6.18\)$

$$0 \le C_{w-out,i} \le C_{w-out,i,max},\tag{6.19}$$

$$d_{w-out,i,min} \le d_{w-out,i}(k) \le d_{w-out,i,max}$$
 for all $k \in \{0,1,...,N-1\}$. (6.20)

The notations used in the above optimization problems are explained in the nomenclature (Table 6.1). Equations (6.13) and (6.18) represent energy conservation equations (first law of Thermodynamics) applied to the wall surfaces. Note that since *i* is an internal wall, $T_{int,i}^{E+}(k)$ and $T_{ext,i}^{E+}(k)$ appearing above represent temperatures of zones that flank it on either side. The resistances $R_{w-in,i}$ and $R_{w-out,i}$ are assumed to be known since they can be determined from the zone level identification described in Section 6.3.2. The length of the optimization time-window *N*, and the bounds $R_{ww,i,max}$, $C_{w-in,i,max}$, $d_{w-in,i,min}$, $d_{w-in,i,max}$, $C_{w-out,i,max}$, $d_{w-out,i,min}$ and $d_{w-out,i,max}$ are treated as design parameters which can be tuned if necessary. The pair of optimization problems corresponding to the *i*th external wall is given by:

(i)
$$\left\{ R_{ww,i}^*, C_{w-in,i}^*, \left\{ d_{w-in,i}^*(l) \right\}_{l=0}^{N-1} \right\} =$$

$$\arg \min_{\left\{R_{ww,i},C_{w-in,i},\left\{d_{w-in,i}(l)\right\}_{l=0}^{N-1}\right\}} \sum_{k=0}^{N-1} \left(T_{w-in,i}(k+1) - T_{w-in,i}^{\mathrm{E}+}(k+1)\right)^{2} (6.21)$$

Subject to:

$$C_{w-in,i} \frac{T_{w-in,i}(k+1) - T_{w-in,i}(k)}{T_s} = \frac{T_{int,i}^{\mathrm{E}+}(k) - T_{w-in,i}(k)}{R_{w-in,i}} + \frac{T_{w-out,i}^{\mathrm{E}+}(k) - T_{w-in,i}(k)}{R_{ww,i}} + d_{w-in,i}(k)$$

for all
$$k \in \{0, 1, \dots, N - 1\}$$
, (6.22)

$$0 \le R_{ww,i} \le R_{ww,i,max},\tag{6.23}$$

$$0 \le C_{w-in,i} \le C_{w-in,i,max},\tag{6.24}$$

$$d_{w-in,i,min} \le d_{w-in,i}(k) \le d_{w-in,i,max}$$
 for all $k \in \{0,1,\dots,N-1\}$. (6.25)

(ii)
$$\begin{cases} R_{w-out,i}^{*}, C_{w-out,i}^{*}, \left\{ d_{w-out,i}^{*}(l) \right\}_{l=0}^{N-1} \end{cases} = \\ \underset{\{R_{w-out,i}, C_{w-out,i}, \left\{ d_{w-out,i}(l) \right\}_{l=0}^{N-1} \end{cases}}{\operatorname{arg min}} \sum_{k=0}^{N-1} \left(T_{w-out,i}(k+1) - T_{w-out,i}^{E+}(k+1) \right)^{2} (6.26) \end{cases}$$

Subject to:

$$C_{w-out,i} \frac{T_{w-out,i}(k+1) - T_{w-out,i}(k)}{T_s} = \frac{T_{ext,i}^{E+}(k) - T_{w-out,i}(k)}{R_{w-out,i}} + \frac{T_{w-in,i}^{E+}(k) - T_{w-out,i}(k)}{R_{ww,i}} + d_{w-out,i}(k)$$

for all $k \in \{0, 1, \dots, N - 1\}$, (6.27)

$$0 \le R_{w-out,i} \le R_{w-out,i,max},\tag{6.28}$$

$$0 \le C_{w-out,i} \le C_{w-out,i,max},\tag{6.29}$$

$$d_{w-out,i,min} \le d_{w-out,i}(k) \le d_{w-out,i,max}$$
 for all $k \in \{0,1, ..., N-1\}$. (6.30)

The notations used in the above optimization problems are explained in the nomenclature (Table 6.1). Note that since *i* is an external wall, $T_{int,i}^{E+}(k)$ represents a zone temperature and $T_{ext,i}^{E+}(k)$ represent ambient/ground temperature. The resistance $R_{w-in,i}$ is assumed to be known since it can be determined from the zone level identification described in Section 6.3.2. The length of the optimization time-window *N*, and the upper and lower bounds $R_{ww,i,max}$, $C_{w-in,i,max}$, $d_{w-in,i,min}$, $d_{w-in,i,max}$, $R_{w-out,i,max}$, $C_{w-out,i,max}$, $d_{w-out,i,min}$ and $d_{w-out,i,max}$ are treated as design parameters which can be tuned if necessary.

The above identification procedure was applied to the wall number 34 in the model, which is an external wall that faces zone F2 on one side and ambient on the other. A 10 hour long time window corresponding to 8 am - 6 pm was used for the optimization, which

corresponds to the period when the building is occupied. The data used for the EnergyPlus simulation was the same as that generated in Section 6.3.2 using a PRBS set-point signal. The sample time, T_s was set to 1 minute. The upper bound, $R_{ww,34,max}$, was chosen to be 100 K/kW. Figure 6.24 shows a comparison of predicted values of inside surface temperature $T_{w-in.34}$ obtained from the identified model with its values obtained from the EnergyPlus simulation. It is observed that the two plots match very well. However, an investigation of the identified parameters reveals that the identified resistance $R_{ww,34}$ has a value of 96 K/kW which is significantly large when compared to the theoretical value of 31 K/kW computed from material properties (see Section 6.3.3.3 for details on computation based on material properties). Similarly, on applying a more conservative upper bound of $R_{ww,34,max} = 40$ K/kW, the predictions from the identified model matched well with the EnergyPlus simulation as shown in Figure 6.25. The identified resistance $R_{ww,34}$ had a value of 40 K/kW in this case, which was close to its afore-mentioned theoretical value. Upon further lowering R_{ww,34,max} to 10 K/kW, predictions from identified model were again close to EnergyPlus simulation data (Figure 6.26), but $R_{ww,34}$ was identified as 9 K/kW which was significantly smaller than its theoretical value. Note that for visual clarity, the results in Figure 6.24 to Figure 6.26 are shown only for the first hour of the optimization time window, i.e. from 8 am to 9 am. The relevant codes for performing these optimizations are provided in Appendix H.

These observations suggest that the least squares identification methodology presented above is not reliable since the identified resistance appears to be a function of the upper bound imposed on it. This behavior can be explained on the basis that since $d_{w-in,i}(k)$ is an independent variable in the above optimization framework, for each $k \in \{0,1, ..., N-1\}$, it can be chosen in such a way that $T_{w-in,i}$ matches $T_{w-in,i}^{E+}$, irrespective of the value of $R_{ww,i}$. This over-parameterization can be easily verified from the following constraint used in the optimization

$$C_{w-in,i}\frac{T_{w-in,i}(k+1)-T_{w-in,i}(k)}{T_s} = \frac{T_{int,i}^{E+}(k)-T_{w-in,i}(k)}{R_{w-in,i}} + \frac{T_{w-out,i}^{E+}(k)-T_{w-in,i}(k)}{R_{ww,i}} + d_{w-in,i}(k).$$

To address the above limitation of the least squares methodology presented, we seek a

framework where $d_{w-in,i}(k)$ is not an independent variable for all values of $k \in \{0, 1, ..., N-1\}$, as proposed in the next section.



Figure 6.24 Inside surface temperature of wall 34 from least squares identification methodology (section 6.3.3.1) compared with EnergyPlus data ($R_{ww,34,max} = 100$ K/kW)



Figure 6.25 Inside surface temperature of wall 34 from least squares identification methodology (section 6.3.3.1) compared with EnergyPlus data ($R_{ww,34,max} = 40$ K/kW)



Figure 6.26 Inside surface temperature of wall 34 from least squares identification methodology (section 6.3.3.1) compared with EnergyPlus data ($R_{ww,34,max} = 10$ K/kW)

6.3.3.2 Least squares identification with known wall thermal loads

The least squares identification procedure described here assumes that the wall thermal loads $d_{w-out,i}(k)$ and $d_{w-in,i}(k)$ are known at each time k. This is based on the assumption that these thermal loads are a result of radiation – both long-wave and short-wave – acting on the wall surfaces. Radiation heat transfer associated with each wall surface can be obtained from EnergyPlus simulations. The reader is directed to [97] for more details on procedure to obtain radiation data from EnergyPlus.

The identification procedure involves solving a pair of optimization problems for each internal and external wall, which are modifications of the optimization problems in Section 6.3.3.1. The pair of optimization problems corresponding to the i^{th} internal wall is given by:

(i)
$$\{R_{ww,i}^*, C_{w-in,i}^*\} = \underset{\{R_{ww,i}, C_{w-in,i}\}}{\arg\min} \sum_{k=0}^{N-1} \left(T_{w-in,i}(k+1) - T_{w-in,i}^{E+}(k+1)\right)^2 (6.31)$$

Subject to:

$$C_{w-in,i} \frac{T_{w-in,i}(k+1) - T_{w-in,i}(k)}{T_s} = \frac{T_{int,i}^{E+}(k) - T_{w-in,i}(k)}{R_{w-in,i}} + \frac{T_{w-out,i}^{E+}(k) - T_{w-in,i}(k)}{R_{ww,i}} + d_{w-in,i}^{E+}(k)$$

for all $k \in \{0, 1, \dots, N - 1\}$, (6.32)

$$0 \le R_{ww,i} \le R_{ww,i,max},\tag{6.33}$$

$$0 \le C_{w-in,i} \le C_{w-in,i,max}. \tag{6.34}$$

(ii)
$$C_{w-out,i}^* = \underset{C_{w-out,i}}{\operatorname{arg min}} \sum_{k=0}^{N-1} \left(T_{w-out,i}(k+1) - T_{w-out,i}^{E+}(k+1) \right)^2$$
 (6.35)

Subject to:

$$C_{w-out,i} \frac{T_{w-out,i}(k+1) - T_{w-out,i}(k)}{T_s} = \frac{T_{ext,i}^{E+}(k) - T_{w-out,i}(k)}{R_{w-out,i}} + \frac{T_{w-in,i}^{E+}(k) - T_{w-out,i}(k)}{R_{ww,i}} + d_{w-out,i}^{E+}(k)$$

for all $k \in \{0, 1, \dots, N-1\}, (\ 6.36\)$

$$0 \le C_{w-out,i} \le C_{w-out,i,max}.$$
 (6.37)

The pair of optimization problems corresponding to the i^{th} external wall is given by:

(i)
$$\left\{R_{ww,i}^*, C_{w-in,i}^*\right\} = \arg\min_{\left\{R_{ww,i}, C_{w-in,i}\right\}} \sum_{k=0}^{N-1} \left(T_{w-in,i}(k+1) - T_{w-in,i}^{E+}(k+1)\right)^2 (6.38)$$

Subject to:

$$C_{w-in,i} \frac{T_{w-in,i}(k+1) - T_{w-in,i}(k)}{T_s} = \frac{T_{int,i}^{E+}(k) - T_{w-in,i}(k)}{R_{w-in,i}} + \frac{T_{w-out,i}^{E+}(k) - T_{w-in,i}(k)}{R_{ww,i}} + d_{w-in,i}^{E+}(k)$$

for all $k \in \{0, 1, \dots, N-1\}, (\, 6.39 \,)$

$$0 \le R_{ww,i} \le R_{ww,i,max},\tag{6.40}$$

$$0 \le C_{w-in,i} \le C_{w-in,i,max}.$$
(6.41)

(ii)
$$\left\{R_{w-out,i}^*, C_{w-out,i}^*\right\} =$$

$$\arg\min_{\{R_{w-out,i},C_{w-out,i}\}} \sum_{k=0}^{N-1} \left(T_{w-out,i}(k+1) - T_{w-out,i}^{E+}(k+1) \right)^2 (6.42)$$

Subject to:

$$C_{w-out,i} \frac{T_{w-out,i}(k+1) - T_{w-out,i}(k)}{T_s} = \frac{T_{ext,i}^{E+}(k) - T_{w-out,i}(k)}{R_{w-out,i}} + \frac{T_{w-in,i}^{E+}(k) - T_{w-out,i}(k)}{R_{ww,i}} + d_{w-out,i}^{E+}(k)$$

for all $k \in \{0, 1, \dots, N-1\}, (6.43)$

$$0 \le R_{w-out,i} \le R_{w-out,i,max},\tag{6.44}$$

$$0 \le C_{w-out,i} \le C_{w-out,i,max}.$$
(6.45)

The notations used in the above optimization problems are explained in the nomenclature (Table 6.1). Similar to Section 6.3.3.1, the above identification procedure was applied to the inside surface of wall 34 in the model. A 10 hour long time window corresponding to 8 am – 6 pm was used for the optimization. The data used for the EnergyPlus simulation was the same as that generated in Section 6.3.2 using a PRBS set-point signal and the sample time, T_s was set to 1 minute. Figure 6.27 shows the comparison of predicted values of surface temperature $T_{w-in.34}$ obtained using the identified model with its values obtained from the EnergyPlus simulation. It was observed that the two plots deviate significantly from one another. The relevant codes for performing this optimization are provided in Appendix I.



Figure 6.27 Inside surface temperature of wall 34 from least squares identification methodology (section 6.3.3.2) compared with EnergyPlus data

A possible reason for the above deviation is the difference in the order of the corresponding models. While the identified model for each wall is second because of its representation by two capacitances, EnergyPlus uses a model which is much higher in order, typically representing each construction layer by 6 - 18 capacitances connected by resistances [97].

The inability of the two identification approaches presented in Sections 6.3.3.1 and 6.3.3.2 to provide accurate and reliable parameter estimates suggests that data-based identification from EnergyPlus is not a suitable framework for obtaining 3R2C models for walls.

Therefore, we proceed to calculate the resistances and capacitances associated with the walls directly based on material properties of their corresponding construction layers, as described in the next section.

6.3.3.3 Direct computation of parameters

The steps to compute the parameters associated with the walls from the material properties of the construction layers are as follows:

1. For each wall type in the building (Table 6.3), the quantities \overline{C} and \overline{R} are computed as follows:

$$\bar{C} = \sum_{j=1}^{n} \rho_j c_{pj} l_j \tag{6.46}$$

$$\bar{R} = \sum_{i=1}^{n} \frac{l_j}{k_j}$$
(6.47)

In the above equations, *n* is the number of material layers, whereas ρ_j , c_{pj} , l_j and k_j represent the density, specific heat capacity, thickness and thermal conductivity, respectively, of layer *j*.

2. For each wall i, the total capacitance associated with it, C_i is computed as

$$C_i = A_i \bar{C_i}, \tag{6.48}$$

where A_i is the surface area of wall *i* and \overline{C}_i is the value of \overline{C} for the corresponding wall type computed using (6.46). We assume that the capacitance C_i is equally divided to represent the capacitances associated with the two surfaces of wall *i*. Therefore,

$$C_{w-in,i} = C_{w-out,i} = \frac{C_i}{2}.$$
 (6.49)

3. For each wall *i*, the resistance $R_{ww,i}$ associated with heat transfer between its two surfaces, is computed as

$$R_{ww,i} = \frac{\bar{R}_i}{A_i},\tag{6.50}$$

where, as before, A_i is the surface area of wall *i* and \overline{R}_i is the value of \overline{R} for the corresponding wall type computed using (6.47).

The above calculations together with the resistances calculated via the zone level

identification described in section 6.3.2 provide all the parameters shown in Figure 6.12 except the resistances $R_{w-out,i}$ between external walls *i* and ambient/ground. These resistances are computed as follows:

1. The convection heat transfer coefficient, $h_{w-out,i}$ between ambient and an external wall *i* facing ambient can be directly obtained from EnergyPlus simulations. Let $\bar{h}_{w-out,i}$ denote the average value of $h_{w-out,i}$ over an appropriate simulation window such as 24 hours, i.e.

$$\bar{h}_{w-out,i} = \frac{\sum_{k=0}^{N-1} h_{w-out,i}(k)}{N}, \qquad (6.51)$$

where *N* denotes the number of samples in the simulation time window. The resistance $R_{w-out,i}$ is then computed as

$$R_{w-out,i} = \frac{1}{A_i \overline{h}_{w-out,i}}.$$
 (6.52)

2. The resistance $R_{w-out,i}$ between ground and an external wall *i* facing ground is set to 0, which is consistent with the modeling assumption in EnergyPlus that ground facing walls have the same temperature as ground. In other words, for such a wall, the temperature $T_{w-out,i}$ of the surface facing ground is not a state. Therefore, the dynamics for such a wall is first order represented only by the state $T_{w-in,i}$.

For the test building under consideration, the calculated values of \bar{C} and \bar{R} using (6.46) and (6.47) for the various wall types are shown in Table 1. Using the steps mentioned above, all resistances and capacitances corresponding to the wall surface level identification were computed using a spreadsheet which is provided in the media accompanying this thesis.

6.3.4 Construction of LTI model

The resistances and capacitances computed in sections 6.3.2 and 6.3.3 were used to construct an LTI model to represent the thermal dynamics for the test building. This was done in accordance with the graph based procedure described in Algorithm 5.1 The code used to obtain the model is provided in Appendix J. The resulting state space model has the structure shown in (5.1) and is of order 112. Of these, 103 states represent wall temperatures for the inside and outside surfaces and 9 states represent zone temperatures.

Wall type	\overline{C} (kJ/K-m ²)	\overline{R} (K/kW-m ²)
Ceilings facing ambient	6.705×10^{1}	4.355×10^{3}
Interior walls	3.314×10^{1}	3.387×10^2
Interior doors	2.517×10^{1}	1.693×10^2
Walls facing ambient	4.271×10^{2}	2.071×10^{3}
Interior floors/ceilings	1.188×10^{3}	2.577×10^{3}
Floors facing ground	1.19×10^{2}	2.94×10^{2}
Windows facing ambient	6.3	2.256×10^2

Table 6.6 \overline{C} and \overline{R} values for various wall types in the test building

6.4 Control architecture determination

In this section, the LTI model developed in Section 6.3.4 for the test building under investigation is used to determine appropriate control architectures which provide a satisfactory tradeoff between optimality and robustness. The OLF-FPM method described in Chapter 4 is used for this purpose. The reasons for using the OLF-FPM method over the CLF-MCS method were explained in Chapter 4.

6.4.1 Modifications to LTI model

The EnergyPlus model of the test building that was constructed in Section 6.2 was based on the assumption that the thermal zones are completely separated from one another by solid walls. However in the actual building being modeled, openings are present in the walls at several places to facilitate movement of people in the building as illustrated in Figure 6.28. From a thermodynamic point of view, these openings allow direct thermal interaction between the zones, in addition to the thermal interactions occurring through the walls which separate them. These additional thermal interactions can be modeled as resistances which directly couple the associated zone capacitances.



Figure 6.28 Illustration of wall opening present between thermal zones E and D in the test building

In the absence of accurate information about these openings, we assume that each pair of adjacent zones on a floor has a wall opening whose area is 5% of the surface area of the wall that separates these zones. The resistance R_{ij} associated with the thermal interaction between each such pair of zones $\{i, j\}$ can be computed as

$$R_{ij} = \frac{1}{h_{opening}(0.05A_{ij})}.$$
 (6.53)

Here, $h_{opening}$ is the coefficient for heat transfer between the pair of zones $\{i, j\}$ through the wall opening between them. Its value, based on the properties of the "air wall" element in OpenStudio is 0.06 kW/K-m². The values of these resistances for each adjacent pair of zones in each floor of the test building computed based on (6.53) are shown in Table 6.7

The LTI model obtained in Section 6.3.4 is modified by incorporating coupling terms in in its state transition matrix between the zone temperatures states. The code used to obtain the modified model is provided in Appendix J. Note that even after modification, the resulting LTI model has the structure given by (5.1). The LTI model was then discretized using a sample time of 60 seconds. The OLF-FPM partitioning procedure presented in Section 4.3 was applied based on this model. The OLF calculations were based on the parameters N = 30, $\alpha = 1e_9$ and $\beta = 10^3 e_{11}$ (see nomenclature in Chapter 4). The p-partitions obtained via agglomerative clustering (Algorithm 4.1), are shown in Table 6.8. The resulting optimality robustness trade-off curve is shown in Figure 6.29. The codes used for discretization of the model and application of agglomerative clustering are included in Appendix J.

Pair of zones	Thermal resistance due to wall opening (K/kW)
{C,D}	1.318
{G,D}	2.079
{SR,D}	6.615
{G,SR}	6.615
{G,E}	1.582
{G,F2}	3.164
{G,F1}	3.164
{TCB},{NTCB}	1.683

Table 6.7 Thermal resistances due to wall opening between each pair of adjacent zones oneach floor of the test building



Figure 6.29 Optimality-robustness tradeoff curve for test building (p denotes the number of clusters in a partition).

р	Partition from agglomeration
9	${G} {E} {C} {SR} {F1} {D} {F2} {TCB} {NTCB}$
8	${G} {E} {C} {SR,D} {F1} {F2} {TCB} {NTCB}$
7 (knee 1)	${G,SR,D} {E} {C} {F1} {F2} {TCB} {NTCB}$
6	${G,SR,D,E} {C} {F1} {F2} {TCB} {NTCB}$
5 (knee 2, optimally decentralized partition)	${G,SR,D,E} {C} {F1} {F2} {TCB,NTCB}$
4	$\{G,SR,D,E,F2\}$ $\{C\}$ $\{F1\}$ $\{TCB,NTCB\}$
3	{G,SR,D,E,F2,C} {F1} {TCB,NTCB}
2 (knee 3)	{G,SR,D,E,F2C,F1} {TCB,NTCB}
1	{G,SR,D,E,F2C,F1,TCB,NTCB}

Table 6.8 Partitions using agglomeration for test building

The following observations are made from these results.

- 1. From visual inspection, three knee points are observed as shown in Figure 6.29. These correspond to partitions of size 2, 5 and 7 clusters respectively.
- 2. Knee 2 is more centrally located than the other two knees and is therefore treated as the partition with the most appropriate balance between optimality and robustness. Therefore, we refer to the architecture corresponding to this partition as the *optimally decentralized* architecture.

A proposed explanation for the clusters appearing in the optimally decentralized architecture is presented below based on the building layout described in Section 6.2.1.

1. The zones G, D and E form a closely coupled triplet. This is because these zones are interconnected in a hub and spoke manner with zone G acting as the hub. The zone SR is fully contained within zone D and therefore is expected to be strongly coupled with it. The zones F1 and F2, which also act as spokes connected to the hub G, are not as strongly coupled to zone G as zones D and E. This is because the area of the wall

connecting zones F1 or F2 to G is almost half of the area of the wall connecting zone E or D to G. Hence, it is expected that zones G, D, SR and E form a closely coupled quartet, verified from the fact that they constitute one cluster in the optimally decentralized architecture (see Table 6.8).

- 2. The zones F1 and F2 are expected to be weakly coupled because they lie on two different floors. The internal floor separating them has a significant amount of insulation due to air-gap in the construction layers (Table 6.3). Also, there is no wall opening because these zones are on different floors. Hence it is expected that zones F1 and F2 lie in different clusters in the optimally decentralized architecture, which is verified from Table 6.8.
- 3. Based on the layout shown in Figure 6.4, the zone C is expected to be weakly coupled to the rest of the building. This is because it is connected to the building through a single wall. Note that every other zone is connected to the rest of the building through at least two walls (for example, zone E is connected to zones G and NTCB). Hence, it is expected that zone C be put into a single cluster in the optimally decentralized architecture, which is verified from Table 6.8.
- 4. The zones TCB and NTCB are located entirely in the basement of the building and are therefore expected to be weakly coupled to the other zones in the building. However, the coupling between them is expected to be relatively strong because of a wall with a large surface area, and a proportionally large opening separating them. Hence, it is expected that zones TCB and NTCB constitute a cluster in the optimally decentralized architecture, which is verified from Table 6.8.

6.5 Control design and analysis

In this section, centralized and decentralized model predictive controllers are designed using the LTI model of the building thermal dynamics obtained in Section 6.3. The architectures for decentralized control are based on the results in Section 6.4. The control design is based on the framework presented in Chapter 5. Next, the performance of these controllers is evaluated in simulation on a nonlinear model of the building thermal dynamics, which is derived based on the EnergyPlus model developed in Section 6.2. Specifically, the effect of control architecture on optimality and robustness is investigated.

6.5.1 Control design

Centralized and decentralized output feedback model predictive controllers were designed using the control and observation framework described in Chapter 5. For the benefit of the reader, the underlying steps in the control design are summarized in Figure 6.30. The relevant sections in Chapter 5 are referenced. The codes used to implement each of the steps in Figure 6.30 for the test building are provided in Appendix K.



Figure 6.30 Steps used for cluster level control design

The decentralized controllers designed correspond to the partitions given by p = 2, 5, 7and 9 in Table 6.8. These architectures correspond to the three knee points in Figure 6.29 and also the fully decentralized case where each cluster is a zone. The centralized controller was designed using the same principles as for decentralized control. This is because a centralized architecture can be viewed as a decentralized architecture with only one cluster.

The observers were designed such that in the continuous time domain, their poles were 10 times further left on the real axis than the poles of the open loop model. As discussed in Section 5.3.1, for each cluster *i*, the temperature estimates $\hat{\mathbf{T}}_{\mathbf{z}}^{\mathbf{j}}$ for the zones in the clusters other

than *i* appearing in (5.11) are set to the corresponding set-point temperatures $T_{z,ref}^{j}$. The other parameters used in the control design procedure are listed in Table 6.9 (refer to nomenclature in Chapter 5). The maximum supply air mass flow rates and reheating power which are required to set up the constraints in Section 5.4.2 are shown in Table 6.10.

Parameter	Description	Value
N _p	Prediction horizon	30 samples
N _u	Control horizon	15 samples
α_{i}	Penalty on performance term	$\mathbf{e}_{N_{zi}}$
β_{i}	Penalty on cost term	$10^3 \mathbf{e}_{N_{zi}}$
T _{supp}	Supply air temperature to each zone	12.8 °C
C _{p,air}	Specific heat capacity of air	1.005 kJ/kg-K

Table 6.9 Parameters used in designing controllers for each cluster *i*

Table 6.10 Maximum supply air mass flow rate and reheat power available to each zor

Zone number	Maximum supply air mass	Maximum reheat power
(as per Table 6.2)	flow rate (kg/s)	(kW)
1	12.7	140
2	12.7	140
3	12.7	140
4	4.2	50
5	8.5	70
6	12.7	140
7	8.5	70
8	4.2	50
9	4.2	50

6.5.2 Closed loop performance assessment

6.5.2.1 Nonlinear model for control evaluation

Reduced order LTI models were used for control design in Section 6.5.1, which were based on the assumption that the unmodeled thermal loads acting on the walls and zones were slowly time varying quantities. It is desired to evaluate the closed loop performance on a model which has higher fidelity when compared to the models used for control design. This is accomplished by developing a full order model which uses realistic nonlinear expressions for the thermal disturbances.

The afore-mentioned nonlinear model is obtained from the full order LTI model (5.1), by using the expressions (6.54) to (6.56) to model the thermal disturbances $\mathbf{d_w}$ and $\mathbf{d_z}$ for the walls and zones. These expressions emulate the models used by EnergyPlus to compute these disturbances [97]. Note that the temperature values in these equations should be in Kelvin. The reader is directed to the nomenclature for an explanation of the notations used. For each internal wall surface $i \in \{1, 2, ..., N_w\}$,

$$d_{w,i}(t) = \underbrace{\sigma A_i \sum_{j \in \mathcal{N}_i^w} \mathbf{F}_{i,j} \left(T_{w,j}(t)^4 - T_{w,i}(t)^4 \right)}_{long wave radiation} + \underbrace{\alpha_i A_i q_{swr,i}(t)}_{short wave radiation}.$$
(6.54)

For each external wall surface $i \ (i \in \{1, 2, ..., N_w\})$ facing the ambient,

$$d_{w,i}(t) = \underbrace{\sigma A_i \epsilon_i F_{gnd,i} \left(T_{gnd}(t)^4 - T_{w,i}(t)^4 \right)}_{long wave radiation from ground} + \underbrace{\sigma A_i \epsilon_i F_{sky,i} \left(T_{sky}(t)^4 - T_{w,i}(t)^4 \right)}_{long wave radiation from sky} + \underbrace{\sigma A_i \epsilon_i F_{air,i} \left(T_a(t)^4 - T_{w,i}(t)^4 \right)}_{long wave radiation from air} + \underbrace{\alpha_i A_i q_{swr,i}(t)}_{short wave radiation}$$
(6.55)

There is no thermal disturbance acting on external wall surfaces facing the ground. For each thermal zone, $i \in \{1, 2, ..., N_z\}$,

$$d_{z,i}(t) = \eta_{occ,i}(t)N_{occ,i}Q_{occ}(t) + \eta_{light,i}(t)Q_{light,i} + \eta_{appl,i}(t)Q_{appl,i}.$$
 (6.56)

In the above equations, **F** is a matrix of Script-F factors [97, 99], \mathcal{N}_i^w is the set of all internal wall surfaces which share the same zone as internal wall surface *i*, and $F_{gnd,i}$, $F_{sky,i}$ and $F_{air,i}$ are view factors with respect to the ground, sky and air respectively for the external wall surface *i*. These quantities, along with ϵ_i , α_i , A_i , T_{gnd} , T_{sky} , T_a and $q_{swr,i}$ can be directly obtained from

the EnergyPlus model. A spreadsheet showing the values of these quantities is included in the media accompanying this thesis.

A 24-hour time window which starts at 12:00 AM on June 3 and ends at 12:00 AM on June 4 is used for simulation. The signal T_a obtained from EnergyPlus for the simulation time is plotted in Figure 6.31. The quantities $\eta_{occ,i}$, $N_{occ,i}$, $\eta_{light,i}$, $Q_{light,i}$, $\eta_{appl,i}$ and $Q_{appl,i}$ in (6.56) correspond to nominal values and schedules related to occupancy, lighting and equipment, and their values used in the simulation are the same as described in Section 6.2.4. The temperature T_g of the ground below the floor of the building at all times is set to be 18 °C which corresponds to the value used by EnergyPlus.



Figure 6.31 Ambient temperature obtained from EnergyPlus weather file for the simulation time window

6.5.2.2 Optimality Analysis

The five controllers (1 centralized and 4 decentralized) designed in Section 6.5.1 were implemented on the nonlinear model obtained in Section 6.5.2.1. This was done in accordance with the block diagram shown in Figure 5.4. The associated MATLAB codes are provided in Appendix L and the SIMULINK models are enclosed in the media accompanying the thesis. A 24-hour long simulation was performed based on the settings described in Section 6.5.2.1. The desired set-point temperature for each zone is shown in Figure 6.32, which is obtained from the OpenStudio template called 'Medium_Office_ClgSetp'. The initial temperature of all zones and walls is assumed to be $25 \, {}^{\circ}C$. The ambient, ground and sky temperatures and short wave radiation data required to compute the disturbances in Section 6.5.2.1 were obtained by first

simulating the EnergyPlus model for that day. This data is obtained from EnergyPlus in a spreadsheet which is provided in the media accompanying the thesis.

The zone temperature responses and associated control inputs (heating/cooling provided by HVAC system) corresponding to the various control architectures are shown in Figure 6.33 and Figure 6.42. From these figures, it can be observed that the temperature responses and control inputs signals vary depending on the control architecture.



Figure 6.32 Set-point temperatures for all zones during the simulation time window



Figure 6.33 Evolution of zone temperatures between 12:00 AM to 1:00 AM for centralized architecture



Figure 6.34 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 3



Figure 6.35 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 2



Figure 6.36 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 1



Figure 6.37 Evolution of zone temperatures between 12:00 AM to 1:00 AM for fully decentralized architecture


Figure 6.38 Control inputs between 12:00 AM to 1:00 AM for centralized architecture



Figure 6.39 Control inputs between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 3



Figure 6.40 Control inputs between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 2



Figure 6.41 Control inputs between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 1



Figure 6.42 Control inputs between 12:00 AM to 1:00 AM for fully decentralized architecture

To compare the performance of various control architectures with regard to optimality, the integral J_{opt} defined in (6.57) below is used, which is the continuous time analogue of the system-wide (centralized) objective function that the control was designed to minimize (see (5.41) and the parameters in Table 6.10).

$$J_{opt} = \int_{t=0}^{t=24 \text{ hours}} [\mathbf{u}(t)^T \mathbf{u}(t) + 10^3 \mathbf{e}(t)^T \mathbf{e}(t)] dt, \qquad (6.57)$$

where,

$$\mathbf{e}(t) = \mathbf{T}_{\mathbf{z}}(t) - \mathbf{T}_{\mathbf{z},\mathbf{ref}}(t).$$

A comparison of the performance (optimality) for various control architectures is shown in Table 6.11. It is observed that the deviation in optimality increases with the level of decentralization, as expected.

6.5.2.3 Robustness analysis

We set up a case study to examine the robustness of the controllers designed in Section 6.5.1 to a sensor failure event. It is assumed that the thermostat in zone G has developed a fault such that its temperature reading is 0° C – the assumed lower limit of its sensing range – at all times. Zone G is chosen in this case study because it is connected to the most number of zones in

the building (Figure 6.4). The simulation was re-run as described in Section 6.5.2.2 without changing any other setting. The associated MATLAB codes are provided in Appendix M and the SIMULINK models are enclosed in the media accompanying the thesis.

The zone temperature responses corresponding to the various control architectures are shown in Figure 6.43 to Figure 6.47. The resulting magnitudes of temperature deviations from set-points, i.e. $|T_{z,i}(t) - T_{z,i,ref}(t)|$ for all zones $i \in \{1, 2, ..., 9\}$ in the building, at t = 1 hour, for the five control architectures are shown in Figure 6.48. Note that the zone numbers indicated correspond to Table 6.2.

% deviation in J_{opt} from centralized **Control architecture** Jopt 6.29×10^{6} Centralized 0.00 6.29×10^{6} Knee 3 partition 0.00 6.85×10^{6} Knee 2 partition 8.90 Knee 1 partition 7.72×10^{6} 22.73 9.62×10^{6} Fully decentralized 52.94

Table 6.11 Optimality analysis for test building under various control architectures



Figure 6.43 Evolution of zone temperatures between 12:00 AM to 1:00 AM for centralized architecture in the event of sensor failure in zone 1



Figure 6.44 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 3 in the event of sensor failure in zone 1



Figure 6.45 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 2 in the event of sensor failure in zone 1



Figure 6.46 Evolution of zone temperatures between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 1 in the event of sensor failure in zone 1



Figure 6.47 Evolution of zone temperatures between 12:00 AM to 1:00 AM for fully decentralized architecture in the event of sensor failure in zone 1



Figure 6.48 Regulation errors evaluated at the end of 1 hour for all zones in the building under various control architectures in the event of sensor failure in zone 1 (zone numbers indicated correspond to Table 6.2)

From Figure 6.43 to Figure 6.48, it is observed that large deviations from the set-point temperature result in zone 1 in each of the five control architectures. However, as seen in Figure 6.47, set-point temperatures are achieved in all other zones in the building in case of fully decentralized control. Table 6.12 shows the number of affected zones – where the temperatures do not achieve the set-point – and the corresponding fraction of building volume affected, obtained on the basis of data in Figure 6.48 for each of the control architectures considered. The spreadsheet used for computation of fraction of building volume affected is enclosed in the media accompanying the thesis. The clusters that constitute each control architecture, as obtained

from Table 6.8, are also shown in Table 6.12 for reference. As expected, in each case it is observed from Table 6.12 that the number of affected zones matches with the size of the cluster containing zone 1 where the fault originates.

A thermodynamic explanation of the fault propagation phenomenon is as follows. Since the sensor in zone 1 records an incorrect value of 0° C, the controller corresponding to the cluster containing zone 1 dictates the HVAC system to overheat this zone at it maximum allowable heating capacity (Figure 6.49). This controller also overheats other zones in the cluster (see Figure 6.50 and Figure 6.51) – at their corresponding maximum allowable heating capacities because the model for the thermal dynamics of the cluster incorrectly predicts a significant loss of thermal energy from these zones to zone 1 which is assumed to be at 0°C. Since the models used for other clusters do not use information from the sensors of this cluster, they are insulated from the sensor fault in zone 1 (verified from Figure 6.52).

Control	Number of	% building	
architecture	affected zones	volume affected	Clusters
Centralized	9	100.00	{1,2,3,4,5,6,7,8,9}
Knee 3	6	79.25	{1,2,3,4,5,6,7} {8,9}
Knee 2	4	45.30	$\{1,2,4,6\}$ $\{3\}$ $\{5\}$ $\{7\}$ $\{8,9\}$
Knee 1	1	29.79	${2}{3}{1,4,6}{5}{7}{8}{9}$
Fully	1	12.94	{1}{2}{3}{4}{5}{6}{7}{8}{9}
decentralized			

Table 6.12 Closed loop robustness analysis for test building for various controlarchitectures in the event of sensor failure in zone 1



Figure 6.49 Control input for zone 1 between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 3 in the event of sensor failure in zone 1



Figure 6.50 Control input for zone 4 (lying in same cluster as zone 3) between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 3 in the event of sensor failure in zone 1



Figure 6.51 Control input for zone 6 (lying in same cluster as zone 1) between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 3 in the event of sensor failure in zone 1



Figure 6.52 Control input for zone 2 (in different cluster from zone 1) between 12:00 AM to 1:00 AM for decentralized architecture corresponding to knee 3 in the event of sensor failure in zone 1

6.5.2.4 Discussion

As seen from Table 6.12, less than half of the building volume is affected in case of the optimally decentralized partition (knee 2) in the event of sensor failure in zone 1. Also, as previously observed in Table 6.11, the deviation in optimality from centralized control in a situation without failures is less than 10%. Therefore, knee 2 appears to provide an appropriate trade-off between optimality and robustness objectives. This verifies the observation from Figure 6.29 that it is more centrally located on the optimality-robustness trade-off curve than other knees.

6.6 Concluding remarks

The two-step process of control architecture selection and control design presented in the previous chapters in the thesis was successfully applied in simulation on a real world building model. The optimality and robustness trends as were quantitatively investigated as a function of the degree of decentralization. Therefore this chapter demonstrates the applicability of the tools developed in the thesis and can be used by the interested reader to implement them on other building systems. The associated MATLAB codes, SIMULINK models and spreadsheets which are referenced in the chapter should be modified accordingly.

Chapter 7 Conclusions

In this final chapter, the work presented in this thesis is summarized in Section 7.1 followed by concluding remarks in Section 7.2. The main contributions of this work and future avenues of research are discussed in Sections 7.3 and 7.4.

7.1 Summary

A chapter-wise summary of this thesis is presented below:

- 1. Chapter 1 motivates the problem of building thermal control and describes the research objectives of this work. In particular, it motivates the need for decentralized thermal control of buildings. It also provides a literature survey of the area of building thermal control and the tools used in this work.
- The physical aspects of building thermal control were presented in chapter 2. This included a description of underlying energy management systems, and sensing, actuation and control infrastructure.
- 3. Chapter 3 presented mathematical details of centralized and decentralized control architectures, which are used as preliminaries in the development of methodologies for partitioning a building for decentralized control. The objective functions, models and optimization frameworks corresponding to both centralized and decentralized architectures are described in detail.
- Two methodologies CLF-MCS and OLF-FPM were developed in Chapter 4 to partitioning a building for decentralized control. In each of these methodologies, appropriate optimality and robustness metrics were developed and optimality-

robustness trade-off curves were generated to decide a control architecture which provides a satisfactory balance between optimality and robustness. The metrics were based on the results of Chapter 3 and were tested on simple examples in simulation.

- 5. Chapter 5 considers the design of decentralized controllers for any partition of a building. Reduced order, observable models were developed for the thermal dynamics in a cluster. An observation framework which used these models to estimate both known states and disturbances was proposed. A state feedback decentralized model predictive framework was then developed which uses the estimates provided by the observer to minimize an objective function subject to physical constraints.
- 6. A real world simulation study was presented in Chapter 6 to demonstrate the applicability of the tools developed in this work for control architecture selection and control design. Both optimality and robustness analysis was performed to compare the closed loop performance of various control architectures.

7.2 Conclusions

The important conclusions from this work are as follows:

- The thermal control of a complex interconnected system such as a building has multiple underlying objectives, most importantly occupant comfort, energy efficiency, robustness to faults and scalability.
- 2. The control architecture affects the extent to which these objectives are achieved. In particular, a fundamental tradeoff between optimality (occupant comfort and energy efficiency) and robustness (fault resilience) exists with respect to the degree of decentralization. Increase in the degree of decentralization results in improvements in robustness at the cost of optimality.
- 3. Two key challenges were identified with regard to the problem of determination of a control architecture that appropriately balances optimality and robustness requirements. Firstly, appropriate metrics are needed to quantify optimality and robustness. Secondly, the problem of partitioning a building into clusters for decentralized control is inherently computationally complex due to its combinatorial nature.

- 4. A CLF-MCS approach which used coupling loss factor (CLF) and mean cluster size (MCS) as heuristically defined optimality and robustness metrics was proposed. It used a divisive, stage-by-stage partitioning method to generate a family of partitions represented on an optimality-robustness trade-off curve. Its application to simulated examples revealed that the control architectures obtained were in sync with physical intuition.
- 5. An OLF-FPM approach was also proposed which used analytically derived optimality and robustness metrics – optimality loss factor (OLF) and fault propagation metric (FPM). It used an agglomerative clustering approach to generate a family of partitions represented on an optimality-robustness trade-off curve. Similar to the CLF-MCS approach, its application to simulated examples revealed that the control architectures obtained were in sync with physical intuition. It was concluded that the OLF-FPM approach was an improvement over the CLF-MCS approach because the metrics were analytically derived and the complexity of the partitioning procedure was only cubic in the number of zones, as opposed to the exponential complexity associated with the CLF-MCS approach.
- 6. The unavailability of measurements for wall temperature states and thermal disturbances was identified as a key challenge in the design of decentralized controllers, once a control architecture has been determined. A model reduction framework was proposed, which after a suitable state transformation resulted in an observable representation of the cluster level thermal dynamics. This allowed the design of extended state observers to estimate both unknown disturbances and states, which in turn allowed the design of output feedback decentralized controllers for thermal comfort. A model predictive framework with the ability to handle constraints was used for control design.
- 7. Identification of a simplified linear time invariant (LTI) model for a real world test building from its EnergyPlus model was investigated. It was observed that zone level identification using a standard least squares identification approach resulted in physically untenable parameters, potentially due to over-parameterization. This issue was addressed using a modified least squares identification framework which used

fewer parameters. Use of least squares identification approaches for wall level identification resulted in inaccurate and unreliable parameter estimates, potentially due to over-parameterization and mismatch between the order of the identified model and the order of the model used in EnergyPlus. Therefore, direct computation of wall level parameters from construction layer properties was used as an alternative to least squares identification.

8. The application of the OLF-FPM approach on the LTI model of the real world test building resulted in a decentralized architecture which was physically explained on the basis of layout and construction properties of the building. Closed loop evaluation of this architecture on a nonlinear model of the building thermal dynamics verified that it provides a satisfactory tradeoff between optimality and robustness. The simulations also demonstrated and quantified the fundamental tradeoff between optimality and robustness that exists as a function of the degree of decentralization.

7.3 Contributions

Intelligent energy management in buildings is important due to the large scale impacts of the building sector on the economic and environmental aspects of energy. In this context, efficient thermal control is especially important because of the relatively significant contribution of space heating and cooling to the end use energy consumption of buildings. This thesis makes some important contributions to the problem of building thermal control, which are listed below.

- The role of control architecture in achieving the objectives associated with the thermal control of buildings has not been properly investigated in literature. This work contributes to the area of building thermal control by motivating the incorporation of control architecture as an important dimension to be considered in control design. In this context, it specifically investigates the impact of decentralization of the control architecture on the attainment of optimality and robustness objectives.
- A specific contribution of this work is the development of appropriate metrics to quantify the optimality and robustness attributes of any decentralized architecture for the thermal control of a building. Both heuristic (CLF and MCS) and analytically derived (OLF and FPM) metrics were presented.

- 3. Another contribution of this work is the adoption of existing tools and concepts from parallel technological fields, which have previously not been applied to area of building thermal control. In particular, the concept of agglomerative clustering was successfully employed to address the computational complexity concerns in the problem of control architecture determination.
- 4. In addition to providing methodologies for control architecture selection, this work also provides a methodology for control design based on the control architectures selected. It proposes a control design framework which addresses practical issues such as unavailability of certain states and disturbances and presence of physical constraints.
- 5. Lastly, this thesis complements the theoretical frameworks proposed for control architecture selection and control design by showing their applicability on a real world building example in simulation.

7.4 Future extensions

We identify the following areas of future research to build upon the work presented in this thesis.

7.4.1 Incorporation of HVAC system

The scope of this thesis was limited to consideration of the building side dynamics and control. The control variables considered correspond to the energy transfer rates (heating/cooling) provided by the HVAC system to the zones in the building. However, the HVAC itself was not considered.

To manipulate the heating or cooling provided to the zones for building thermal control, appropriate actuators in the HVAC system – such as dampers and reheaters in a VAV system (Chapter 2) – need to be adjusted. Therefore, for practical implementation of closed loop control, the HVAC system dynamics should be included. This can be achieved by modifying the control design framework in Chapter 5 by including appropriate actuator dynamics.

7.4.2 Experimental investigation

Implementation of the tools proposed in this work on a real building system can be undertaken. Such experimental studies would serve to complement the simulation studies presented in Chapter 6, and provide additional validation of the tools for control architecture selection and control design presented in this thesis.

As described in Section 7.4.1, the HVAC system dynamics would have to be incorporated in the control design framework for experimental implementation. This may necessitate the consideration of other practical aspects such as actuator limitations and slew rates in addition to the constraints described in Chapter 5.

Besides experimental validation, co-simulation approaches, which allow controllers based in MATLAB to interface directly with the higher fidelity EnergyPlus models can also be explored. This can be enabled by the use of appropriate platforms such as the Building Control Virtual Test Bed (BCVTB) [100].

7.4.3 Extension to other applications

Although the proposed tools in this thesis are developed specifically in the context of thermal control of buildings, they can also be potentially applied to other large scale energy management applications such as data center cooling, district heating and cooling for campuses, and distributed refrigeration systems for supermarkets. Moreover, the modularity of the decentralized framework allows for extensions to other energy efficiency domains apart from thermal. This includes electrical grid based systems having generation, distribution, consumption and recovery.

References

- [1] The White House. Energy, Climate Change and our Environment. [Online]. Available: http://www.whitehouse.gov/ energy
- [2] K. Annan. Toward a sustainable future. Environment: Science and Policy for Sustainable Development, 44 (7): 10–15, 2002.
- [3] EIA. Annual Energy Outlook 2011. [Online]. Available: http://www.eia.gov/forecasts/ aeo/data.cfm
- [4] EIA. 2003 Commercial Buildings Energy Consumption Survey. [Online]. Available: http:// www.eia .gov/emeu/cbecs /contents.html
- [5] EIA. 2009 Residential Energy Consumption Survey. [Online]. Available: http://www.eia. gov/consumption/residential/reports/2009overview.cfm
- [6] G. Holness. Sustaining Our Future By Rebuilding Our Past: Energy Efficiency in Existing Buildings – Our Greatest Opportunity for a Sustainable Future. ASHRAE J, (51(8), 16–21, 2009.
- [7] H. H. Mathews, C. P. Botha, D. C. Arndt, and A. Malan. HVAC control strategies to enhance comfort and minimise energy usage. *Energy and buildings*, 33(8):853-863, 2001.
- [8] F. Oldewurtel, et al. Energy efficient building climate control using stochastic model predictive control and weather predictions *Proceedings of the American Control Conference* (ACC), 5100--5105, 2010.
- [9] S. Wang and Z. Ma. Supervisory and optimal control of building HVAC systems: A review. *HVAC&R Res.*, 14(107):3-32, 2008.
- [10] A. Alleyne, et al. Final report. NSF CMMI Woskshop on Building Systems, University of Illinois at Urbana-Champaign, 2010. [Online]. Available: http://arg.mechse.illinois.edu/ index.php?id=1093|Publications
- [11] J. Morris, D. Kroening, and P. Koopman. Fault tolerance tradeoffs in moving from decentralized to centralized embedded systems. *International Conference on Dependable Systems and Networks*, 377-386, 2004.

- [12] P. J. Campo and M. Morari. Achievable closed-loop properties of systems under decentralized control: Conditions involving the steady-state gain. *IEEE Transactions on Automatic Control*, 39(5):932-943, 1994.
- [13] D. D. Šiljak, Decentralized control of complex systems. Academic Press, 1991.
- [14] J. T. Feddema, C. Lewis and D. A. Schoenwald. Decentralized control of cooperative robotic vehicles: theory and application. *Robotics and Automation, IEEE Transactions on*, 18(5):852-864, 2002.
- [15] T. Huntsberger, et al. CAMPOUT: A control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 33(5):550-559, 2003.
- [16] J. C. Russell, Decentralized control of satellite formations. International Journal of Robust and Nonlinear Control, 12(2-3):141-161, 2002.
- [17] R. W. Brennan, S. Balasubramanian and D. H. Norrie. Dynamic control architecture for metamorphic control of advanced manufacturing systems. *Proceedings of SPIE*, 1997.
- [18] D. M. Dilts, N. P. Boyd, and H. H. Whorms. The evolution of control architectures for automated manufacturing systems. *Journal of manufacturing systems*, 10(1):79-93, 1991.
- [19] V. Chandan, S Mishra, and A. Alleyne, Predictive control of complex hydronic systems. *Proceedings of the American Control Conference (ACC)*, pages 5112-5117, 2010.
- [20] P. Davidsson and M. Boman. Distributed monitoring and control of office buildings by embedded agents. *Information Sciences*, 171(4):293-307, 2005.
- [21] K. L. Moore, T. Vincent, F. Lashhab and C. Liu. Dynamic Consensus Networks with Application to the Analysis of Building Thermal Processes. *Proceedings of the IFAC World Congress*, 3078-3083, 2011.
- [22] S. Sharples, V. Callaghan and G. Clarke. A multi-agent architecture for intelligent building sensing and control. *Sensor Review*, 19(2):135-140, 1999.
- [23] M. Zaheer-Uddin, R. V. Patel and S. A. K. Al-Assadi. Design of decentralized robust controllers for multizone space heating systems. *IEEE Transactions on Control Systems Technology*, 1(4):246-261, 1993.
- [24] A. Aswani, et al. Identifying models of HVAC systems using semiparametric regression. In Proceedings of the American Control Conference, 2012.

- [25] J. Široký, et al. Experimental analysis of model predictive control for an energy efficient building heating system. *Applied Energy* 88(9): 3079-3087, 2011.
- [26] S. Goyal, H. Ingley and P. Barooah. Zone-level control algorithms based on occupancy information for energy efficient buildings. In *American Control Conference*, 2012.
- [27] U.S. Department of Energy, EnergyPlus energy simulation software. http://apps1.eere. energy.gov/buildings/energyplus/
- [28] Ma, Jingran, S. J. Qin and T. Salsbury. Model predictive control of building energy systems with balanced model reduction. In *American Control Conference (ACC)*, 3681-3686, 2012.
- [29] K. Deng, et al. Building thermal model reduction via aggregation of states. In *American Control Conference (ACC)*, 5118-5123, 2010
- [30] The American Society of Heating, Refrigeration and Air-Conditioning Engineers. http:// www.ashrae.org/.
- [31] The Energy Information Administration. http://www.eia.doe.gov/.
- [32] Europe's Energy Portal. http://www.energy.eu/.
- [33] The U.S. Green Building Council. http://www.usgbc.org/.
- [34] facilitiesnet. http://www.facilitiesnet.com/.
- [35] EERE: Building Technologies Program. http://www1.eere.energy.gov/buildings/.
- [36] Energy Star. http://www.energystar.gov/.
- [37] The Green Grid. http://www.thegreengrid.org/.
- [38] F. Lorenz and G. Masy, M'ethode d''evaluation de l''economie d''energie apport'ee par l'intermittence de chauffage dans les b'atiments, Traitement par differences finies d'un model a deux constantes de temps, *Report No. GM820130-01. Faculte des Sciences Appliquees, University de Liege, Liege, Belgium,* 1982 (in French).
- [39] G. Hudson and C. Underwood. A simple building modeling procedure for MATLAB/ SIMULINK. in Proc. Int. Build. Perform. Simul., Conf., 1-7. 1999
- [40] M. Gouda, S. Danaher and C. Underwood. Building thermal model reduction using nonlinear constrained optimization. *Build. Environ.*, 37(12), 1255-1265, 2002.
- [41] S. Wang and X. Xu. Simplified building model for transient thermal performance estimation using GA-based parameter identification. *International journal of thermal sciences* 45(4):419-432, 2006.

[42] A. P. R. González, M. E. Eames and D. A. Coley. Lumped Parameter Models for Building Thermal Modelling: An Analytic approach to simplifying complex multi-layered constructions.

http://ceae.colorado.edu/~zhiqiang/COBEE12/PDF/Topic%2011/T11 49 v2.pdf

- [43] P. Radecki and B. Hencey. Online building thermal parameter estimation via Unscented Kalman Filtering. In *American Control Conference (ACC)*, 3056-3062, 2012.
- [44] S. Prívara et al. Modeling and identification of a large multi-zone office building. In *Control Applications (CCA), IEEE International Conference on*, 55-60, 2011.
- [45] S. Goyal and P. Barooah. A method for model-reduction of nonlinear building thermal dynamics. In *American Control Conference (ACC)*,2077-2082, 2011.
- [46] US Department of Energy, Whole Building Analysis: Energy Simulation. http://apps1. eere.energy.gov/buildings/tools_directory/subjects.cfm/pagename=subjects/pagename_menu =whole_building_analysis/pagename_submenu=energy_simulation
- [47] ESP-r. http://www.esru.strath.ac.uk/Programs/ESP-r.htm
- [48] Hamlab. http://archbps1.campus.tue.nl/bpswiki/index.php/Hamlab
- [49] Building Sim. http://www.buildingsim.com/
- [50] Building Simulation with SMILE. http://english.dezentral.de/warp.html
- [51] E. F. Camacho, et al. *Model predictive control*. Springer, 1999.
- [52] D. Q. Mayne, et al. Constrained model predictive control: Stability and optimality. *Automatica* 36(6): 789-814, 2000.
- [53] G. P. Henze, et al. Impact of forecasting accuracy onpredictive optimal control of active and passive building thermal storage inventory. *HVAC & R Research*, 10(2):153-178, 2004.
- [54] G. P Henze, et al.. Experimental analysis of model-based predictive optimal control for active and passive building thermal storage inventory. *International Journal of Heating Ventilating Air Conditioning and Refrigerating Research*, 11(2):189-214, 2005.
- [55] S. Prívara, et. al. Role of MPC in Building Climate Control. Computer Aided Chemical Engineering, 29:728-732, 2011.
- [56] T. X. Nghiem and G. J. Pappas. Receding-horizon supervisory control of green buildings. In American Control Conference (ACC), 4416-4421, 2011.

- [57] Y. Ma, G. Anderson and F. Borrelli. A distributed predictive control approach to building temperature regulation. In *American Control Conference (ACC)*, 2089-2094. 2011.
- [58] P. Moroşan, et al. Building temperature regulation using a distributed model predictive control. *Energy and Buildings* (42)9:1445-1452, 2010.
- [59] M. Maasoumy, and A Sangiovanni-Vincentelli. Optimal control of building hvac systems in the presence of imperfect predictions. In *Dynamic Systems and Control Conference*, 2012.
- [60] K. Deng, P. Barooah and P. G. Mehta. Mean-field control for energy efficient buildings. In American Control Conference (ACC), 3044-3049, 2012.
- [61] Snoonian, D, Smart buildings. Spectrum, IEEE, 40(8): 18-23, 2003.
- [62] Levermore, G. Building Energy Management Systems: An Application to Heating, Natural Ventilation, Lighting and Occupant Satisfaction. Taylor & Francis, 2000.
- [63] McDowall, R. Fundamentals of HVAC systems. Elsevier Science Limited, 2006.
- [64] Commercial Lighting Tax Deduction, Introduction to lighting controls. http://www. lightingtaxdeduction.org/ technologies/intro-lighting-controls.html
- [65] Lighting Controls Association. http://lightingcontrolsassociation.org/
- [66] Galaxias, Building security solutions. http://www.galaxias.in/?page_id=107
- [67] Schneider Electric, Integrated security solutions. http://www.schneider-electric.com/ solutions/ww/en/sol/ 4871515-integrated-security-solutions?domain=4664816
- [68] Drexel University, VAV systems (online lecture notes). http://www.pages.drexel.edu/ ~ea38/AE390/A5 /products.htm
- [69] Madison Gas and Electric Company, Economizers. http://www.mge.com/business/saving/ BEA / escrc 0013000000DP22YAAT-2 BEA1 PA PA Ventilation PA-08.html
- [70] Chandan, Vikas. Modeling and control of hydronic building HVAC systems., MS Thesis (2010). http://arg.mechse.illinois.edu/index.php?id=1144|ms
- [71] Handbook, A. S. H. R. A. E. HVAC systems and equipment. *American Society of Heating, Refrigerating, and Air Conditioning Engineers, Atlanta, GA* (2012).
- [72] Emerson, Programmable Universal Staging Thermostats. http://www.emersonclimate.com/ en-US/ products/thermostats/Pages/programmable_universal_thermostats.aspx
- [73] ASIC/1-6000 VAV Controller with Integrated Actuator, http://www.asicontrols.com/ products/programmed/asic_1_6000/.

- [74] Salsbury, T. A survey of control technologies in the building automation industry. In 16th IFAC World Congress. 2005.
- [75] Scattolini, R. Architectures for distributed and hierarchical model predictive control-a review. *Journal of Process Control.* 19(5): 723-731, 2009.
- [76] Soyguder, S., Karakose, M., & Alli, H. Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system. *Expert Systems with Applications*, 36(3 PART 1): 4566-4573, 2009.
- [77] Athienitis, A. K., & Chen, T. Y. Experimental and theoretical investigation of floor heating with thermal storage. In *the 1993 Winter Meeting of ASHRAE Transactions*. *Part 1*,1049-1057. 1993.
- [78] Maasoumy, M., Pinto, A., & Sangiovanni-Vincentelli, A. Model-based hierarchical optimal control design for HVAC systems. In *Dynamic System Control Conference (DSCC)*, 2011.
- [79] Ma Y. et al., Predictive control for energy efficient buildings with thermal storage: Modeling, stimulation, and experiments. *Control Systems, IEEE*, 32 (1):44-64, 2012.
- [80] Prívara, S., Široký, J., Ferkl, L., and Cigler, J. Model predictive control of a building heating system: The first experience. *Energy and Buildings*, 43(2): 564-572. 2011.
- [81] Huang, G., Wang, S., and Xu, S. A robust model predictive control strategy for improving the control performance of air-conditioning systems. *Energy Convers. Manage.*, 50(10): 2650-2658, 2009.
- [82] Boyd, S.P. and Vandenberghe, L. Convex optimization. Cambridge Univ Pr, 2004.
- [83] Skogestad, S. and Postlethwaite, I. Multivariable feedback control: analysis and design. Wiley New York, 1996.
- [84] Rota, G.C. The number of partitions of a set. *American Mathematical Monthly*, 71(5):498--504, 1964.
- [85] Stoer, M. and Wagner, F. A simple min-cut algorithm. Journal of the ACM (JACM), 44(4):585--591, 1997.
- [86] Chidananda Gowda, G. and others. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105--112, 1978.
- [87] Eisen, M.B. and Spellman, P.T. and Brown, P.O. and Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of*

Sciences of the United States of America, 95(25):14863, 1998.

- [88] Jain, A.K. and Murty, M.N. and Flynn, P.J. Data clustering: a review. ACM computing surveys (CSUR), 31(3):264--323, 1999.
- [89] S. Li, et al., "Generalized extended state observer based control for systems with mismatched uncertainties," *Industrial Electronics, IEEE Transactions on*, vol.59, no. 12, 4792-4802.
- [90] R. Miklosovic, A. Radke, and Z. Gao. "Discrete implementation and generalization of the extended state observer," in *Proc. Amer. Control Conf.*, 2006.
- [91] R. Horn, and C. Johnson. Matrix Analysis, Cambridge University Press, 1985
- [92] Matlab Optimization Toolbox. http://www.mathworks.com/products/optimization/
- [93] Glassdoor. http://www.glassdoor.com/Photos/Siemens-Corporate-Research-Office-Photos-E220560.htm
- [94] Google Sketchup. http://www.sketchup.com/intl/en/download/index2.html
- [95] National Renewable Energy Laboratory, Openstudio. http://open studio.nrel.gov/
- [96] EnergyPlus weather data. http://apps1.eere.energy.gov/buildings/energyplus/cfm/weather data .cfm
- [97] EnergyPlus engineering documentation. http://apps1.eere.energy.gov/buildings/energyplus/ pdfs/engineeringreference.pdf
- [98] MATLAB, fmincon documentation. http://www.mathworks.com/help/optim/ug/fmincon. html
- [99] H.C. Hottel, and A.F. Sarofim, *Radiative transfer*. McGraw-Hill New York, 1967.
- [100] Lawrence Berkeley National Laboratory, Building Controls Virtual Test Bed http:// simulationresearch.lbl.gov/ bcvtb

Appendix A

Codes for 12-zone building example in section 4.2.8

To obtain the results presented in this section, the following programs need to be run in the specified sequence:

```
STEP 1: Obtain weighted incidence matrix, and capacitance matrices (see Algorithm 3.1)
clc;
clear all;
N alpha = [];
Rinv ext h in = 1/29.99;
Rinv ext h out = 1/81.08;
Rinv ext v in = 1/36.84;
Rinv ext v out = 1/82.00;
Rinv int h in = 1/21.32;
Rinv int h out = 1/21.32;
Rinv int v in = 1/21.32;
Rinv int v out = 1/21.32;
% Factors rho 1, rho 2 and rho 3 being set to 1. These factors can be set to other values % to
run the various cases in section 4.2.8
rho 1 = 1;
rho 2 = 1;
rho 3 = 1;
% Set 1: Incidence matrix entries for horizontal external walls
count1 = 0;
vect 1 = [1,2,3,4,5,6,7,8,9,10,11,12];
vect 2 = [1,2,3,4,5,6,7,8,9,10,11,12];
for i = 1:length(vect 1)
  N alpha(vect 1(i), vect 2(i)) = Rinv ext h in;
  N alpha(vect 1(i), 13) = Rinv ext h out;
  count1 = count1+1;
end
% Set 2: Incidence matrix entries for vertical external walls
count2 = 0;
vect 1 = [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24];
vect 2 = [1,2,3,4,5,6,7,8,9,10,11,12];
for i = 1:length(vect 1)
  N alpha(vect 1(i), vect 2(i)) = Rinv ext v in;
  N alpha(vect 1(i), 13) = Rinv ext v out;
  count2 = count2+1;
end
```

vect 1 = [37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48];vect 2 = [1,2,5,6,11,12,7,8,9,10,11,12];for i = 1:length(vect_1) N alpha(vect 1(i), vect 2(i)) = Rinv ext v in; N alpha(vect 1(i), 13) = Rinv ext v out; count2 = count2+1;end % Set 3: Incidence matrix entries for horizontal internal walls count3 = 0;vect 1 = [25, 26, 27, 28, 29, 30];vect 2 = [3,4,7,8,9,10];vect 3 = [5,6,9,10,11,12];for i = 1:length(vect 1) N alpha(vect 1(i),vect 2(i)) = Rinv int h in/rho 1; N alpha(vect 1(i),vect 3(i)) = Rinv int h out/rho 1; count3 = count3+1;end % Set 4: Incidence matrix entries for vertical internal walls count4 = 0; vect 1 = [31, 32, 33, 34, 35, 36];vect 2 = [1,3,5,7,9,11];vect_3 = [2,4,6,8,10,12]; % Set 4, subset 1: Symmetrically splitting internal walls for i = 1:length(vect 1) N alpha(vect 1(i),vect 2(i)) = Rinv int V in/rho 3; N alpha(vect 1(i),vect 3(i)) = Rinv int V out/rho 3; count4 = count4+1;end % Set 4, subset 2: Column separating internal walls N alpha(37,1) = Rinv int V in/rho 2; N alpha(37,3) = Rinv int V out/rho 2; N alpha(38,2) = Rinv int V in/rho 2; N alpha(38,4) = Rinv int V out/rho 2; N alpha(39,3) = Rinv int V in/rho 2; N alpha(39,7) = Rinv int V out/rho 2; N alpha(40,4) = Rinv int V in/rho 2; N alpha(40,8) = Rinv int V out/rho 2; N alpha(41,5) = Rinv int V in/rho 2; N alpha(41,9) = Rinv int V out/rho 2; N alpha(42,6) = Rinv int V in/rho 2; N_alpha(42,10) = Rinv_int V out/rho 2; count4 = count4+6;% Capacitance matrices C w = blkdiag(8329.15*eye(count 1), 8329.15*eye(count 2), 4660*eye(count 3), 4660*eye(count 4));

```
C z = blkdiag(250*eye(12));
C cap = blkdiag(C w,C z);
STEP 2: Obtaining matrices A, B, C and D for continuous time state space model (see Algorithm
3.1)
nw = count 1 + count 2 + count 3 + count 4;
nz = 12;
N beta = -N alpha;
N alpha z = N alpha(:,1:nz);
N beta z = N beta(:,1:nz);
S r beta = diag(sum(N beta'));
S c alpha z = diag(sum(N alpha z));
N alpha a = N alpha(:,nz+1:nz+1);
inv C cap = inv(C cap);
% Generation of A, B, C and D matrices denotes using the subscript "bldg."
A bldg = inv C cap*[[S r beta, N alpha z];[-N beta z',-S c alpha z]];
B bldg = inv C cap*[[zeros(nw,nz), zeros(nw,nz), N alpha a];[eye(nz), eye(nz), zeros(nz,1)]];
C bldg = [zeros(nz,nw),eye(nz)];
D bldg = zeros(nz,2*nz+1);
0/0 *************
                       STEP 3: Discretization of model and generation of Hessian matrix
% Part 1: Discretization of continuous state space model, resulting matrices are A fd, %B fd,
and C fd
global N Hess Nu
sys = ss(A \ bldg, B \ bldg, C \ bldg, D \ bldg);
Ts = 600;
sysd = c2d(sys,Ts,'zoh');
A fd = sysd.A;
B f = sysd.B;
C fd = sys.C;
B fd = B f(:,1:9);
% Part 2: Finding unswapped Hessian
N = 24; % Length of prediction horizon
Nu = 12:
for i = 0:N-1
 P(:,:,i+1) = C fd^{*}(A fd)^{i*}B fd;
end
Hess us = [];
for i = 0:N-1
 blah = [];
  for i = 0:N-1
   blah = [blah, (P(:,:,i+1))'*(P(:,:,j+1))];
```

```
end
 Hess us = [Hess us;blah];
end
% Part 3: Swapping to create Hess which is the Hessian matrix in desired form
H s = [];
for i = 1:Nu
 for j = 0:N-1
   H s = [H s;Hess us(j*Nu+i,:)];
 end
end
Hess =[];
for i = 1:Nu
 for i = 0:N-1
   Hess = [Hess, H s(:, j*Nu+i)];
 end
end
STEP 4: Stage by stage combinatorial optimization (see section 4.2.3) and generation of
optimality-robustness tradeoff curve
global N Hess Nu
parentset = struct('values', {[1:1:Nu]});
rcs = 1;
k = 1:
clf vect(k) = 0;
res vect(\mathbf{k}) = rcs;
while rcs > 1/Nu
[childset,clf,mcs] = comboptm(parentset); % call the function "comboptm" to calculate
% the child partition for a given parent partition (stage level combinatorial optimization)
childset.values % displays result (output child partition) of the combinatorial optimization
parentset = childset;
k = k + 1;
clf vect(k) = clf;
mcs vect(k) = mcs;
end
plot(mcs vect, clf vect, '-') %Generates optimality-robustness tradeoff curve
```

MATLAB Function "comboptm" used in the above code is as shown:

```
minweight = 99999;
for l = 1:no parents
  list = parentset(l).values;
  n = length(list);
  for k = 1:n-1
    C = combnk(list,k); % Creates all possible intermediate cluster pairs
    nc = length(C(:,1));
% The following loop compares the intermediate cluster pairs to find the ones with
% smallest ILF
    for i = 1:nc
       innerlist = C(i,:);
       innerweight = weight calculate(innerlist,list)/self weight(list); %ILF (see (4.14))
       if(innerweight<minweight)
          minweight = innerweight;
          minlist = innerlist;
         minlist comp = setdiff(list,minlist);
          minparent = 1;
       end
    end
  end
end
% Create child clusters
childset = struct('values', {});
for l = 1:no parents;
  if l<minparent
    childset(1).values = parentset(1).values;
  end
  if 1 == minparent
    childset(1).values = minlist;
    childset(l+1).values = minlist comp;
  end
  if(l>minparent)
    childset(l+1).values = parentset(l).values;
  end
end
% The remainder of the program computes the CLF and MCS of child clusters
c = no parents + 1;
tot coupling lost = [];
for i = 1:c-1
  tot coupling lost i = [];
  for j = i+1:c
    coupling i j = findcoupling(childset(i).values,childset(j).values);
    if (i == j)
       coupling i j = 0^* coupling i j;
    end
```

```
The MATLAB functions "weight calculate", "self weight" and "findcoupling" used in the
above function are shown below
function result = weight calculate(innerlist,list)
global N Hess Nu
cluster1 = innerlist;
cluster2 = setdiff(list,innerlist);
bout = [];
for i = 1:length(cluster1)
 bin = [];
 for j = 1:length(cluster2)
   bin = [bin, Hess((cluster1(i)-1)*N+1:cluster1(i)*N, (cluster2(j)-1)*N+1:cluster2(j)*N)];
 end
 bout = [bout;bin];
end
result = norm(bout);
function result = self weight(list)
global N Hess Nu
bout = [];
for i = 1:length(list)
 bin = [];
 for j = 1:length(list)
   bin = [bin, Hess((list(i)-1)*N+1:list(i)*N, (list(j)-1)*N+1:list(j)*N)];
 end
 bout = [bout;bin];
end
result = norm(bout);
function result = findcoupling(cluster1,cluster2)
global N Hess Nu
bout = [];
for i = 1:length(cluster1)
```

Appendix B

Codes for 9-zone building example in section 4.2.9

To obtain the results presented in this section, the following programs need to be run in the specified sequence:

```
STEP 1: Obtain weighted incidence matrix, and capacitance matrices (see Algorithm 3.1)
% 9 zone building
clc
clear all
N alpha = [];
rho = 1; % Factor rho is set here
rat1 = rho;
rat2 = 1/rho;
% SET 1: Incidence matrix entries for horizontal external walls
count 1 = 0;
for i = 1:3
  count 1 = \text{count } 1+1;
  currentrow = zeros(1,10);
  currentrow(i) = 1/29.99;
  currentrow(10) = 1/81.08;
  N alpha = [N alpha; currentrow];
end
for i = 7:9
  currentrow = zeros(1,10);
  currentrow(i) = 1/29.99;
  currentrow(10) = 1/81.08;
  count 1 = \text{count } 1+1;
  N alpha = [N alpha; currentrow];
end
% SET 2: Incidence matrix entries for vertical external walls
count 2 = 0;
for i = 1:3:7
  currentrow = zeros(1,10);
  currentrow(i) = 1/36.84;
  currentrow(10) = 1/82.00;
  count 2 = \text{count } 2+1;
  N alpha = [N alpha; currentrow];
end
for i = 1:9
  currentrow = zeros(1,10);
```

```
currentrow(i) = 1/36.84;
  currentrow(10) = 1/82.00;
  count 2 = \text{count } 2+1;
 N alpha = [N alpha; currentrow];
end
for i = 3:3:9
  currentrow = zeros(1,10);
  currentrow(i) = 1/36.84;
  currentrow(10) = 1/82.00;
  count 2 = \text{count } 2+1;
  N alpha = [N alpha; currentrow];
end
for i = 1:9
  currentrow = zeros(1,10);
  currentrow(i) = 1/36.84;
  currentrow(10) = 1/82.00;
  count 2 = \text{count } 2+1;
  N alpha = [N alpha;currentrow]:
end
%SET 3: Incidence matrix entries for horizontal internal walls
count 3 = 0;
for i = 1:6
currentrow = zeros(1,10);
  currentrow(i) = 1/(rat1*21.32);
  currentrow(i+3) = 1/(rat1*21.32);
  count 3 = \text{count } 3+1;
  N alpha = [N alpha; currentrow];
end
%SET 4: Incidence matrix entries for vertical internal walls
count 4 = 0;
for i = 1:2
  for j = i:3:i+6
    currentrow = zeros(1,10);
    currentrow(j) = 1/(rat2*21.32);
    currentrow(j+1) = 1/(rat2*21.32);
    count 4 = \text{count } 4+1;
    N alpha = [N alpha; currentrow];
  end
end
%Capacitance matrices
C w = blkdiag(8329.15*eye(count 1), 8329.15*eye(count 2), 4660*eye(count 3),
4660*eye(count 4));
C z = blkdiag(250*eye(9));
C cap = blkdiag(C w,C z);
```

STEP 2: Obtaining matrices A, B, C and D for continuous time state space model (see Algorithm 3.1)

```
nw = count 1 + count 2 + count 3 + count 4;
nz = 9;
N beta = -N alpha;
N alpha z = N alpha(:,1:nz);
N beta z = N beta(:,1:nz);
S r beta = diag(sum(N beta'));
S c alpha z = diag(sum(N alpha z));
N alpha a = N alpha(:,nz+1:nz+1);
inv C cap = inv(C cap);
% Generation of A, B, C and D matrices denotes using the subscript "bldg."
A bldg = inv C cap*[[S r beta, N alpha z];[-N beta z',-S c alpha z]];
B bldg = inv C cap*[[zeros(nw,nz),zeros(nw,nz),N alpha a];[eye(nz),eye(nz),zeros(nz,1)]];
C bldg = [zeros(nz,nw),eye(nz)];
D bldg = zeros(nz,2*nz+1);
STEP 3: Discretization of model and generation of Hessian matrix
% Part 1: Discretization of continuous state space model, resulting matrices are A fd, %B fd,
and C fd
global N Hess Nu
sys = ss(A bldg, B bldg, C bldg, D bldg);
Ts = 600;
sysd = c2d(sys, Ts, 'zoh');
A fd = svsd.A;
B f = sysd.B;
C fd = sys.C;
B fd = B f(:,1:9);
N = 24; %Prediction horizon
Nu = 9:
%Finding unswapped Hessian
for i = 0:N-1
 P(:,:,i+1) = C fd^{*}(A fd)^{i*}B fd;
end
Hess us = [];
for i = 0:N-1
 blah = [];
 for j = 0:N-1
   blah = [blah, (P(:,:,i+1))'*(P(:,:,j+1))];
 end
 Hess us = [Hess us;blah];
end
```

```
%Finding swapped Hessian
H s = [];
for i = 1:Nu
 for j = 0:N-1
   H s = [H s;Hess us(j*Nu+i,:)];
 end
end
Hess =[];
for i = 1:Nu
 for j = 0:N-1
   Hess = [Hess, H s(:, j*Nu+i)];
 end
end
STEP 4: Stage by stage combinatorial optimization (see section 4.2.3) and generation of
optimality-robustness tradeoff curve
global N Hess Nu
parentset = struct('values', {[1:1:Nu]});
rcs = 1:
k = 1;
clf vect(k) = 0;
res vect(k) = rcs;
while rcs > 1/Nu
[childset,clf,mcs] = comboptm(parentset); % call the function "comboptm" to calculate
% the child partition for a given parent partition (stage level combinatorial optimization)
% To perform optimization using the MINCUT method instead, replace "comboptm" with
"comboptm maxcut"
childset.values % displays result (output child partition) of the combinatorial optimization
parentset = childset;
k = k + 1:
clf vect(k) = clf;
mcs vect(k) = mcs;
end
plot(mcs_vect,clf_vect,'-') %Generates optimality-robustness tradeoff curve
```

The MATLAB functions "comboptm" and "comboptm_maxcut" used in the above codes are as shown:

```
minweight = 99999;
for l = 1:no parents
  list = parentset(l).values;
  n = length(list);
  for k = 1:n-1
    C = combnk(list,k); % Creates all possible intermediate cluster pairs
    nc = length(C(:,1));
% The following loop compares the intermediate cluster pairs to find the ones with
% smallest ILF
    for i = 1:nc
       innerlist = C(i,:);
       innerweight = weight calculate(innerlist,list)/self weight(list); %ILF (see (4.14))
       if(innerweight<minweight)
          minweight = innerweight;
          minlist = innerlist;
         minlist comp = setdiff(list,minlist);
          minparent = 1;
       end
    end
  end
end
% Create child clusters
childset = struct('values', {});
for l = 1:no parents;
  if l<minparent
    childset(1).values = parentset(1).values;
  end
  if 1 == minparent
    childset(1).values = minlist;
    childset(l+1).values = minlist comp;
  end
  if(l>minparent)
    childset(l+1).values = parentset(l).values;
  end
end
% The remainder of the program computes the CLF and MCS of child clusters
c = no parents + 1;
tot coupling lost = [];
for i = 1:c-1
  tot coupling lost i = [];
  for j = i+1:c
    coupling i j = findcoupling(childset(i).values,childset(j).values);
    if (i == j)
       coupling i j = 0^* coupling i j;
    end
```
```
tot coupling lost i = [tot coupling lost i;coupling i j];
  end
  tot coupling lost = [tot coupling lost;tot coupling lost i];
end
clf1 = (norm(tot coupling lost))/norm(Hess);
mcs = 1/c;
function [childset,clf1,mcs] = comboptm maxcut(parentset)
global N Hess Nu
no parents = length(parentset);
minweight = 99999;
for l = 1:no parents
  list = parentset(1).values;
  n = length(list);
  if(n>1)
  %Create Hp
  Hp = [];
  for i = 1:n
    Hp row = [];
    for j = 1:n
      Hp row = [Hp row, Hess((list(i)-1)*N+1:list(i)*N, (list(j)-1)*N+1:list(j)*N)];
    end
    Hp = [Hp; Hp row];
  end
  for i = 1:n
  Hp((i-1)*N+1:i*N,(i-1)*N+1:i*N) = zeros(N,N);
  end
    %Create Q
  Q = zeros(N*n,n);
  for i = 1:n
    Q((i-1)*N+1:i*N,i) = ones(N,1);
  end
  %Create weighting matrix and run MINCUT
  W = O'*(Hp.*Hp)*O;
  [MinCutGroupsList, MinCutWeight] = MinCut([1],W);
  innerlist indices = MinCutGroupsList(1,:);
  innerlist indices = innerlist indices(innerlist indices \sim = 0);
  innerlist = zeros(1,length(innerlist indices));
  for i = 1:length(innerlist indices)
    innerlist(i) = list(innerlist indices(i));
  end
  %Compare across all parents
  innerweight = weight calculate(innerlist,list)/self weight(list);
```

```
if(innerweight<minweight)
    minweight = innerweight;
    minlist = innerlist;
    minlist comp = setdiff(list,minlist);
    minparent = 1;
  end
  end
end
childset = struct('values', {});
for l = 1:no parents;
  if l<minparent
    childset(l).values = parentset(l).values;
  end
  if 1 == minparent
    childset(1).values = minlist;
    childset(l+1).values = minlist comp;
  end
  if(l>minparent)
    childset(l+1).values = parentset(l).values;
  end
end
c = no parents + 1;
tot coupling lost = [];
for i = 1:c-1
  tot coupling lost i = [];
  for j = i+1:c
    coupling i j = findcoupling(childset(i).values,childset(j).values);
    if (i == i)
      coupling i j = 0*coupling i j;
    end
     tot coupling lost i = [tot coupling lost i; coupling i j];
  end
  tot coupling lost = [tot coupling lost;tot coupling lost i];
end
clf1 = (norm(tot coupling lost))/norm(Hess);
mcs = 1/c;
The MATLAB functions "weight calculate", "self weight" and "findcoupling" used in the
```

```
cluster2 = setdiff(list,innerlist);
```

```
bout = [];
for i = 1:length(cluster1)
 bin = [];
 for j = 1:length(cluster2)
   bin = [bin, Hess((cluster1(i)-1)*N+1:cluster1(i)*N, (cluster2(j)-1)*N+1:cluster2(j)*N)];
 end
 bout = [bout;bin];
end
result = norm(bout);
function result = self weight(list)
global N Hess Nu
bout = [];
for i = 1:length(list)
 bin = [];
 for j = 1:length(list)
   bin = [bin, Hess((list(i)-1)*N+1:list(i)*N, (list(j)-1)*N+1:list(j)*N)];
 end
 bout = [bout;bin];
end
result = norm(bout);
function result = findcoupling(cluster1,cluster2)
global N Hess Nu
bout = [];
for i = 1:length(cluster1)
 bin = [];
 for j = 1:length(cluster2)
   bin = [bin, Hess((cluster1(i)-1)*N+1:cluster1(i)*N, (cluster2(j)-1)*N+1:cluster2(j)*N)];
 end
 bout = [bout;bin];
end
result = norm(bout);
function [MinCutGroupsList, MinCutWeight] = MinCut(SourceVertices, WeightedGraph)
%%% performs Min Cut algorithm described in "A Simple Min Cut Algorithm" by
%%% M. Stoer and F. Wagner
%%%[source: http://www.mathworks.com/matlabcentral/fileexchange/13892-a-simple-min-cut-
```

```
algorithm]
```

%%% input -

```
%%%
        SourceVertices - a list of vertices that are forced to be kept in one side of the cut.
%%%
        WeightedGraph - symetric matrix of edge weights. Wi, j is the edge connecting
%vertices i.j
%%%
                  use W_{i,j}=0 or W_{i,j}=0 inf to indicate unconnected vertices.
%%% output -
%%% MinCutGroupsList - two lists of verices, SECOND one contains the source vertives
%%% MinCutWeight - sum of weight of edges alosng the cut
%%% (C) Yohai Devir 2006
%%% <my first name> AT WHOEVER D0T COM
  GraphDim = size(WeightedGraph, 1);
  SourceVertices = SourceVertices(SourceVertices \sim = 0); %remove zero vertices
  %%% remove self edges and ZEROed ones
  WeightedGraph = WeightedGraph+diag(inf(1,GraphDim));
  % for ii = 1:GraphDim
  % WeightedGraph(ii,ii) = inf;
  % end
  WeightedGraph(WeightedGraph == 0) = inf;
  %%%Merge all Source Vrtices to one, so they'll be unbreakable, descending order is
%VITAL!!!
  SourceVertices = sort(SourceVertices);
  GroupsList = zeros(GraphDim); %each row are the vertices melted into one vertex in %the
table.
  GroupsList(:,1) = 1:GraphDim;
  for ii=length(SourceVertices):-1:2;
    [WeightedGraph,GroupsList] =
MeltTwoVertices(SourceVertices(1),SourceVertices(ii),WeightedGraph,GroupsList);
  end
  Split = GroupsList(:,1);
  %%% By now I have a weighted graph in which all seed vertices are
  %%% merged into one vertex. Run Mincut algrithm on this graph
  [MinCutGroupsList L, MinCutWeight] = MinCutNoSeed(WeightedGraph);
  %%% Convert Data so the seed vertices will be reconsidered as different
  %%% vertices and not one vertex.
  for ii = 1:2
    MinCutGroupsList(ii.:) = Local2GlobalIndices(MinCutGroupsList L(ii.:), Split);
  end
  if (length(find(MinCutGroupsList(1,:) == SourceVertices(1))) == 1)
    SeedLocation = 1;
  else
    SeedLocation = 2;
  end
  MinCutGroupsList withSeed =
[MinCutGroupsList(SeedLocation,(MinCutGroupsList(SeedLocation,:)~=0))
SourceVertices(2:length(SourceVertices))];
```

```
MinCutGroupsList withSeed = sort(MinCutGroupsList withSeed);
  MinCutGroupsList withSeed = [MinCutGroupsList withSeed zeros(1,GraphDim -
length(MinCutGroupsList withSeed))];
  MinCutGroupsList NoSeed = MinCutGroupsList(3 - SeedLocation,(MinCutGroupsList(3 -
SeedLocation,:)~=0));
  MinCutGroupsList NoSeed = sort(MinCutGroupsList NoSeed);
  MinCutGroupsList NoSeed = [MinCutGroupsList NoSeed zeros(1,GraphDim -
length(MinCutGroupsList NoSeed))];
  MinCutGroupsList = [MinCutGroupsList NoSeed ; MinCutGroupsList withSeed];
return
%%% Perform ordinary Stoer & Wagner algorithm Min Cut algorithm
<u>%</u>
function [MinCutGroupsList, MinCutWeight] = MinCutNoSeed(WeightedGraph)
  GraphDim = size(WeightedGraph, 1);
  GroupsList = zeros(GraphDim);
  GroupsList(:,1) = 1:GraphDim;
  MinCutWeight = inf;
 MinCutGroup = [];
  for ii = 1:GraphDim-1
   [OneBefore, LastVertex] = MinimumCutPhase(WeightedGraph);
   if OneBefore == -1 %Graph is not connected. LastVertex is a group of vertices not
connected to Vertex 1
     MinCutGroup L = LastVertex(LastVertex \sim = 0); clear LastVertex; %it's not the last
vertex
     MinCutGroupsList = [];
     for jj = 1:length(MinCutGroup L);
       MinCutGroup temp = GroupsList(MinCutGroup L(jj));
       MinCutGroup temp = MinCutGroup temp(MinCutGroup temp~=0);
       MinCutGroupsList = [MinCutGroupsList MinCutGroup temp];
     end
     MinCutGroupsList = [MinCutGroupsList zeros(1,GraphDim -
length(MinCutGroupsList))];
     ii = 1;
     for kk=1:GraphDim
       if (find(MinCutGroupsList(1,:) == kk))
         MinCutGroupsList(2,jj) = kk;
         jj = jj + 1;
       end
     end
     MinCutWeight = 0;
     return
   end %of: If graph is not connected
```

```
206
```

```
Edges = WeightedGraph(LastVertex.:);
    Edges = Edges(isfinite(Edges));
    MinCutPhaseWeight = sum(Edges);
    if MinCutPhaseWeight < MinCutWeight
      MinCutWeight = MinCutPhaseWeight;
      MinCutGroup = GroupsList(LastVertex,:);
      MinCutGroup = MinCutGroup(MinCutGroup~=0);
    end
    [WeightedGraph,GroupsList] =
MeltTwoVertices(LastVertex, OneBefore, WeightedGraph, GroupsList);
  end
  MinCutGroup = sort(MinCutGroup);
  MinCutGroupsList = [MinCutGroup zeros(1,GraphDim - length(MinCutGroup))];
  ii = 1;
  for kk=1:GraphDim
    if isempty(find(MinCutGroup(1,:) == kk,1))
      MinCutGroupsList(2,jj) = kk;
      ii = ii + 1;
    end
  end
return
%%% This function takes V1 and V2 as vertexes in the given graph and MERGES
%%% THEM INTO V1 !!
%%% The output is the UpdatedGraph inwhich both vertices are considered
%%% one, and updated GroupsList to reflect the change.
function
            [UpdatedGraph,GroupsList] =
MeltTwoVertices(V1,V2,WeightedGraph,GroupsList)
  t = min(V1, V2);
  V2 = max(V1, V2);
  V1 = t;
  GraphDim = size(WeightedGraph,1);
  UpdatedGraph = WeightedGraph;
  Mask1 = isinf(WeightedGraph(V1,:));
  Mask2 = isinf(WeightedGraph(V2,:));
  UpdatedGraph(V1,Mask1) = 0;
  UpdatedGraph(V2,Mask2) = 0;
  infMask = zeros(1,size(Mask1,2));
% infMask(find(Mask1 & Mask2)) = inf;
  infMask((Mask1 & Mask2)) = inf;
  UpdatedGraph(V1,:) = UpdatedGraph(V1,:) + UpdatedGraph(V2,:) + infMask;
  UpdatedGraph(:,V1) = UpdatedGraph(V1,:)';
  selectVec = true(1,GraphDim); selectVec(V2) = false;
% UpdatedGraph = [UpdatedGraph(1:V2-1,:); UpdatedGraph(V2+1:GraphDim,:)]; %remove
second vertex from graph
```

```
% UpdatedGraph = [UpdatedGraph(:,1:V2-1) UpdatedGraph(:,V2+1:GraphDim)];
```

```
UpdatedGraph = UpdatedGraph(selectVec,selectVec);
  UpdatedGraph(V1,V1) = inf;
                                                % group-group distance
  V1list = GroupsList(V1, (GroupsList(V1,:)\sim=0));
  V2list = GroupsList(V2, (GroupsList(V2,:)\sim=0));
 GroupsList(V1,:) = [V1list V2list zeros(1,size(GroupsList,2)-length(V1list) - length(V2list))
]; %shorten grouplist
% GroupsList = [GroupsList(1:V2-1,:);GroupsList(V2+1:GraphDim,:)];
  GroupsList = GroupsList(selectVec,:);
return
%%% perform one phase of the algorithm
%%%
%%% return [-1, B] in case of Unconnected Graph when B is a subgraph(s)
%%% that are not connected to Vertex 1
function [OneBefore, LastVertex] = MinimumCutPhase(WeightedGraph)
  GraphDim = size(WeightedGraph,1);
  GroupsList = zeros(GraphDim);
 GroupsList(:,1) = 1:GraphDim;
 if size(WeightedGraph, 1) > 2
   FarestVertexGroup = 0;
   for ii = 1:GraphDim-1
     OneBefore
                  = FarestVertexGroup(1);
     PossibleVertices = WeightedGraph(1,1:size(WeightedGraph,2));
     PossibleVertices(isinf(PossibleVertices)) = 0;
     FarestVertex = find(PossibleVertices == max(PossibleVertices),1,'first');
     if FarestVertex == 1 %In case the Graph is not connected
       OneBefore = -1;
       LastVertex = GroupsList(1,:);
       return
     end
     FarestVertexGroup = GroupsList(FarestVertex,:);
     [WeightedGraph,GroupsList] =
MeltTwoVertices(1,FarestVertex,WeightedGraph,GroupsList);
   end
   LastVertex = FarestVertexGroup(1);
  else
   OneBefore = 1:
   LastVertex = 2;
 end
return
%%% Having a local list of indices in a global list and sublist of the
```

```
%%% local list, find the corresponding global indices.
```

Appendix C

Codes for 9-zone building example in section 4.3.10

To obtain the results presented in this section, the following programs need to be run in the specified sequence:

```
STEP 1: Obtain weighted incidence matrix, capacitance matrices, and state space matrices (see
Algorithm 3.1)
clc
clear all;
% Part 1: Generation of Incidence matrix (similar to code used for 9-zone building % model in
section 4.2.8)
clc
clear all
N alpha = [];
Incid = [];
overall rat = 0.06;
rat1 = 1;
rat2 = 1/3;
% Set 1: Horizontal external walls
count 1 = 0;
for i = 1:3
  count 1 = \text{count } 1+1;
  currentrow = zeros(1,10);
  booleanrow = zeros(1,10);
  currentrow(i) = 1/(29.99*overall rat);
  booleanrow(i) = 1;
  currentrow(10) = 1/(81.08*overall rat);
  booleanrow(10) = 1;
  N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
end
for i = 7:9
  currentrow = zeros(1,10);
  booleanrow = zeros(1,10);
  currentrow(i) = 1/(29.99*overall rat);
```

```
booleanrow(i) = 1;
  currentrow(10) = 1/(81.08*overall rat);
  booleanrow(10) = 1;
  count 1 = \text{count } 1+1;
  N_alpha = [N alpha;currentrow];
  Incid = [Incid;booleanrow];
end
% Set 2: Vertical external walls
count 2 = 0;
for i = 1:3:7
  currentrow = zeros(1,10);
  booleanrow = zeros(1,10);
  currentrow(i) = 1/(36.84*overall rat);
  booleanrow(i) = 1;
  currentrow(10) = 1/(82.00*overall rat);
  booleanrow(10) = 1;
  count 2 = \text{count } 2+1;
  N alpha = [N alpha;currentrow];
  Incid = [Incid;booleanrow];
end
for i = 1:9
  currentrow = zeros(1,10);
  booleanrow = zeros(1,10);
  currentrow(i) = 1/(36.84*overall rat);
  booleanrow(i) = 1;
  currentrow(10) = 1/(82.00*overall rat);
  booleanrow(10) = 1;
  count 2 = \text{count } 2+1;
  N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
end
for i = 3:3:9
  currentrow = zeros(1,10);
  booleanrow = zeros(1,10);
  currentrow(i) = 1/(36.84*overall rat);
  booleanrow(i) = 1;
  currentrow(10) = 1/(82.00*overall rat);
  booleanrow(10) = 1;
  count 2 = \text{count } 2+1;
```

```
N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
end
for i = 1:9
  currentrow = zeros(1,10);
  booleanrow = zeros(1,10);
  currentrow(i) = 1/(36.84*overall rat);
  booleanrow(i) = 1;
  currentrow(10) = 1/(82.00*overall rat);
   booleanrow(10) = 1;
  count 2 = \text{count}_2 + 1;
  N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
end
% Set 3: Horizontal internal walls
count 3 = 0;
for i = 1:6
currentrow = zeros(1,10);
booleanrow = zeros(1,10);
  currentrow(i) = 1/(rat1*21.32*overall rat);
  booleanrow(i) = 1;
  currentrow(i+3) = 1/(rat1*21.32*overall rat);
  booleanrow(i+3) = 1;
  count 3 = \text{count } 3+1;
  N alpha = [N alpha;currentrow];
  Incid = [Incid;booleanrow];
end
% Set 4: Vertical internal walls
count 4 = 0;
for i = 1:2
  for j = i:3:i+6
    currentrow = zeros(1,10);
    booleanrow = zeros(1,10);
    currentrow(j) = 1/(rat2*21.32*overall rat);
     booleanrow(j) = 1;
     currentrow(j+1) = 1/(rat2*21.32*overall rat);
    booleanrow(j+1) = 1;
    count 4 = \text{count } 4+1;
    N alpha = [N alpha; currentrow];
```

```
Incid = [Incid;booleanrow];
  end
end
% Part 2: Generation of capacitance matrices
C w =
0.1*blkdiag(8329.15*eye(count 1),8329.15*eye(count 2),4660*eye(count 3),4660*eye(count
4));
c z = 1*250*ones(9,1);
C z = blkdiag(250*eye(9));
C cap = blkdiag(C_w, C_z);
% Part 3: Generation of state space matrices
nw = count 1+count 2+count 3+count 4;
nz = 9;
nx = nw + nz;
nu = 9;
N beta = -N alpha;
N alpha z = N alpha(:,1:nz);
N beta z = N beta(:,1:nz);
S r beta = diag(sum(N beta'));
S c alpha z = diag(sum(N alpha z));
N alpha a = N alpha(:,nz+1:nz+1);
inv C cap = inv(C cap);
A bldg = inv C cap*[[S r beta,N alpha z];[-N_beta_z',-S_c_alpha_z]];
B_bldg = inv_C_cap*[[zeros(nw,nz),N_alpha_a];[eye(nz),zeros(nz,1)]];
C bldg = [zeros(nz,nw),eye(nz)];
D bldg = zeros(nz,nz+1);
% Part 4: State space matrices in desired form, and zone-wall matrix
A ww = A bldg(1:nw,1:nw);
A wz = A bldg(1:nw,nw+1:nw+nz);
A zw = A bldg(nw+1:nw+nz,1:nw);
A_zz = A_bldg(nw+1:nw+nz,nw+1:nw+nz);
B zd = B bldg(nw+1:nw+nz,1:nz);
ZW = (Incid(:,1:nz))';
Nw = nw;
Nz = nz;
% Part 5: MPC Parameters
Ts = 60:
alphaa = 0.1;
N = 5;
```

```
i = 1:
min olf = 99999999999;
  res = SetPartition(Nz,i); %Obtains set partitions
  n part i = length(res);
  local part = struct;
  for k = 1:n part ilo
    for i = 1:i
      local part(j).value = (res{k}{j})';
    end
    local olf(k) = find new olf(i,local part);
    hold on
    if (local olf(k) < min olf)
      min olf = local olf(k);
      min part = local part;
    end
  end
  hold on:
   scatter(i*ones(n part i,1),local olf,'k','filled'); %Plots trade-off curve
  optimal olf(i) = min olf;
  optimal part = min part;
The function "SetPartition" appearing in the above program is part of a package downloaded
from MATLAB Central. It is included in the media accompanying this thesis.
```

```
s dc parent(6).value = [6]';
s dc parent(7).value = [7]';
s dc parent(8).value = [8]';
s dc parent(9).value = [9]';
olf = zeros(Nz, 1);
olf(Nz) = find new olf(n parent, s dc parent);
fpm(Nz) = find fpm(s dc parent);
for p = Nz-1:-1:1
s dc child min = struct;
for i = 1:n parent-1
  for j = i+1:n parent
    n child = n parent-1;
    s dc child = struct;
      for k = 1:j-1
      s dc child(k).value = s dc parent(k).value;
      end
      s dc child(i).value = [s dc parent(i).value;s dc parent(j).value];
    if(j<n parent)
       for k = j:n parent-1
         s dc child(k).value = s dc parent(k+1).value;
       end
    end
    olf child = find new olf(n child,s dc child);
    if(olf child <= olf min)
       olf min = olf child;
       1 \text{ max} = \text{length}(\text{s dc child});
       for 1 = 1:1 \text{ max}
       s dc child min(l).value = s dc child(l).value;
       end
    end
  end
end
n parent = n child;
1 \text{ max} = \text{length}(\text{s dc child min});
for l = 1:l \max
s dc parent(l).value = s dc child min(l).value;
end
olf(p) = olf min;
fpm(p) = find fpm(s dc child min);
disp('The Optimal child partition for p = ');
disp(p);
for l = 1:p
  disp(s dc child min(l).value);
```

The functions "find_new_olf" and "find_fpm" used in the above code are shown below.

```
function OLF = find new olf(n,s dc)
load nine zone model
%Decentralized Boolean matrices
% Generate r dc data structure (find walls for each cluster)
r dc = struct;
N dc = zeros(n,1);
N dc w = zeros(n,1);
for i = 1:n
 N dc(i) = length(s dc(i).value);
 current wall set = [];
 zone set = s dc(i).value;
 for l = 1:N dc(i)
   wall choice = [];
   for j = 1:Nw
     if(ZW(zone set(1),j)==1)
      wall choice = union(wall choice,j);
     end
   end
  current wall set = union(current wall set, wall choice);
 end
 r dc(i).value = current wall set;
 N dc w(i) = length(current wall set);
end
%generate Pi and Qi
P dc = struct;
O dc = struct;
for i = 1:n
 mat P = zeros(N dc(i),Nz);
 si = s dc(i).value;
 for l = 1:N dc(i)
   mat_P(1,si(1)) = 1;
 end
 P dc(i).value = mat P;
 mat Q = zeros(N dc w(i), Nw);
 ri = r dc(i).value;
```

```
for l = 1: N dc w(i)
   mat Q(l,ri(l)) = 1;
 end
 Q dc(i).value = mat Q;
end
%Generation of combined and lifted matrices
P 1 dc = struct:
P = zeros(Nz,Nz);
P 1 = zeros(N*Nz,N*Nz);
for i = 1:n
 local mat = zeros(N*N dc(i), N*N dc(i));
 for j = 1:N
   local mat(N dc(i)*(j-1)+1:N dc(i)*j,Nz*(j-1)+1:Nz*j) = P dc(i).value;
 end
 P 1 dc(i).value = local mat;
end
rowpos = 0;
for i = 1:n
 P(rowpos+1:rowpos+N dc(i),:) = P dc(i).value;
 rowpos = rowpos + N dc(i);
end
rowpos = 0;
for i = 1:n
 P l(rowpos+1:rowpos+N dc(i)*N,:) = P 1 dc(i).value;
 rowpos = rowpos+N dc(i)*N;
end
%Decentralized MPC simplified model
A dc = struct;
B dc = struct;
for i = 1:n
 for j = 1:n
   if (i == i)
     A dc(i,j).value = [(Q dc(i).value)*Ad ww*(Q dc(i).value)'
(Q dc(i).value)*Ad wz*(P dc(i).value)';(P dc(i).value)*Ad zw*(Q dc(i).value)'
(P_dc(i).value)*Ad_zz*(P_dc(i).value)'];
   else
     A dc(i,j).value = [(Q dc(i).value)*Ad wz*(P dc(j).value)';zeros(N dc(i),N dc(j))];
   end
 end
  B_dc(i).value = [zeros(N_dc_w(i),N_dc(i));(P_dc(i).value)*Bd_zd*(P_dc(i).value)'];
end
%M dc data structure
```

```
M dc = struct;
for i = 1:n
          for k = 1:N
                    temp mat = zeros(N dc(i)+N dc w(i),(N dc(i)+N dc w(i))*N);
                    for l = 1:k
                              temp mat(:,(N dc(i)+N dc w(i))*(l-1)+1:(N dc(i)+N dc w(i))*l) =
(A dc(i,i).value)(k-l);
                    end
                    temp mat(:,(N dc(i)+N dc w(i))*k+1:(N dc(i)+N dc w(i))*N) =
\operatorname{zeros}((N \operatorname{dc}(i)+N \operatorname{dc} w(i)),(N-k)*(N \operatorname{dc}(i)+N \operatorname{dc} w(i)));
                    M dc(i,k).value = temp mat;
          end
end
%B diag data structure
B diag = struct;
for i = 1:n
         local mat = zeros((N dc(i)+N dc w(i))*N,N dc(i)*N);
          for k = 1:N
                    local mat((N dc(i)+N dc w(i))*(k-1)+1:(N dc(i)+N dc w(i))*k,N dc(i)*(k-1)+1:(N dc w(i))*k,N dc(i)*k,N dc(
1)+1:N dc(i)*k = B dc(i).value;
          end
          B diag(i).value = local mat;
end
%R dc data structure
R dc = struct;
for i = 1:n
          mat sum = zeros((N dc(i)+N dc w(i))*N,(N dc(i)+N dc w(i)));
          for k = 1:N
                    mat sum = mat sum + (M dc(i,k).value)';
          end
          R dc(i).value = 2^{(B)} diag(i).value)'*mat sum;
end
%S dc data structure
S dc = struct;
for i = 1:n
          for i = 1:n
                   if(i==j)
                              S dc(i,j).value = zeros(N*N dc(i),N*N dc(j));
                    else
                    mat sum = zeros((N dc(i)+N dc w(i))*N,(N dc(i)+N dc w(i))*N);
                    blk mat = zeros(N*(N dc(i)+N_dc_w(i)),N*N_dc(j));
                    for k = 1:N
                              mat sum = mat sum + (M dc(i,k).value)'*(M dc(i,k).value);
                             blk mat((N dc(i)+N dc w(i))*(k-1)+1:(N dc(i)+N dc w(i))*k,N dc(j)*(k-1)+1:(N dc w(i))*(k-1)+1:(N dc w(i))*(k-1)+1:(N dc w(i))*(k-1)+1:(N dc w(i))*k,N dc(j)*(k-1)+1:(N dc w(i))*k,N dc(j)*(k-1)+1:(N dc w(i))*k,N dc(j)*(k-1)+1:(N dc w(i))*k,N dc(j)*(k-1)+1:(N dc w(i))*(k-1)+1:(N dc w(i))*(k-1)+1
1)+1:N dc(j)*k)=A dc(i,j).value;
```

```
end
   S dc(i,j).value = 2*(B diag(i).value)'*mat sum*blk mat;
   end
 end
end
%H dc data structure
H dc = struct;
for i = 1:n
 mat sum = zeros((N dc(i)+N dc w(i))*N,(N dc(i)+N dc w(i))*N);
 for k = 1:N
   mat sum = mat sum + (M dc(i,k).value)'*(M dc(i,k).value);
 end
 H dc(i).value = alphaa*eye(N*N dc(i))+(B diag(i).value)'*mat sum*B diag(i).value;
end
%Combination Matrices
%Hdc
Hdc = zeros(N*Nz,N*Nz);
pos = 0;
for i = 1:n
 Hdc(pos+1:pos+N*N dc(i),pos+1:pos+N*N dc(i)) = H dc(i).value;
 pos = pos + N*N dc(i);
end
%Sdc
Sdc = zeros(N*Nz,N*Nz);
rowpos = 0;
for i = 1:n
 colpos = 0;
 for j = 1:n
 Sdc(rowpos+1:rowpos+N*N dc(i),colpos+1:colpos+N*N dc(j)) = S dc(i,j).value;
 colpos = colpos + N*N dc(i);
 end
 rowpos = rowpos + N*N dc(i);
end
%Rdc
Rdc = zeros(N*Nz,Nz+Nw);
rowpos = 0;
colpos = 0;
for i = 1:n
 Rdc(rowpos+1:rowpos+N*N dc(i),colpos+1:colpos+N dc(i)+N dc w(i)) = R dc(i).value;
 rowpos = rowpos + N*N dc(i);
 colpos = colpos + N dc(i) + N dc w(i);
end
```

```
%Centralized model matrices
A c = [Ad ww, Ad wz; Ad zw, Ad zz];
B c = [zeros(Nw,Nz);Bd zd];
%M c data structure
M c = struct;
 for k = 1:N
   temp mat = zeros(Nz+Nw,(Nz+Nw)*N);
   for l = 1:k
    temp mat(:,(Nz+Nw)*(l-1)+1:(Nz+Nw)*l) = (A c)^(k-l);
   end
   temp mat(:,(Nz+Nw)*k+1:(Nz+Nw)*N) = zeros((Nz+Nw),(N-k)*(Nz+Nw));
   M c(k).value = temp mat;
 end
%B diag c matrix
 B diag c = zeros((Nz+Nw)*N,Nz*N);
 for k = 1:N
   B diag c((Nz+Nw)*(k-1)+1:(Nz+Nw)*k,Nz*(k-1)+1:Nz*k) = B c;
 end
%H c matrix
 mat sum = zeros((Nz+Nw)*N,(Nz+Nw)*N);
 for k = 1:N
   mat sum = mat sum + (M c(k).value)'*(M c(k).value);
 end
 H c = alphaa*eye(N*Nz)+(B diag c)'*mat sum*B diag c;
0/0***********************
M = inv(P \ l)*inv(Hdc)*Sdc*P \ l;
OLF mat = M'*H c*M;
OLF = max(eig(OLF mat));
\%OLF = norm(M,inf);
End
function FPM = find fpm(s dc)
load nine zone model
p = length(s dc);
N dc = zeros(p,1);
summ = 0:
for i = 1:p
 cluster i = s dc(i).value;
 N dc i = length(cluster i);
 cap summ i = 0;
 for j = 1:N dc i
```

Appendix D

Codes for 11-zone building example in section 4.3.11

To perform the agglomerative clustering analysis presented in this section, the following programs need to be run in the specified sequence:

STEP 1: Obtain weighted incidence matrix, capacitance matrices, and state space matrices (see Algorithm 3.1) %Model Inputs and MPC parameters for 11 zone circular bldg clc clear all; % Part 1: Generation of Incidence matrix clc clear all N alpha = []; Incid = [];rat1 = 50;rat2 = 1;% Set 1: Horizontal external walls count 1 = 0; for i = 1:11count 1 = count 1+1; currentrow = zeros(1,12); booleanrow = zeros(1,12); if(i==1)currentrow(i) = 1/(0.5*18.356); booleanrow(i) = 1: currentrow(12) = 1/(0.5*18.356);booleanrow(12) = 1; N alpha = [N alpha; currentrow];Incid = [Incid;booleanrow]; $elseif((i \ge 2)\&\&(i \le 6))$ currentrow(i) = 1/(0.5*30.594); booleanrow(i) = 1; currentrow(12) = 1/(0.5*30.594);booleanrow(12) = 1; N alpha = [N alpha; currentrow];Incid = [Incid;booleanrow];

```
else
    currentrow(i) = 1/(0.5*16.6874);
    booleanrow(i) = 1;
    currentrow(12) = 1/(0.5*16.6874);
    booleanrow(12) = 1;
    N alpha = [N alpha;currentrow];
    Incid = [Incid;booleanrow];
  end
end
for i = 1:11
  count 1 = \text{count } 1+1;
  currentrow = zeros(1,12);
  booleanrow = zeros(1,12);
  if(i=1)
    currentrow(i) = 1/(0.5*18.356);
    booleanrow(i) = 1;
    currentrow(12) = 1/(0.5*18.356);
    booleanrow(12) = 1;
    N alpha = [N alpha;currentrow];
    Incid = [Incid;booleanrow];
  elseif((i \ge 2)\&\&(i \le 6))
    currentrow(i) = 1/(0.5*30.594);
    booleanrow(i) = 1;
    currentrow(12) = 1/(0.5*30.594);
    booleanrow(12) = 1;
    N alpha = [N alpha; currentrow];
    Incid = [Incid;booleanrow];
  else
    currentrow(i) = 1/(0.5*16.6874);
    booleanrow(i) = 1;
    currentrow(12) = 1/(0.5*16.6874);
    booleanrow(12) = 1;
    N alpha = [N alpha; currentrow];
    Incid = [Incid;booleanrow];
  end
end
% Set 2: Vertical external walls
count 2 = 0;
for i = 7:11
  currentrow = zeros(1,12);
  booleanrow = zeros(1,12);
  currentrow(i) = 1/(0.5*18.827);
  booleanrow(i) = 1;
  currentrow(12) = 1/(0.5*18.827);
  booleanrow(12) = 1;
```

```
count 2 = \text{count } 2+1;
  N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
end
%Set 3: Horizontal internal walls – no such walls since single floor building
count 3 = 0;
%Set 4: Vertical internal walls
count 4 = 0;
for i = 2:6
  currentrow = zeros(1,12);
  booleanrow = zeros(1,12);
  currentrow(1) = 1/(0.5*18.32);
  booleanrow(1) = 1;
  currentrow(i) = 1/(0.5*18.32);
  booleanrow(i) = 1;
  count 4 = \text{count } 4+1;
  N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
end
for i = 2:5
  currentrow = zeros(1,12);
  booleanrow = zeros(1,12);
  currentrow(i) = 1/(0.5*23.03);
  booleanrow(i) = 1;
  currentrow(i+1) = 1/(0.5*23.03);
  booleanrow(i+1) = 1;
  count 4 = \text{count } 4+1;
  N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
end
  currentrow = zeros(1,12);
  booleanrow = zeros(1,12);
  currentrow(2) = 1/(0.5*46.06*rat1);
  booleanrow(2) = 1;
  currentrow(6) = 1/(0.5*46.06*rat1);
  booleanrow(6) = 1;
  count 4 = \text{count } 4+1;
  N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
for i = 2:6
  currentrow = zeros(1,12);
  booleanrow = zeros(1,12);
  currentrow(i) = 1/(0.5*17.25*rat1);
  booleanrow(i) = 1;
```

```
currentrow(i+5) = 1/(0.5*17.25*rat1);
```

```
booleanrow(i+5) = 1;
  count 4 = \text{count } 4+1;
  N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
end
for i = 7:10
  currentrow = zeros(1,12);
 booleanrow = zeros(1,12);
  currentrow(i) = 1/(0.5*23.03);
  booleanrow(i) = 1;
  currentrow(i+1) = 1/(0.5*23.03);
  booleanrow(i+1) = 1;
  count 4 = \text{count } 4+1;
  N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
end
  currentrow = zeros(1,12);
  booleanrow = zeros(1,12);
  currentrow(7) = 1/(0.5*46.06*rat1);
  booleanrow(7) = 1;
  currentrow(11) = 1/(0.5*46.06*rat1);
  booleanrow(11) = 1;
  count 4 = \text{count } 4+1;
 N alpha = [N alpha; currentrow];
  Incid = [Incid;booleanrow];
% Part 2: Generation of capacitance matrices
c w =
*2.77*10^{4};ones(4,1)*5.2*10^{3};1.04*10^{4};
C w = diag(c w);
c z = [643.72; ones(5,1)*386.23; ones(5,1)*708.09];
C z = diag(c z);
C cap = blkdiag(C w,C z);
% Part 3: Generation of state space matrices
nw = count 1 + count 2 + count 3 + count 4;
nz = 11:
N beta = -N alpha;
N alpha z = N alpha(:,1:nz);
N beta z = N beta(:,1:nz);
S r beta = diag(sum(N beta'));
S c alpha z = diag(sum(N alpha z));
N alpha a = N alpha(:,nz+1:nz+1);
inv C cap = inv(C cap);
A_bldg = inv_C_cap*[[S_r_beta,N_alpha_z];[-N_beta_z',-S_c_alpha_z]];
```

%B bldg = inv C cap*[[zeros(nw,nz),zeros(nw,nz),N alpha a];[eye(nz),eye(nz),zeros(nz,1)]]; B bldg = inv C cap*[[zeros(nw,nz),N alpha a];[eye(nz),zeros(nz,1)]]; C bldg = [zeros(nz,nw),eye(nz)];D bldg = zeros(nz,nz+1); % Part 4: State space matrices in desired form, and zone-wall matrix A ww = A bldg(1:nw,1:nw);A wz = A bldg(1:nw,nw+1:nw+nz); A zw = A bldg(nw+1:nw+nz,1:nw); A zz = A bldg(nw+1:nw+nz,nw+1:nw+nz); B normal w = B bldg(1:nw,:); B normal z = B bldg(nw+1:nw+nz,:); B zd = B bldg(nw+1:nw+nz,1:nz);ZW = (Incid(:,1:nz))';Nw = nw;Nz = nz;%Part 5: MPC Parameters Ts = 600: alphaa = 0.1;N = 5: %Part 6: Discretization of model Ad ww = eye(Nw) + Ts*A ww;Ad $wz = Ts^*A wz;$ Ad $zw = Ts^*A zw$; Ad zz = eye(Nz) + Ts*A zz;Bd zd = Ts*B zd;%Part 7: Simulation amplitudes – needed for open loop simulations later in this section a1 = 0.5;a2 = 0.5;b1 = 10;b2 = 20;save circular model 0/0 ***************

STEP 2: To run the agglomerative clustering, run the program in Step 3 of Appendix C with the following changes:

(i) Replace "n_parent = 9;" with "n_parent = 11;"

(ii) Include the lines "s_dc_parent(10).value = [10]';" and "s_dc_parent(11).value = [11]';" after the line "s_dc_parent(9).value = [9]';"

(iii) In the functions "find_new_olf" and "find_fpm", replace "load nine_zone_model" with "load circular_model"

To obtain the plot in Fig. 4.21 corresponding to the scalarized analysis, the following code was used. Note that this code provides the plot only for the case when $\lambda = 0.5$. It can be re-run with the appropriate modification "lambda = 0.85;" for the case when $\lambda = 0.85$.

To obtain the open loop simulation results, cluster level state-space models can be obtained using the following code. The models hence obtained are used in Simulink models to perform the necessary simulations. These Simulink models (for centralized, knee2 and fully decentralized cases) are included in the media accompanying this thesis.

```
zone set = [1,2,3,4,5,6]; % Zone numbers corresponding to the cluster go here. In this
% example, we use the first cluster for knee 2 in Fig. 4.20
n zone set = length(zone set);
wall set = [];
for i = 1:n zone set
  w i = [];
  for j = 1:nw
    if(ZW(zone set(i),j)==1)
       w i = [w i; j];
    end
  end
  wall set = union(wall set, w i);
end
n wall set = length(wall set);
Adecen ww = zeros(n wall set, n wall set);
Adecen wz = zeros(n wall set,n zone set);
Adecen zw = zeros(n zone set, n wall set);
Adecen zz = zeros(n zone set, n zone set);
Bdecen normal w = zeros(n wall set, nz+1);
Bdecen normal z = zeros(n zone set, nz+1);
Bdecen est w = zeros(n wall set, nz);
Bdecen est z = zeros(n zone set, nz);
for i = 1:n wall set
  Adecen ww(i,i) = A ww(wall set(i), wall set(i));
end
for i = 1:n wall set
  for j = 1:n zone set
    Adecen wz(i,j) = A wz(wall set(i), zone set(j));
  end
```

end for i = 1:n zone set for j = 1:n wall set Adecen zw(i,j) = A zw(zone set(i),wall set(j));end end for i = 1:n zone set Adecen zz(i,i) = A zz(zone set(i), zone set(i));end for i = 1:n wall set for j = 1:nz+1Bdecen normal w(i,j) = B normal w(wall set(i),j); end end for i = 1:n zone set for j = 1:nz+1Bdecen normal z(i,j) = B normal z(zone set(i),j); end end for i = 1:n wall set for j = 1:nzBdecen est w(i,j) = A wz(wall set(i),j); end end for i = 1:n wall set for j = 1:n zone set Bdecen est w(i,zone set(j)) = 0;end end Adecen = [Adecen ww,Adecen wz;Adecen zw,Adecen zz]; Bdecen = [Bdecen normal w,Bdecen est w;Bdecen normal z,Bdecen est z]; Cdecen = [zeros(n zone set, n wall set), eye(n zone set)]; Ddecen = zeros(n zone set, 2*nz+1);% The use of the number "1" in the variables "nwpart1" to "Dpart1" below corresponds to % the variable names that are used to refer to the state space model corresponding to this % cluster in the SIMULINK model nwpart1 = n wall set;nzpart1 = n zone set;Apart1= Adecen; Bpart1= Bdecen; Cpart1= Cdecen; Dpart1 = Ddecen:

Appendix E Steps used in creating an EnergyPlus model using OpenStudio

The following video tutorials provide information on creation of an EnergyPlus model using OpenStudio: <u>http://www.youtube.com/user/NRELOpenStudio</u>. Based on these tutorials, the procedure involves the following basic steps:

Step 1: Choose the construction template using the OpenStudio plugin in Google SketchUp Step 2: Draw the building in Google SketchUp. This involves the following substeps:

- (i) Create the plan drawing of the building using SketchUp drawing tools
- (ii) Use the "spaces from diagram" tool in the plugin to create floors based on the above plan
- (iii) Draw fenestration surfaces using the SketchUp drawing tools
- (iv) Use the "project loose geometry" tool in the plugin to project the fenestration surfaces on appropriate building surfaces
- (v) Additional geometry if needed can be created using the "shading surface tool", "internal partition surface tool, etc.
- Step 3: Use "Surface matching tool" in plugin to set appropriate boundary conditions for each surface.
- Step 4: Use "Space attributes tool" in plugin to assign stories, thermal zones and other attributes to each of the spaces in the building. "Render modes" can be used to check the successful application of these attributes.
- Step 5: Launch the OpenStudio application from the plugin to open the .osm file.
- Step 6: In the .osm file, set the paths for weather and design day files.
- Step 7: Examine the schedules via the schedules tab. The relevant schedules should already have been set to default values based on the construction template chosen in Step 1 above. If necessary, one or more schedules can be changed at this point by dragging and dropping schedules from the 'my model' or 'library' tabs.
- Step 8: Examine the constructions via the constructions tab. Default construction sets should already be present based on the construction template chosen in Step 1 above. For each construction set, the relevant applicability –the entire building, a story, a space type or a space should also be assigned by default. This information can be changed if needed. Also, each construction set consists of constructions, which in turn consist of a set of

materials. The constructions and/or materials constituting any such construction set can be changed at this point using the 'my model' or 'library' tabs.

- Step 9: Examine the load objects and the underlying attributes. Default values can be changed at this point.
- Step 10: Examine the space types using the "space types" tab. Construction sets, schedule sets and loads are assigned to each space type. These assignments can be changed for existing space types. Also new space types can be created with appropriate construction set, schedule set and load assignments.
- Step 11: Examine the stories using the "stories" tab. For each story, the underlying construction and schedule sets can be changed. Also new stories can be created by assigning appropriate construction and schedule sets.
- Step 12: Click on the "facilities" tab. Choose appropriate view type view by story/ space type/ thermal zone etc. Check the underlying assignments and ensure that the sets "unassigned space type", "unassigned thermal zone" etc. are empty. The facilities tab can be used to assign attributes such as loads at the building/space type/story level.
- Step 13: Click on the "thermal zones" tab. The "ideal air loads" option can be turned on/off at this stage. If the "ideal air loads" option is turned off, zone equipment can be assigned at this stage. Also, thermostats can be set at this point to each thermal zone.
- Step 14: The "HVAC systems tab" can be used to create HVAC system loops and specify the thermal zones which are serviced by the thermal "sink" element (e.g. Air handling Unit) in the loops. It can be used to modify the existing loops by adding/removing sink/source components, valves, terminal units etc.
- Step 15: The variables which are desired to be included the SQL file generated at the end of the EnergyPlus simulation can be selected, along with the reporting frequency using "output variables" tab.
- Step 16: Run the simulation using the "run simulation" tab. The directory containing the EnergyPlus input data file (idf) can be accessed using the "tree" sub-tab.
- Step 17: The "EP-Launch" EnergyPlus application can be used to perform an EnergyPlus simulation via the idf file generated above. The results of the simulation are stored in a .csv file which can be accessed using Microsoft Excel. The columns of this file provide the values of the output variables which were selected in Step 15 above.

Appendix F

Codes for performing zone level identification as per optimization framework presented in (6.3) – (6.6)

To obtain the results presented in section 6.3.2 for the optimization framework corresponding to (6.3) - (6.6), the following steps are required. For the purposes of illustration, these steps are explained for Zone F2. For the other zones, the steps are the same and the appropriate codes are provided in the media accompanying this thesis.

STEP 1: In the EnergyPlus model, PRBS set-points are prescribed for all zones in the building. The EnergyPlus model (.idf file), hence modified, is included in the media accompanying this thesis.

STEP 2: A day-long simulation is run using the above EnergyPlus model and the generic spreadsheet containing the output variables is obtained.

STEP 3: From the above generic spreadsheet, columns containing the variables needed for the zone level identification are used to create a new spreadsheet called "sim_data_zone_F2". This file is also included in the media accompanying this thesis.

STEP 4: Simulation data contained in "sim_data_zone_F2" is exported to the MATLAB workspace and saved in a MATLAB file called "zone F2 data" using the following MATLAB code:

```
% Import data for zone F2 in MATLAB
```

clc;

```
clear all;

Q_dot_{in} = 0.001*xlsread('sim_data_zone_F2.xls','F2_data','a2:a1441');

m_dot_supp = xlsread('sim_data_zone_F2.xls','F2_data','b2:b1441');

T_supp = xlsread('sim_data_zone_F2.xls','F2_data','c2:c1441');

T_z = xlsread('sim_data_zone_F2.xls','F2_data','d2:d1441');

T_w1 = xlsread('sim_data_zone_F2.xls','F2_data','e2:e1441');

T_w2 = xlsread('sim_data_zone_F2.xls','F2_data','g2:g1441');

T_w3 = xlsread('sim_data_zone_F2.xls','F2_data','g2:g1441');

T_w3_sub1 = xlsread('sim_data_zone_F2.xls','F2_data','h2:h1441');

T_w3_sub2 = xlsread('sim_data_zone_F2.xls','F2_data','i2:i1441');

T_w4 = xlsread('sim_data_zone_F2.xls','F2_data','j2:j1441');

T_w5 = xlsread('sim_data_zone_F2.xls','F2_data','k2:k1441');

T_w5_sub1 = xlsread('sim_data_zone_F2.xls','F2_data','k2:k1441');
```

T w6 = xlsread('sim data zone F2.xls','F2 data','m2:m1441'); V inf = xlsread('sim data zone F2.xls','F2 data','n2:n1441'); T amb = xlsread('sim data zone F2.xls','F2 data','o2:o1441'); save zone F2 data ***** STEP 5: The following MATLAB code can then be run to perform the zone level optimization. clc clear all % Part 1: Generation of optimal parameter values area1 = 98.4271*82.0208*0.3048^2; area2 = 98.4271*13.7812*0.3048^2; area4 = (72.1771-62.3333)*(7.875-2.95312)*0.3048^2; $area5 = (16.4062 - 1.64062) * (7.875 - 2.95312) * 0.3048^{2};$ area3 = 82.0208*13.7812*0.3048^2 - area4-area5; area6 = 82.0208*13.7812*0.3048^2; $area8 = (72.2344 - 3.28125) * (7.875 - 2.95312) * 0.3048^{2};$ $area7 = 98.4271 \times 13.7812 \times 0.3048^{2} - area8;$ area9 = 98.4271*82.0208*0.3048^2; areas = [area1 area2 area3 area4 area5 area6 area7 area8 area9]'; matlabpool open 4 options = optimset('Algorithm','sqp','MaxFunEvals',100000,'MaxIter',100000,'UseParallel','always','TolFun ', 1e-4, 'TolX', 1e-12, 'TolCon', 1e-12); init param = [1*0.01*800*ones(9,1);5000]; % initial parameter valued UB = [0.025*800*ones(9,1);10000]; % upper bounds on parameter values LB = [0*ones(9,1);0]; % lower bounds on parameter values optim param = fmincon(@zone F2 objective fn case1,init param,[],[],[],LB,UB,[],options); % call to % fmincon to obtain optimal parameter values using least squares optimization % Part 2: Comparison of zone temperature from least squares identification with EnergyPlus simulation data UA = optim param(1:9);C = optim param(10);load zone F2 data Tw meas = [T w1 T w2 T w3 T w3 sub1 T w3 sub2 T w4 T w5 T w5 sub1 T w6];startt index = 0; startt = 60*startt index; Tz pred(startt+1,1) = T z(startt+1); for k = startt+2:60*(startt index+24)Tz pred(k,1) = (Q dot in(k) + m dot supp(k)*1.012*T supp(k) + C*T z(k-1)/60 +(Tw meas(k-1,:))*(UA) + 1.2041*V inf(k-1)*1.012*T amb(k-1))/(C/60 + sum(UA) + 1.2041*V inf(k-1)*1.012*T amb(k-1)*1.012*T amb(k-1))/(C/60 + sum(UA) + 1.2041*V inf(k-1)*1.012*T amb(k-1)*1.012*T amb(k-1)*1.01.2041*V inf(k-1)*1.012 + m dot supp(k)*1.012);end

figure plot(T z(startt+1:60*(startt index+1),1),'k'); hold on plot(Tz pred(startt+1:60*(startt index+1))); The MATLAB function "zone F2 objective fn case1" used in the above code is as follows: function cost = zone F2 objective fn case1 (param) % Part 1: Extract optimization variables UA = param(1:9,1);C = param(10);% Part 2: Load measurement data load zone F2 data Tw meas = [T w1 T w2 T w3 T w3 sub1 T w3 sub2 T w4 T w5 T w5 sub1 T w6];% Part 3: Evolve the dynamics area1 = $98.4271 \times 82.0208 \times 0.3048^{2}$; $area2 = 98.4271*13.7812*0.3048^{2};$ area4 = $(72.1771-62.3333)*(7.875-2.95312)*0.3048^{2};$ $area5 = (16.4062 - 1.64062) * (7.875 - 2.95312) * 0.3048^{2};$ area3 = 82.0208*13.7812*0.3048^2 - area4-area5; $area6 = 82.0208 \times 13.7812 \times 0.3048^{2};$ area8 = (72.2344-3.28125)*(7.875-2.95312)*0.3048^2; $area7 = 98.4271 \times 13.7812 \times 0.3048^{2} - area8;$ area9 = 98.4271*82.0208*0.3048^2; areas = [area1 area2 area3 area4 area5 area6 area7 area8 area9]'; startt index = 0; startt = 60*startt index; Tz pred(startt+1,1) = T z(startt+1); for k = startt+2:60*(startt index+24)Tz pred(k,1) = (Q dot in(k) + m dot supp(k)*1.012*T supp(k) + C*T z(k-1)/60 + (Tw meas(k-1,:))*(UA) + 1.2041*V inf(k-1)*1.012*T amb(k-1))/(C/60 + sum(UA) + 1.2041*V inf(k-1)*1.012*T amb(k-1)*1.012*T amb(k-1))/(C/60 + sum(UA) + 1.2041*V inf(k-1)*1.012*T amb(k-1)*1.012*T amb(k-1)*1.01.2041*V inf(k-1)*1.012 + m dot supp(k)*1.012);end % Part 4: Calculate cost deviation = Tz pred(startt+1:60*(startt index+24),1) - T z(startt+1:60*(startt index+24),1); cost = norm(deviation);

Appendix G Codes for performing zone level identification as per optimization framework presented in (6.7) – (6.10)

To obtain the results presented in section 6.3.2 for the optimization framework corresponding to (6.7) - (6.10), the following steps are required. For the purposes of illustration, these steps are explained for Zone F2. For the other zones, the steps are the same and the appropriate codes are provided in the media accompanying this thesis.

STEPS 1-4: Same as Steps 1-4 in Appendix F. STEP 5: The following MATLAB code can then be run to perform the zone level optimization. clc clear all % Part 1: Generation of optimal parameter values area1 = $98.4271 \times 82.0208 \times 0.3048^{2}$; area2 = 98.4271*13.7812*0.3048^2; $area4 = (72.1771-62.3333)*(7.875-2.95312)*0.3048^{2};$ area5 = $(16.4062 - 1.64062) * (7.875 - 2.95312) * 0.3048^{2};$ area3 = 82.0208*13.7812*0.3048^2 - area4-area5; area6 = 82.0208*13.7812*0.3048^2; $area8 = (72.2344 - 3.28125) * (7.875 - 2.95312) * 0.3048^{2};$ area7 = 98.4271*13.7812*0.3048^2 - area8; area9 = 98.4271*82.0208*0.3048^2; areas = [area1 area2 area3 area4 area5 area6 area7 area8 area9]'; %matlabpool open 4 options = optimset('Algorithm','sqp','MaxFunEvals',100000,'MaxIter',100000,'UseParallel','always','TolFun ', 1e-4, 'TolX', 1e-12, 'TolCon', 1e-12); init param = [0.001;5000];UB = [0.01; 10000];LB = [0;0];optim param = fmincon(@zone F2 objective fn case2, init param, [], [], [], LB, UB, [], options); % call to % fmincon to obtain optimal parameter values using least squares optimization

simulation data h = optim param(1);C = optim param(2);load zone F2 data Tw meas = [T w1 T w2 T w3 T w3 sub1 T w3 sub2 T w4 T w5 T w5 sub1 T w6];startt index = 0; startt = 60*startt index; Tz pred(startt+1,1) = T z(startt+1); for k = startt+2:60*(startt index+24)Tz pred(k,1) = (Q dot in(k) + m dot supp(k)*1.012*T supp(k) + C*T z(k-1)/60 +(Tw meas(k-1,:))*(h*areas) + 1.2041*V inf(k-1)*1.012*T amb(k-1))/(C/60 + sum(h*areas) + 1.2041*V inf(k-1)*1.012*T amb(k-1)*1.012*T amb(k-1)/(C/60 + sum(h*areas) + 1.2041*V inf(k-1)*1.012*T amb(k-1)*1.012*T amb(k-1.2041*V inf(k-1)*1.012 + m dot supp(k)*1.012);end figure plot(T z(startt+1:60*(startt index+1),1),'k');hold on plot(Tz pred(startt+1:60*(startt index+1))); The MATLAB function "zone F2 objective fn case2" used in the above code is as follows: function cost = zone F2 objective fn case2 (param) % Part 1: Extract optimization variables h = param(1);C = param(2);% Part 2: Load measurement data load zone F2 data Tw meas = [T w1 T w2 T w3 T w3 sub1 T w3 sub2 T w4 T w5 T w5 sub1 T w6];% Part 3: Evolve the dynamics area1 = $98.4271 \times 82.0208 \times 0.3048^{2}$; $area2 = 98.4271*13.7812*0.3048^{2};$ area4 = (72.1771-62.3333)*(7.875-2.95312)*0.3048^2; $area5 = (16.4062 - 1.64062) * (7.875 - 2.95312) * 0.3048^{2};$ $area3 = 82.0208 \times 13.7812 \times 0.3048^{2} - area4 - area5;$ area6 = 82.0208*13.7812*0.3048^2; $area8 = (72.2344 - 3.28125) * (7.875 - 2.95312) * 0.3048^{2};$ $area7 = 98.4271 \times 13.7812 \times 0.3048^{2} - area8;$ area9 = 98.4271*82.0208*0.3048^2; areas = [area1 area2 area3 area4 area5 area6 area7 area8 area9]'; startt index = 0: startt = 60*startt index; Tz pred(startt+1,1) = T z(startt+1); for k = startt+2:60*(startt index+24)Tz pred(k,1) = (Q dot in(k) + m dot supp(k)*1.012*T supp(k) + C*T z(k-1)/60 +(Tw meas(k-1,:))*(h*areas) + 1.2041*V inf(k-1)*1.012*T amb(k-1))/(C/60 + sum(h*areas) + 1.2041*V inf(k-1)*1.012*T amb(k-1)*1.012*T amb(k-1)/(C/60 + sum(h*areas) + 1.2041*V inf(k-1)*1.012*T amb(k-1)/(C/60 + sum(h*areas) + 1.2041*V inf(k-1)*1.012*T amb(k-1))/(C/60 + sum(h*are

Appendix H

Codes for performing wall level identification as per optimization framework presented in section 6.3.3.1

To obtain the results presented in section 6.3.3.1 for the optimization framework corresponding to (6.12) - (6.20), the following steps are required. Note that these results are for the internal (zone F2 facing) surface of wall 34.

STEPS 1-2: Same as Steps 1-2 in Appendix F.

STEP 3: From the generic spreadsheet obtained in Step 2, columns containing the variables needed for the wall level identification are used to create a new spreadsheet called "sim_data_surf34_internal". This file is included in the media accompanying this thesis.

STEP 4: Simulation data contained in "sim_data_surf34_internal" is exported to the MATLAB workspace and saved in a MATLAB file called "surf34_internal_data" using the following MATLAB code:

% Import data for inside surface of wall 34 in MATLAB clc: clear all: T in = xlsread('sim data surf34 internal.xls','surf34 internal data','a2:a1441'); T out = xlsread('sim data surf34 internal.xls','surf34 internal data','b2:b1441'); T z = x lsread('sim data surf34 internal.xls','surf34 internal data','c2:c1441');T surf = xlsread('sim data surf34 internal.xls','surf34 internal data','d2:k1441'); save surf34 internal data STEP 5: The following MATLAB code can then be run to perform the wall level optimization. clc clear all % Part 1: Generation of optimal parameter values matlabpool open 4

options =

optimset('Algorithm','sqp','MaxFunEvals',1000000,'MaxIter',100000,'UseParallel','always','TolF un', 1e-4, 'TolX', 1e-12, 'TolCon', 1e-12); window, length = 24;

window_length = 24;

 $init_param = [1e2;1;1*ones(24*60,1)];$
UB = [1e4;100;5*ones(24*60,1)]; % the upper bound on the resistance is set to 100 here. It can be changed to 40 and 10 here for the other two cases presented in section 6.3.3.1. LB = [0;0;-5*ones(24*60,1)];optim param = fmincon(@id surf34 internal case1, init param, [], [], [], LB, UB, [], options)% call to % fmincon to obtain optimal parameter values using least squares optimization % Part 2: Comparison of wall temperature from least squares identification with EnergyPlus simulation data C in = optim param(1); R cond = optim param(2); d in = optim param(2+1:2+24*60); R in = 1/0.207; load surf34 internal data T in pred(1,1) = T in(1); for k = 2:60*(window length) T in pred(k,1) = T in pred(k-1) + $(60/C \text{ in})^*((T z(k-1)-T \text{ in pred}(k-1))/R \text{ in } + (T out(k-1))/R \text{ in } + (T$ 1)-T in pred(k-1))/R cond + d in(k-1)); end figure plot(T in(1:60*window length,1),'k'); hold on plot(T in pred(1:60*window length,1)); The MATLAB function "id surf34 internal case1" used in the above code is as follows: function cost = id surf34 internal case1 (param) % Part 1: Extract optimization variables window length = 24; C in = param(1); R cond = param(2); d in = param(2+1:2+24*60); R in = 1/0.207; % Part 2: Load measurement data load surf34 internal data % Part 3: Evolve the dynamics T in pred(1,1) = T in(1); for k = 2:60*(window length) T in pred(k,1) = T in pred(k-1) + $(60/C_in)*((T_z(k-1)-T_in_pred(k-1))/R_in + (T_out(k-1))/R_in + (T_out$ 1)-T in pred(k-1))/R cond + d in(k-1)); end % Part 4: Calculate cost deviation = T in pred(1:60*window length,1) - T in(1:60*window length,1); cost = norm(deviation);

Appendix I

Codes for performing wall level identification as per optimization framework presented in section 6.3.3.2

To obtain the results presented in section 6.3.3.2 for the optimization framework corresponding to (6.31) - (6.45), the following steps are required. Note that these results are for the internal (zone F2 facing) surface of wall 34.

STEP 1: In the EnergyPlus model, PRBS set-points are prescribed for all zones in the building. Also, the short-wave radiation incident on each wall surface is specified as an output variable. The EnergyPlus model (.idf file), hence modified, is included in the media accompanying this thesis.

STEP 2: A day-long simulation is run using the above EnergyPlus model and the generic spreadsheet containing the output variables is obtained.

STEP 3: From the above generic spreadsheet, columns containing the variables needed for the wall level identification for the internal surface of wall 34 are used to create a new spreadsheet called "sim_data_surf34_internal_with_loads". In this spreadsheet, the long-wave radiation incident on the wall surface is computed using the ScriptF factors obtained from EnergyPlus and the wall surface temperatures. The long-wave radiation and short-wave radiation values are added to give the total thermal load acting on the wall surface. This spreadsheet is also included in the media accompanying this thesis.

STEP 4: Simulation data contained in "sim_data_surf34_internal_with_loads" is exported to the MATLAB workspace and saved in a MATLAB file called "surf34_internal_data_with_loads" using the following MATLAB code:

clear all;

T_in = xlsread('sim_data_surf34_internal_with_loads.xls','a33:a73');

T_out = xlsread('sim_data_surf34_internal_with_loads.xls','b33:b73');

T_z = xlsread('sim_data_surf34_internal_with_loads.xls','c33:c73');

d in = 0.001*xlsread('sim data surf34 internal with loads.xls','o33:o73');

save surf34 internal data with loads

clear all % Part 1: Generation of optimal parameter values matlabpool open 4 options =optimset('Algorithm','sqp','MaxFunEvals',100000,'MaxIter',100000,'UseParallel','always','TolFun ', 1e-4, 'TolX', 1e-12, 'TolCon', 1e-12); window length = 23; init param = [1.1e3;1;1];UB = [2e3;10;10];LB = [0;0;0];optim param = fmincon(@id surf34 internal case2, init param, [], [], [], LB, UB, [], options); % call to fmincon to obtain optimal parameter values using least squares optimization % Part 2: Comparison of wall temperature from least squares identification with EnergyPlus simulation data C in = optim param(1); R cond = optim param(2); R rad = optim param(3); R in = 1/0.207; load surf34 internal data T in pred(1,1) = T in(1); for k = 2:60*(window length) d in(k-1) = (1/R rad)*(sum(T surf(k-1,:))-8*T in pred(k-1));T in pred(k,1) = T in pred(k-1) + $(60/C \text{ in})^*((T z(k-1)-T \text{ in pred}(k-1))/R \text{ in } + (T out(k-1))/R \text{ in } + (T$ 1)-T in pred(k-1))/R cond + d in(k-1)); end figure plot(T in(1:60*window length,1),'k'); hold on plot(T in pred(1:60*window length,1));

The MATLAB function "id_surf34_internal_case2" used in the above code is as follows: %

```
function cost = id_surf34_internal_case2 (param)
% Part 1: Extract optimization variables
window_length = 24;
C_in = param(1);
R_cond = param(2);
R_rad = param(2);
R_in = 1/0.207;
% Part 2: Load measurement data
load surf34_internal_data_with_loads
% Part 3: Evolve the dynamics
T_in_pred(1,1) = T_in(1);
for k = 2:60*(window length)
```

Appendix J

Codes to obtain LTI model of SCR building and perform agglomerative clustering (sections 6.3.4 – 6.4)

To perform the agglomerative clustering analysis presented in this section, the following programs need to be run in the specified sequence:

```
STEP 1: Obtain weighted incidence matrix, capacitance matrices, and state space matrices (see Algorithm 5.1). The spreadsheet "identified_parameters_mod.xlsx" contains the data obtained from the zone and wall level identification exercises performed in section 6.3. It is provided in the media accompanying this thesis %
```

```
% Part 1: Generation of A G (see Algorithm 5.1)
clc
clear all;
Nw = 103;
N_{z} = 9;
Nw int = 59;
resistance matrix = xlsread('identified parameters mod', 'consolidated', 'B2:BH11');
no col = size(resistance matrix, 2);
inside interface = resistance matrix(1,:);
outside interface = resistance matrix(2,:);
internal surface = resistance matrix(3,:);
external surface = resistance matrix(4,:);
R cond = resistance matrix(6,:);
R in = resistance matrix(9,:);
R out = resistance matrix(10,:);
A G = zeros(114, 114);
for i = 1:no col
  A G(inside interface(i), internal surface(i))=1/R in(i);
  A G(internal surface(i), inside interface(i))=1/R in(i);
  A G(internal surface(i), external surface(i))=1/R cond(i);
  A G(external surface(i), internal surface(i))=1/R cond(i);
  if (external surface(i)\sim=114)
     A G(external surface(i),outside interface(i)) = 1/R out(i);
     A G(outside interface(i), external surface(i)) = 1/R out(i);
  end
```

```
end
inv R inf =
1*(0.06*0.05)*[252.859;160.319;50.392;50.392;210.712;105.356;105.356;198.027];
% Implementation of modifications as per section 6.4.1.
A G(103+3,103+6) = inv R inf(1);
A G(103+6,103+3) = inv R inf(1);
A G(103+1,103+6) = inv R inf(2);
A G(103+6,103+1) = inv R inf(2);
A G(103+6,103+4) = inv R inf(3);
A G(103+4,103+6) = inv R inf(3);
A G(103+1,103+4) = inv R inf(4);
A G(103+4,103+1) = inv R inf(4);
A G(103+1,103+2) = inv R inf(5);
A G(103+2,103+1) = inv R inf(5);
A G(103+1,103+7) = inv R inf(6);
A G(103+7,103+1) = inv R inf(6);
A G(103+1,103+5) = inv R inf(7);
A G(103+5,103+1) = inv R inf(7);
A G(103+8,103+9) = inv R inf(8);
A G(103+9,103+8) = inv R inf(8);
% Part 2: Generation of L Gx
D G = diag(sum(A G,2));
L G = A G - D G;
L Gx = L G(1:Nw+Nz,1:Nw+Nz);
% Part 3: Generation of C w
C in = resistance matrix(7,:);
C out = resistance matrix(8,:);
c w = zeros(Nw,1);
for i = 1:no col
  c w(internal surface(i)) = C in(i);
  if (external surface(i)\sim =114)
    c w(external surface(i)) = C out(i);
  end
end
C w = diag(c w);
% Part 4: Generation of C z
c z = zeros(Nz,1);
c z(1) = 6181.45; %zone G
c z(2) = 7389.79; %zone E
c z(3) = 8889.79; %zone C
c z(4) = 706.02; %zone comp room
c z(5) = 3709.49; %zone F1
c z(6) = 7390.85; %zone D
c z(7) = 3716.90; %zone F2
c z(8) = 3449.18; %zone basement TC
```

```
c z(9) = 6498.85; %zone basement NTC
C z = diag(c z);
% Part 5: Generation of L a
L a = L G(1:Nw,Nw+Nz+1);
% Part 6: Generation of L Gg
L g = L G(1:Nw,end);
% Part 7: Generation of State Space matrices(continuous time)
A cont = (blkdiag(C w,C z))\L Gx;
B a cont = [C w\L a; zeros(Nz,1)];
B g cont = [C w\L g;zeros(Nz,1)];
B u cont = [zeros(Nw,Nz);inv(C z)];
B dw cont = [inv(C w); zeros(Nz, Nw)];
B dz cont = B u cont;
% Part 8: State space matrices in desired form and zone-wall matrix
A ww = A cont(1:Nw,1:Nw);
A wz = A cont(1:Nw,Nw+1:Nw+Nz);
A zw = A cont(Nw+1:Nw+Nz,1:Nw);
A zz = A cont(Nw+1:Nw+Nz,Nw+1:Nw+Nz);
B zd = B u cont(Nw+1:Nw+Nz,1:Nz);
ZW = zeros(Nz, Nw);
for i = 1:Nz
  for j = 1:Nw
    if (A G(Nw+i,j) \sim = 0)
      ZW(i,j) = 1;
      for k = 1:Nw
        if (A G(j,k) \sim = 0)
          ZW(i,k) = 1;
        end
      end
    end
  end
end
% Part 9: MPC Parameters
Ts = 60;
alphaa = 0.001;
N = 30;
% Part 10: Discretization
Ad ww = eye(Nw) + Ts*A ww;
Ad wz = Ts^*A wz;
Ad zw = Ts^*A zw;
Ad zz = eye(Nz) + Ts*A zz;
Bd zd = Ts*B zd;
save scr model
```

STEP 2: To run the agglomerative clustering, run the program in Step 3 of Appendix C with the following change: in the functions "find_new_olf" and "find_fpm", replace "load nine_zone_model" with "load scr_model".

Appendix K

Codes for cluster level control design (Fig. 6.30)

To perform the steps used in the cluster level control design shown in Figure 6.30, the following programs are used. It is recommended that this Appendix be read along with Appendices L and M.

STEP 1: Obtain weighted incidence matrix, capacitance matrices, and state space matrices (see Algorithm 5.1). The spreadsheet "identified_parameters_mod.xlsx" contains the data obtained from the zone and wall level identification exercises performed in section 6.3. It is provided in the media accompanying this thesis.

```
% Part 1: Generation of A_G (see Algorithm 5.1)
clc
clear all:
Nw = 103;
N_{z} = 9;
Nw int = 59;
resistance matrix = xlsread('identified parameters mod', 'consolidated', 'B2:BH11');
no col = size(resistance matrix, 2);
inside interface = resistance matrix(1,:);
outside interface = resistance matrix(2,:);
internal surface = resistance matrix(3,:);
external surface = resistance matrix(4,:);
R cond = resistance matrix(6,:);
R in = resistance matrix(9,:);
R out = resistance matrix(10,:);
A G = zeros(114, 114);
for i = 1:no col
  A G(inside interface(i), internal surface(i))=1/R in(i);
  A G(internal surface(i), inside interface(i))=1/R in(i);
  A G(internal surface(i), external surface(i))=1/R cond(i);
  A G(external surface(i), internal surface(i))=1/R cond(i);
  if (external surface(i)\sim =114)
    A G(external surface(i),outside interface(i)) = 1/R out(i);
    A G(outside interface(i), external surface(i)) = 1/R out(i);
```

end

end

```
% Implementation of modifications as per section 6.4.1.
inv R inf =
1*(0.06*0.05)*[252.859;160.319;50.392;50.392;210.712;105.356;105.356;198.027];
A G(103+3,103+6) = inv R inf(1);
A G(103+6,103+3) = inv R inf(1);
A G(103+1,103+6) = inv_R_inf(2);
A G(103+6,103+1) = inv R inf(2);
A G(103+6,103+4) = inv R inf(3);
A G(103+4,103+6) = inv R inf(3);
A G(103+1,103+4) = inv R inf(4);
A G(103+4,103+1) = inv R inf(4);
A G(103+1,103+2) = inv R inf(5);
A G(103+2,103+1) = inv R inf(5);
A G(103+1,103+7) = inv R inf(6);
A G(103+7,103+1) = inv R inf(6);
A G(103+1,103+5) = inv R inf(7);
A G(103+5,103+1) = inv R inf(7);
A G(103+8,103+9) = inv R inf(8);
A G(103+9,103+8) = inv R inf(8);
% Part 2: Generation of L Gx
D_G = diag(sum(A_G,2));
L_G = A_G - D_G;
L Gx = L G(1:Nw+Nz,1:Nw+Nz);
%Generation of C w
C in = resistance matrix(7,:);
C out = resistance matrix(8,:);
c w = zeros(Nw, 1);
for i = 1:no col
  c w(internal surface(i)) = C in(i);
  if (external surface(i)\sim=114)
    c w(external_surface(i)) = C_out(i);
  end
end
C w = diag(c w);
%Generation of C z
c z = zeros(Nz,1);
c z(1) = 6181.45; %zone G
```

```
c z(2) = 7389.79; %zone E
c z(3) = 8889.79; %zone C
c z(4) = 706.02; %zone comp room
c z(5) = 3709.49; %zone F1
c z(6) = 7390.85; %zone D
c z(7) = 3716.90; %zone F2
c z(8) = 3449.18; %zone basement TC
c z(9) = 6498.85; %zone basement NTC
C z = diag(c z);
% Part 3: Generation of L Ga
L Ga = L G(1:Nw,Nw+Nz+1);
% Part 4: Generation of L Gg
L Gw = L G(1:Nw,end);
% Part 5: Generation of State Space matrices(continuous time)
A_cont = (blkdiag(C_w,C_z)) L_Gx;
B a cont = [C w \setminus L Ga; zeros(Nz,1)];
B g cont = [C w \land L Gw; zeros(Nz, 1)];
B u cont = [zeros(Nw,Nz);inv(C z)];
B dw cont = [inv(C w); zeros(Nz, Nw)];
B dz cont = B u cont;
ZW = zeros(Nz,Nw);
for i = 1:Nz
  for j = 1:Nw
    if (A G(Nw+i,j) \sim = 0)
      ZW(i,j) = 1;
      for k = 1:Nw
        if (A G(j,k) \sim = 0)
          ZW(i,k) = 1;
        end
      end
    end
  end
end
save fo model
```

STEP 2: Obtain the parameters for design of controller and observer at the cluster level corresponding to any decentralized architecture

% Part 1: Input cluster information for fully decentralized case. For other architectures, the

% code can be modified appropriately

n_clust = 9;

for i = 1:9

struct_clust_zones(i).value = i;

end

% Part 2: Compute parameters for implementation of controller and observer for each cluster % and store them as fields of appropriate data structures

for $i = 1:n_{clust}$

[struct_tfo_aggreg_A_cont(i).value,struct_tfo_aggreg_B_u_cont(i).value,struct_tfo_aggreg_B_a _cont(i).value,struct_tfo_aggreg_B_g_cont(i).value,struct_tfo_aggreg_B_hat_cont(i).value,struct _aggregates(i).value,struct_clust_walls(i).value,struct_ext_zones(i).value,Np,Nu,struct_S(i).valu e,struct_C_bar(i).value,struct_Q1(i).value,struct_T(i).value,struct_W1(i).value,struct_W2(i).valu e,struct_W3(i).value,struct_H(i).value,Ts,struct_A1(i).value,struct_A2(i).value,struct_A3(i).valu e,struct_A4(i).value,struct_A5(i).value,struct_A6(i).value,struct_A7(i).value,rho_a,cp_a,struct_T _supp(i).value,struct_M_bar_max(i).value,struct_C_bar_temp(i).value,struct_Q_max(i).value,str uct_L(i).value,struct_init_est(i).value] =

find_model_and_mpc_param_with_dist(struct_clust_zones(i).value);

end

% Part 3: Save the data structures generated above in the Matlab file "clust_info"

 $save \ clust_info \ struct_tfo_aggreg_A_cont \ struct_tfo_aggreg_B_u_cont$

struct_tfo_aggreg_B_a_cont struct_tfo_aggreg_B_g_cont struct_tfo_aggreg_B_hat_cont struct_Q1 struct_clust_walls Ts Np Nu struct_aggregates c_w struct_clust_zones struct_S struct_C_bar struct_T struct_W1 struct_W2 struct_W3 struct_ext_zones struct_H Nw Nz struct_A1 struct_A2 struct_A3 struct_A4 struct_A5 struct_A6 struct_A7 rho_a cp_a struct_T_supp struct_M_bar_max struct_C_bar_temp struct_Q_max struct_L struct_init_est u prev = zeros(Nz,1);

x init est = [];

for i = 1:n clust

x_init_est = [x_init_est;struct_init_est(i).value]; % create a vector using initial state estimates % computed for each cluster

end

save u_values u_prev % these parameters are required later in the implementation of MPC % based decentralized controllers

The function "find_model_and_mpc_param_with_dist" used in the above code is shown below:

function[tfo_aggreg_A_cont,tfo_aggreg_B_u_cont,tfo_aggreg_B_a_cont,tfo_aggreg_B_g_cont,t fo_aggreg_B_hat_cont,aggregates,clust_walls,ext_zones,Np,Nu,S,C_bar,Q1,T,W1,W2,W3,H,Ts

```
A1,A2,A3,A4,A5,A6,A7,rho a,cp a,T supp,M bar max,C bar temp,Q max,L tfo aggreg,ini
t est] = find model and mpc param with dist(clust zones)
%SECTION 1: FINDING CLUSTER LEVEL MODEL (FULL ORDER)
load fo model
n clust zones = length(clust zones);
% Part 1: Determine walls of cluster
clust walls = [];
for i = 1:n clust zones
 local walls = find(ZW(clust zones(i),:)~=0);
 clust walls = union(clust walls,local walls);
end
n clust walls = length(clust walls);
% Part 2: Determine zones external to cluster
ext zones = [];
for i = 1:n clust walls
 local zones = find(ZW(:,clust walls(i)) \sim = 0);
 ext zones = union(ext zones,local zones);
end
ext zones = setdiff(ext zones,clust zones);
n clust ext zones = length(ext zones);
% Part 3: Determine connectivity among walls and zones inside cluster
clust A G = zeros(n clust walls + n clust zones, n clust walls + n clust zones);
for i = 1:n clust walls
  for j = 1:n clust walls
clust A G(i,j) = A G(clust walls(i), clust walls(j));
  end
  for k = n clust walls+1:n clust walls+n clust zones
    clust A G(i,k) = A G(clust walls(i),Nw+clust zones(k-n clust walls));
    clust A G(k,i) = A G(Nw+clust zones(k-n clust walls), clust walls(i));
  end
end
for i = n clust walls+1:n clust walls+n clust zones
  for j = n clust walls+1:n clust walls+n clust zones
  clust A G(i,j) = A G(Nw+clust zones(i-n clust walls),Nw+clust zones(j-n clust walls));
  end
end
```

```
250
```

% Part 4: Determine connectivity among walls and zones inside cluster and outside zones clust A hat = zeros(n clust walls, n clust ext zones); for i = 1:n clust walls for j = 1:n clust ext zones clust A hat(i,j) = A_G(clust_walls(i),Nw+ext_zones(j)); end end clust Az hat = zeros(n clust zones, n clust ext zones); for i = 1:n clust zones for j = 1:n clust ext zones clust Az hat(i,j) = A G(Nw+clust zones(i),Nw+ext zones(j));end end % Part 5: Determine connectivity of walls with ambient and ground clust B w a = zeros(n clust walls, 1);for i = 1:n clust walls clust B w a(i,1) = A G(clust walls(i),Nw+Nz+1); end clust B w g = zeros(n clust walls, 1);for i = 1:n clust walls clust B w g(i,1) = A G(clust walls(i),Nw+Nz+2); end % Part 6: Determine capacitance matrices clust c w = zeros(n clust walls, 1);for i = 1:n clust walls clust c w(i) = c w(clust walls(i),1); end clust c z = zeros(n clust zones, 1);for i = 1:n clust zones clust c z(i) = c z(clust zones(i), 1);end clust C w = diag(clust c w); clust C z = diag(clust c z);%SECTION 2: OBTAIN REDUCED ORDER CLUSTER LEVEL MODEL % Part 1: Identify internal wall aggregations aggregates = struct;

for $i = n_clust_walls+1:n_clust_walls+n_clust_zones$

```
local int aggregation = find(clust A G(i,1:n \text{ clust walls}) \sim = 0);
  aggregates(i-n clust walls).value = local int aggregation;
end
% Part 2: Identify external wall aggregations
cnt = n clust zones;
for i = n clust walls+1:n clust walls+n clust zones
  local ext aggregation = [];
  local int aggregation = aggregates(i-n clust walls).value;
  n local int aggregation = length(local int aggregation);
  for j = 1:n local int aggregation
    temp = find(clust A G(local int aggregation(j),1:n clust walls)\sim=0);
    if(norm(clust A G(temp,n clust walls+1:n clust walls+n clust zones))==0)
    local ext aggregation = [local ext aggregation,temp];
    end
  end
  if(isempty(local ext aggregation) == 0)
     cnt = cnt + 1;
    aggregates(cnt).value = local ext aggregation;
  end
end
% Part 3: Determine aggregated capacitances
n aggregations = length(aggregates);
aggreg c w = zeros(n aggregations, 1);
for i = 1:n aggregations
  aggreg c w(i) = sum(clust c w(aggregates(i).value));
end
aggreg C w = diag(aggreg c w);
% Part 4: Determine equivalent resistances between aggregated walls and cluster zones
aggreg A G = zeros(n aggregations+n clust zones, n aggregations+n clust zones);
for i = 1:n aggregations
  local i aggregation = aggregates(i).value;
  for j = 1:n aggregations
    local j aggregation = aggregates(j).value;
    summ = 0;
    for m = 1:length(local i aggregation)
       for n = 1:length(local j aggregation)
         summ = summ + clust A G(local i aggregation(m), local j aggregation(n));
       end
    end
```

```
aggreg A G(i,j) = summ;
  end
  for j = 1:n clust zones
    summ = 0;
    for k = 1:length(local i aggregation)
       summ = summ + clust A G(local i aggregation(k),n clust walls+j);
    end
    aggreg A G(i,n aggregations+j) = summ;
    aggreg A G(n \text{ aggregations}+j,i) = \text{summ};
  end
end
for i = 1:n clust zones
  for j = 1:n clust zones
    aggreg A G(n aggregations+i,n aggregations+j) =
clust A G(n clust walls+i,n clust walls+j);
  end
end
% Part 5: Determine equivalent resistances between aggregated walls and cluster zones and %
external zones
aggreg_A_hat = zeros(n_aggregations,n clust ext zones);
for i = 1:n aggregations
  local aggregation = aggregates(i).value;
  for j = 1:n clust ext zones
    summ = 0;
    for m = 1:length(local aggregation)
       summ = summ + clust A hat(local aggregation(m),j);
    end
    aggreg A hat(i,j) = summ;
  end
end
aggreg Az hat = clust Az hat;
% Part 6: Determine equivalent resistances between aggregated walls and ambient and
                                                                                         %
ground
aggreg B w a = zeros(n aggregations, 1);
aggreg B w g = zeros(n aggregations, 1);
for i = 1:n aggregations
  local aggregation = aggregates(i).value;
  summ1 = 0;
  summ2 = 0;
```

```
for m = 1:length(local aggregation)
    summ1 = summ1 + clust B w a(local aggregation(m));
    summ2 = summ2 + clust B w g(local aggregation(m));
  end
  aggreg_B_w_a(i,1) = summ1;
  aggreg B w g(i,1) = summ2;
end
% Part 7: Obtain continuous-time, reduced order state space model
aggreg D G =
sum([aggreg A G,[aggreg A hat;zeros(n clust zones,n clust ext zones)],[aggreg B w a;zer
os(n clust zones,1)],[aggreg B w g;zeros(n clust zones,1)]],2);
aggreg A cont = (blkdiag(aggreg C w,clust C z))/(aggreg A G-diag(aggreg D G));
aggreg B u cont = [zeros(n aggregations, n clust zones);inv(clust C z)];
aggreg B a cont = [aggreg C w\aggreg B w a; zeros(n clust zones, 1)];
aggreg B g cont = [aggreg C w\aggreg B w g;zeros(n clust zones,1)];
aggreg B hat cont = [aggreg C w\aggreg A hat; clust C z\aggreg Az hat];
% Part 8: Obtain continuous-time, reduced order, transformed, state space model
tfo aggreg A cont = 
[aggreg A cont,[zeros(n aggregations,n clust zones);inv(clust C z)];zeros(n clust zones,n a
ggregations+2*n clust zones)];
tfo_aggreg_B_u_cont = [aggreg B u cont;zeros(n clust zones)];
tfo aggreg B a cont = [aggreg B a cont; zeros(n clust zones, 1)];
tfo aggreg B g cont = [aggreg B g cont; zeros(n clust zones, 1)];
tfo aggreg B hat cont = [aggreg B hat cont;zeros(n clust zones,n clust ext zones)];
%SECTION 3: FINDING CLUSTER LEVEL MPC PARAMETERS
% Part 1: specify control and prediction horizons, sample time, penalties, upper and lower
% bounds
Np = 30;
Nu = Np/3;
Ts = 60;
gammaa temp = 1000*ones(n clust zones,1);
gammaa contrl =1*ones(n clust zones,1);
sigmaa = 0*ones(n_clust zones,1); % this parameter is not used
cp a = 1.05;
rho a = 1.02;
m max full = 1*[12.7426; 12.7426; 12.7426; 4.2475; 8.4951; 12.7426; 8.4951; 4.2475; 4.2475];
T supp full = 12.8*ones(9,1);
```

```
254
```

Q_max_full = 1*[140;140;140;50;70;140;70;50;50];

% Part 2: Model discretization

sysc =

ss(tfo_aggreg_A_cont,[tfo_aggreg_B_u_cont,tfo_aggreg_B_a_cont,tfo_aggreg_B_g_cont,tfo_ag greg_B_hat_cont],eye(n_aggregations+2*n_clust_zones),zeros(n_aggregations+2*n_clust_zones), n clust zones+2+n clust ext zones));

sysd = c2d(sysc, Ts, 'zoh');

tfo_aggreg_A_disc = sysd.A;

tfo_aggreg_B_disc = sysd.B;

tfo_aggreg_B_u_disc = tfo_aggreg_B_disc(:,1:n_clust_zones);

tfo_aggreg_B_a_disc = tfo_aggreg_B_disc(:,n_clust_zones+1);

tfo_aggreg_B_g_disc = tfo_aggreg_B_disc(:,n_clust_zones+2);

tfo_aggreg_B_hat_disc =

tfo_aggreg_B_disc(:,n_clust_zones+2+1:n_clust_zones+2+n_clust_ext_zones);

% Part 3: Model augmentation

 $A_mpc =$

[tfo_aggreg_A_disc,tfo_aggreg_B_u_disc;zeros(n_clust_zones,2*n_clust_zones+n_aggregations),eye(n_clust_zones)];

B_mpc = [tfo_aggreg_B_u_disc;eye(n_clust_zones)];

Ba_mpc = [tfo_aggreg_B_a_disc;zeros(n_clust_zones,1)];

Bg_mpc = [tfo_aggreg_B_g_disc;zeros(n_clust_zones,1)];

B_hat_mpc = [tfo_aggreg_B_hat_disc;zeros(n_clust_zones,n_clust_ext_zones)];

C_mpc=

[zeros(n_clust_zones,n_aggregations),eye(n_clust_zones),zeros(n_clust_zones),zeros(n_clust_zones),zeros(n_clust_zones),zeros(n_clust_zones),zeros(n_clust_zones),zeros(n_clust_zones),zeros(n_clust_zones),zeros(n_clust_zones),zeros(n_clust_zones)];

% Part 4: T,S,W,C_bar,Q1 and Q2 matrices

T = zeros((n_aggregations+3*n_clust_zones)*Np,n_aggregations + 3*n_clust_zones); for i = 1:Np

 $T((i-1)*(n_aggregations+3*n_clust_zones)+1:i*(n_aggregations+3*n_clust_zones),:) = (A mpc)^i;$

end

```
S = zeros((n_aggregations+3*n_clust_zones)*Np,n_clust_zones*Nu);
```

for i = 1:Np

for j = 1:Nu

if i-j < 0

 $S((i-1)*(n_aggregations+3*n_clust_zones)+1:i*(n_aggregations+3*n_clust_zones),(j-1)*n_clust_zones+1:j*n_clust_zones) =$

zeros((n_aggregations+3*n_clust_zones),n_clust_zones);

```
elseif i-j == 0
        S((i-1)*(n \text{ aggregations}+3*n \text{ clust zones})+1:i*(n \text{ aggregations}+3*n \text{ clust zones}),(j-1)
1)*n clust zones+1:j*n clust zones) = B mpc;
     elseif i-j > 0
       S((i-1)*(n aggregations+3*n clust zones)+1:i*(n aggregations+3*n clust zones),(j-
1)*n clust zones+1:j*n clust zones) = A mpc^(i-j)*B mpc;
     end
  end
end
W1 = zeros((n aggregations+3*n clust zones)*Np,Np);
for i = 1:Np
  for j = 1:Np
     if i-j < 0
        W1((i-1)*(n \text{ aggregations}+3*n \text{ clust zones})+1:i*(n \text{ aggregations}+3*n \text{ clust zones}),j) =
zeros((n aggregations+3*n clust zones),1);
     elseif i-j == 0
       W1((i-1)*(n \text{ aggregations}+3*n \text{ clust zones})+1:i*(n \text{ aggregations}+3*n \text{ clust zones}),j) =
Ba mpc;
     elseif i - j > 0
       W1((i-1)*(n \text{ aggregations}+3*n \text{ clust zones})+1:i*(n \text{ aggregations}+3*n \text{ clust zones}),j) =
A mpc(i-j)*Ba mpc;
     end
  end
end
W2 = zeros((n aggregations+3*n clust zones)*Np,Np);
for i = 1:Np
  for j = 1:Np
     if i - j < 0
       W2((i-1)*(n aggregations+3*n clust zones)+1:i*(n aggregations+3*n clust zones),j) =
zeros((n aggregations+3*n clust zones),1);
     elseif i-j == 0
       W2((i-1)*(n aggregations+3*n clust zones)+1:i*(n aggregations+3*n clust zones),j) =
Bg mpc;
     elseif i - j > 0
       W2((i-1)*(n aggregations+3*n clust zones)+1:i*(n aggregations+3*n clust zones),j) =
A mpc^(i-j)*Bg mpc;
     end
  end
end
```

```
W3 = zeros((n aggregations+3*n clust zones)*Np,n clust ext zones*Np);
for i = 1:Np
  for j = 1:Np
    if i-j < 0
       W3((i-1)*(n aggregations+3*n clust zones)+1:i*(n aggregations+3*n clust zones),(j-
1)*n clust ext zones+1:j*n clust ext zones) =
zeros((n aggregations+3*n clust zones),n clust ext zones);
     elseif i-j == 0
       W3((i-1)*(n aggregations+3*n clust zones)+1:i*(n aggregations+3*n clust zones),(i-
1)*n clust ext zones+1:j*n clust ext zones) = B hat mpc;
     elseif i - j > 0
       W3((i-1)*(n aggregations+3*n clust zones)+1:i*(n aggregations+3*n clust zones),(j-
1)*n clust ext zones+1:j*n clust ext zones) = A mpc^(i-j)*B hat mpc;
     end
  end
end
C bar = zeros(Np*2*n clust zones,Np*(n aggregations+3*n clust zones));
for i = 1:Np
  C bar((i-1)*2*n clust zones+1:i*2*n clust zones,(i-
1)*(n aggregations+3*n clust zones)+1:i*(n aggregations+3*n clust zones)) = C mpc;
end
gammaa = [gammaa temp;gammaa contrl];
Q1 = zeros(Np*2*n clust zones,Np*2*n clust zones);
for i = 1:Np
  Q1((i-1)*2*n \text{ clust zones}+1:i*2*n \text{ clust zones},(i-1)*2*n \text{ clust zones}+1:i*2*n \text{ clust zones})
= diag(gammaa);
end
Q2 = zeros(Nu*n clust zones,Nu*n clust zones);
for i = 1:Nu
  Q2((i-1)*n \text{ clust zones}+1:i*n \text{ clust zones},(i-1)*n \text{ clust zones}+1:i*n \text{ clust zones}) =
diag(sigmaa);
end
% Part 5: H matrix
H1 = S'*C \text{ bar}'*Q1*C \text{ bar}*S + Q2;
H = (H1 + H1')/2;
% Part 6: Constraint matrices
A1 = zeros(n clust zones*(Np+1), n clust zones*Nu);
for i = 1:Np+1
  for j = 1:Nu
```

```
if i - i \ge 0
      A1((i-1)*n clust zones+1:i*n clust zones,(j-
1)*n clust zones+1:j*n clust zones)=eye(n clust zones);
    else
      A1((i-1)*n \text{ clust zones}+1:i*n \text{ clust zones},(j-1)*n \text{ clust zones}+1:j*n \text{ clust zones})=
zeros(n clust zones);
    end
  end
end
m max = m max full(clust zones);
Q max = Q max full(clust zones);
T supp = T supp full(clust zones);
M max = diag(m max);
for i = 1:Np+1
  M bar max((i-1)*n clust zones+1:i*n clust zones,(i-1)*n clust zones+1:i*n clust zones)
= M max;
end
A4 = [zeros(n aggregations+3*n clust zones, Nu*n clust zones); S];
A2 = zeros((Np+1)*n clust zones, n clust zones);
for i = 1:Np+1
  A2((i-1)*n \text{ clust zones}+1:i*n \text{ clust zones},:) = eye(n \text{ clust zones});
end
A3 = [eye(n aggregations+3*n clust zones);T];
c temp = 
[zeros(n clust zones, n aggregations), eye(n clust zones), zeros(n clust zones, 2*n clust zones)
];
C bar temp = zeros((Np+1)*n clust zones,(Np+1)*(n aggregations+3*n clust zones));
for i = 1:Np+1
  C bar temp((i-1)*n clust zones+1:i*n clust zones,(i-
1)*(n aggregations+3*n clust zones)+1:i*(n aggregations+3*n clust zones)) = c temp;
end
A5 = [zeros(n aggregations+3*n clust zones,Np);W1];
A6 = [zeros(n aggregations+3*n clust zones,Np);W2];
A7 = [zeros(n aggregations+3*n clust zones, n clust ext zones*Np);W3];
%SECTION 4: FINDING OBSERVER GAINS and INITIAL CONDITIONS
gain mat = 0.5*(H(S'*C bar'*Q1'*C bar*T));
A cl = tfo aggreg A cont-tfo aggreg B u cont*gain mat(1:n clust zones,1:end-
```

STEP 3: Implement the decentralized controllers and observers whose parameters were computed in Step 2. This is done using appropriate MATLAB functions invoked real-time from the SIMULINK model. Instead of providing these codes here, they are provided in Appendix L where the Simulation framework used to perform the optimality analysis in chapter 6 is described.

Appendix L Codes for performing the optimality analysis (section 6.5.2.2)

The steps for performing the optimality analysis are listed below:

- 1. Execute the code shown in Step 1 in Appendix K.
- 2. Execute the code shown in Step 2 in Appendix K.. Note that this code must be modified to reflect the appropriate control architecture (centralized/knee3/knee2/knee1/fully decentralized) for which the results are desired.
- 3. Execute the program "disturbance_param.m" provided later in this section. This program sets the parameters used to speify the disturbance vectors "d_z" and "d_w" and the ambient temperature as per section 6.5.2.1.
- Run the SIMULINK model "output_feedback_decentralized_mpc" provided in the media accompanying this thesis. This model invokes the following MATLAB functions in realtime:
 - a. "find_amb": Provides the ambient temperature at each time instant
 - b. "disturbance_function": Creates the vectors "d_w" and "d_z" at each time instant as per section 6.5.2.1.
 - c. "temp_sensor": Introduces a fault in the thermostat of a desired zone. Make sure that the line "meas_temp(1) = 0" is commented out.
 - d. "decen_obsv": Implements the observers designed in Appendix K for each cluster in the architecture
 - e. "extract_eta1": Extract the vector of estimates of "eta_hat_1" for each cluster lumped into a giant vector
 - f. "extract_eta2": Extract the of estimates of "eta_hat_2" for each cluster lumped into a giant vector
 - g. "extract_zone_temp": Extract the vector of zone temperature estimates
 - h. "find_ref": Provide the zone set-point temperature at each time instant
 - i. "decen_mpc_with_dist": Implements the model predictive controllers designed in Appendix K for each cluster in the architecture
 - j. "actuator": Provides the option of introducing actuator faults. This option was not used in this thesis

The MATLAB programs and functions referenced above are shown below:

```
1. Program "disturbance param.m"
%matrix F
F = zeros(Nw);
mat G = xlsread('identified parameters mod','int wall dist','B1:K10');
for i = 2:size(mat_G,1)
  for j = 2:size(mat G,2)
    F(mat G(i,1),mat G(1,j)) = mat G(i,j);
  end
end
mat D = xlsread('identified parameters mod','int wall dist','B12:J20');
for i = 2:size(mat D,1)
  for j = 2:size(mat D,2)
    F(mat D(i,1),mat D(1,j)) = mat D(i,j);
  end
end
mat E = x lsread('identified parameters mod','int wall dist','B22:H28');
for i = 2:size(mat E,1)
  for j = 2:size(mat E,2)
    F(mat E(i,1),mat E(1,j)) = mat E(i,j);
  end
end
mat TC = xlsread('identified parameters mod','int wall dist','B30:J38');
for i = 2:size(mat TC,1)
  for j = 2:size(mat TC,2)
    F(mat TC(i,1),mat TC(1,j)) = mat TC(i,j);
  end
end
mat C = xlsread('identified parameters mod','int wall dist','B40:K49');
for i = 2:size(mat C,1)
  for j = 2:size(mat C,2)
    F(mat C(i,1),mat_C(1,j)) = mat_C(i,j);
  end
end
mat F1 = xlsread('identified parameters mod','int wall dist','B51:J59');
for i = 2:size(mat F1,1)
  for j = 2:size(mat F1,2)
```

```
F(mat F1(i,1),mat F1(1,j)) = mat F1(i,j);
  end
end
mat F2 = xlsread('identified parameters mod','int wall dist','B61:K70');
for i = 2:size(mat F2,1)
  for j = 2:size(mat F2,2)
     F(mat F2(i,1),mat F2(1,j)) = mat F2(i,j);
  end
end
mat SR = xlsread('identified parameters mod','int wall dist','B72:H78');
for i = 2:size(mat SR,1)
  for j = 2:size(mat SR,2)
     F(mat SR(i,1),mat SR(1,j)) = mat SR(i,j);
  end
end
mat NTC = xlsread('identified parameters mod','int wall dist','B80:L90');
for i = 2:size(mat NTC,1)
  for j = 2:size(mat NTC,2)
     F(mat NTC(i,1),mat NTC(1,j)) = mat NTC(i,j);
  end
end
%areas
all walls =
[xlsread('identified parameters mod','consolidated','B4:BH4'),xlsread('identified parameters
mod', 'consolidated', 'B5:BH5')];
all areas =
[xlsread('identified parameters mod','consolidated','B6:BH6'),xlsread('identified parameters
mod', 'consolidated', 'B6:BH6')];
for i = 1:length(all walls)
  areas(all walls(i)) = all areas(i);
end
areas = areas(1:Nw);
%stephen boltzmann constant
sb const = 5.67e-8;
%ground,sky and ambient temp
T_gnd = xlsread('identified parameters mod','rad signals','B2:B7201');
T sky = xlsread('identified parameters mod','rad signals','C2:C7201');
T air = T gnd;
%view factors
```

```
temp mat1 = xlsread('identified parameters mod','ext wall dist','C2:F119');
for i = 1:118
 F gnd(temp mat1(i,1)) = temp mat1(i,2);
 F sky(temp mat1(i,1)) = temp mat1(i,3);
 F air(temp mat1(i,1)) = temp mat1(i,4);
end
F gnd = F gnd(1:Nw);
F sky = F gnd(1:Nw);
F air = F gnd(1:Nw);
%short wave radiations
q swr = zeros(7200, Nw);
temp mat2 = xlsread('identified parameters mod','rad signals','D1:AD7201');
for i = 1:size(temp mat2,2)
 q swr(:,temp mat2(1,i)) = temp mat2(2:end,i);
end
%schedules
0.05 0.05];
0.3 0.3 0.1 0.05];
0.5\ 0.4\ 0.4];
eta appl = ones(1,24);
%Nominal loads
N occ = xlsread('identified parameters mod','zone dist','B2:B10');
W light = xlsread('identified parameters mod','zone dist','C2:C10');
W appl = 0.5*xlsread('identified parameters mod','zone dist','D2:D10');
save dist param F areas sb const T gnd T sky T air F gnd F sky F air q swr eta occ
eta light eta appl N occ W light W appl
2. Function "find amb"
function Ta = find amb(t)
load dist param
Ta = T air(floor(t/60)+1);
end
```

```
3. Function "disturbance function"
function dist = disturbance function(u)
load dist param
Nw = 103;
N_{z} = 9;
wall temp = u(1:Nw);
zone temp = u(Nw+1:Nw+Nz);
curr time = u(Nw+Nz+1);
dw = zeros(Nw,1);
dz = zeros(Nz,1);
%Internal surfaces
for i = 1:Nw
  summ = 0;
  for j = 1:Nw
    summ = summ + F(i,j)*((273.15+wall temp(j))^4 - (273.15+wall temp(i))^4);
  end
  dw in(i) = sb const*areas(i)*summ;
end
%External surfaces
for i = 1:Nw
  if(F sky(i) = 1)
    factor ab = 0.7;
  else
    factor ab = 0.9;
  end
  dw_out(i) = sb_const*areas(i)*(F_gnd(i)*((273.15+T_gnd(floor(curr time/60)+1))^4 - 
(273.15+\text{wall temp}(i))^4) + F \text{ sky}(i)*((273.15+T \text{ sky}(floor(curr time/60)+1))^4 -
(273.15+\text{wall temp}(i))^4 + F air(i)*((273.15+T air(floor(curr time/60)+1))^4 -
(273.15+wall\_temp(i))^{4})+areas(i)*factor_ab*q swr(floor(curr time/60)+1);
end
%Determine hour in which time lies
dayy = floor(curr time/(24*3600))+1;
t bar = curr time-(dayy-1)*24*3600;
hourr = floor(t bar/3600)+1;
%Zones
for i = 1:Nz
```

```
\frac{100}{2} where \frac{100}{2} et a occ(hourr)*N occ(i) + 1000*et a light(hourr)*W light(i) +
1000*eta appl(hourr)*W appl(i);
  dz(i) = 100*eta occ(hourr)*N occ(i) + 1000*eta_light(hourr)*W_light(i) +
400*eta_appl(hourr)*eta occ(hourr)*N occ(i);
end
dw = (dw in + dw out)/1000;
dz = dz/1000;
dist = [dw'; dz];
end
4. Function "temp sensor":
function meas temp = temp sensor(u)
act temp = zeros(9,1);
act temp(1:9,1) = u(1:9);
curr time = u(10);
meas temp = act temp;
% meas temp(1) = 0;
5. Function "decen obsv":
function der estimates = decen obsv(u)
load clust info
%Obtain inputs
n clust = length(struct clust zones);
summ = 0;
for j = 1:n clust
 summ = summ + length(struct aggregates(j).value) + 2*length(struct clust zones(j).value);
end
n all states = summ;
all states = zeros(n all states, 1);
all states(1:n all states,1) = u(1:n \text{ all states});
all inputs = zeros(Nz,1);
all inputs(1:Nz,1) = u(n \text{ all states}+1:n \text{ all states}+Nz);
Ta meas = u(n \text{ all states}+Nz+1);
Tg meas = u(n \text{ all states}+Nz+2);
all err = zeros(Nz, 1);
all err(1:Nz,1) = u(n \text{ all states}+Nz+2+1:n \text{ all states}+Nz+2+Nz);
```

```
overall T hat = zeros(Nz,1);
overall T hat = u(n \text{ all states}+Nz+2+Nz+1:n \text{ all states}+Nz+2+Nz+Nz);
n clust = length(struct clust zones);
summ = 0;
der estimates = [];
for j = 1:n clust
    aggregates = struct aggregates(j).value;
    n aggregations = length(aggregates);
    clust zones = struct clust zones(j).value;
    n clust zones = length(clust zones);
    \mathbf{x} =
all states(summ+1:summ+length(struct aggregates(j).value)+2*length(struct clust zones(j).
value)):
    summ = summ +
length(struct aggregates(j).value)+2*length(struct clust zones(j).value);
    u = zeros(n clust zones, 1);
    u(1:n clust zones,1) = all inputs(clust zones);
    local err = zeros(n clust zones, 1);
    local err(1:n clust zones,1) = all err(clust zones);
    ext zones = struct ext zones(j).value;
    T hat = overall T hat(ext zones);
    der_x = struct_tfo_aggreg_A_cont(j).value*x + struct_tfo_aggreg_B_u_cont(j).value*u
+ struct_tfo_aggreg_B_a_cont(j).value*Ta_meas +
struct_tfo_aggreg_B_g_cont(j).value*Tg_meas +
struct tfo aggreg B hat cont(j).value*T hat + struct L(j).value*local err;
    der estimates = [der estimates; der x];
end
6. Function "extract eta1":
function all eta = extract eta1(u)
load clust info
all eta = [];
n clust = length(struct clust zones);
summ = 0;
for j = 1:n clust
    aggregates = struct aggregates(j).value;
    n aggregations = length(aggregates);
```

 $\mathbf{x} =$

```
u(summ+1:summ+length(struct aggregates(j).value)+2*length(struct clust zones(j).value));
   summ = summ +
length(struct aggregates(j).value)+2*length(struct clust zones(j).value);
   local eta = zeros(n aggregations, 1);
   local eta(1:n aggregations,1) = x(1:n aggregations);
   all eta = [all eta;local eta];
end
7. Function "extract eta2":
function zone dist = extract eta2(u)
load clust info
zone dist = zeros(Nz,1);
n clust = length(struct clust zones);
summ = 0;
for j = 1:n clust
   aggregates = struct aggregates(j).value;
   n aggregations = length(aggregates);
   clust zones = struct clust zones(j).value;
   n clust zones = length(clust zones);
   \mathbf{x} =
u(summ+1:summ+length(struct aggregates(j).value)+2*length(struct clust zones(j).value));
   summ = summ +
length(struct aggregates(j).value)+2*length(struct_clust_zones(j).value);
   local dist = zeros(n clust zones, 1);
   local_dist(1:n_clust_zones,1) =
x(n aggregations+n clust zones+1:n aggregations+n clust zones+n clust zones);
   zone dist(clust zones) = local dist;
end
8. Function "extract zone temp"
function zone temp = extract zone temp(u)
load clust_info
zone temp = zeros(Nz,1);
n clust = length(struct clust zones);
summ = 0;
```

```
for j = 1:n clust
   aggregates = struct aggregates(j).value;
   n aggregations = length(aggregates);
   clust zones = struct clust zones(j).value;
   n clust zones = length(clust zones);
   \mathbf{x} =
u(summ+1:summ+length(struct aggregates(j).value)+2*length(struct clust zones(j).value));
   summ = summ +
length(struct aggregates(j).value)+2*length(struct clust zones(j).value);
   local zone temp = zeros(n clust zones, 1);
   local zone temp(1:n \text{ clust zones}, 1) =
x(n aggregations+1:n aggregations+n clust zones);
   zone temp(clust zones) = local zone temp;
end
9. Function "find ref"
function ref = find ref(t)
sp = [26.7 * ones(1,6), 24 * ones(1,16), 26.7 * ones(1,2)];
%Determine hour in which time lies
dayy = floor(t/(24*3600))+1;
t bar = t - (dayy - 1) + 24 + 3600;
hourr = floor(t bar/3600)+1;
ref = sp(hourr);
10. Function "decen mpc with dist":
function u \text{ star} = \text{decen mpc} with \text{dist}(u)
load clust info
load u values
%Obtain parameters
n clust = length(struct clust zones);
summ = 0;
for j = 1:n clust
  summ = summ +
length(struct aggregates(j).value)+2*length(struct clust zones(j).value);
end
n all states = summ;
```

```
all states = zeros(n all states, 1);
all states(1:n all states,1) = u(1:n \text{ all states});
all inputs = zeros(Nz,1);
all inputs(1:Nz,1) = u(n \text{ all states}+1:n \text{ all states}+Nz);
Ta\_meas = u(n\_all\_states+Nz+1);
Tg meas = u(n \text{ all states}+Nz+2);
all ref = zeros(Nz,1);
all ref(1:Nz,1) = u(n all states+Nz+1+1+1:n all states+Nz+1+1+Nz);
current time = u(n \text{ all states}+Nz+1+1+Nz+1);
overall T hat = all ref;
if(mod(current time,Ts) == 0)
  summ = 0;
   for j = 1:n clust
     %assignments for each cluster
     aggregates = struct aggregates(j).value;
     n aggregations = length(aggregates);
     clust zones = struct clust zones(j).value;
     n clust zones = length(clust zones);
     \mathbf{x}\mathbf{0} =
all states(summ+1:summ+length(struct aggregates(j).value)+2*length(struct clust zones(j)
.value));
     summ = summ +
length(struct aggregates(j).value)+2*length(struct clust zones(j).value);
     u0 = zeros(n clust zones, 1);
     u0(1:n clust zones,1) = all inputs(clust zones);
     r temp = zeros(n clust zones,1);
     r temp(1:n clust zones,1) = all ref(clust zones);
```

 $S = struct_S(j).value;$

C_bar = struct_C_bar(j).value;

 $Q1 = struct_Q1(j).value;$

```
T = struct_T(j).value;
```

```
W1 = struct_W1(j).value;
```

```
W2 = struct_W2(j).value;
```

 $W3 = struct_W3(j).value;$

ext_zones = struct_ext_zones(j).value;

n_clust_ext_zones = length(ext_zones);

```
T_hat = overall_T_hat(ext_zones);
```

 $H = struct_H(j).value;$

 $A1 = struct_A1(j).value;$

A2 = struct A2(j).value;A3 =struct A3(j).value; A4 =struct A4(i).value; A5 = struct A5(j).value; $A6 = struct_A6(j).value;$ A7 =struct A7(j).value; T supp = struct T supp(j).value; M bar max = struct M bar max(j).value; Q max = struct Q max(j).value; C bar temp = struct C bar temp(j).value; %initial conditions, lifted disturbances and references x bar0 = [x0;u0];Ta bar = Ta meas*ones(Np,1); Tg bar = Tg meas*ones(Np,1); T hat bar = zeros(n clust ext zones*Np,1); for i = 1:Np T hat bar((i-1)*n clust ext zones+1:i*n clust ext zones,1) = T hat; end r = [r temp; zeros(n clust zones, 1)];%R matrix R = zeros(Np*2*n clust zones,1);for i = 1:Np R((i-1)*2*n clust zones+1:i*2*n clust zones,1) = r;end %F matrix F = S'*C bar'*Q1'*(C bar*T*x bar0 + C bar*W1*Ta bar + C bar*W2*Tg bar + C bar*W3*T hat bar - R); A constraint = [-(A1+rho a*cp a*M bar max*C bar temp*A4);(A1+0*rho a*cp a*M bar max*C bar t emp*A4)];b constraint = [-rho a*cp a*M bar max*(A2*T supp -C bar temp*(A3*x bar0+A5*Ta bar+A6*Tg bar+A7*T hat bar)) + A2*u0; 0*rho a*cp a*M bar max*(A2*T supp -C bar temp*(A3*x bar0+A5*Ta bar+A6*Tg bar+A7*T hat bar)) - A2*u0 + A2*Q max]; %Optimization options = optimset('display', 'off', 'Algorithm', 'activeset', 'MaxFunEvals', 100000, 'MaxIter', 100000, 'UseParallel', 'always', 'TolFun', 1e-4, 'TolX', 1e-12, 'TolCon', 1e-12);

```
init point = zeros(Nu*n clust zones,1);
   D u star lifted = quadprog(H,F,A constraint,b constraint,[],[],[],[],init point,options);
   %D_u_star_lifted = quadprog(H,F,[],[],[],[],[],[],init_point,options);
   %Extraction of optimal control input
   D_u_star = D_u_star_lifted(1:n_clust_zones);
   u star(clust zones) = D u star + u0;
   u prev(clust zones) = u star(clust zones);
  end
  save u values u prev
else
  u_star = u_prev;
end
\frac{1}{10} = 140;
end
11. Function "actuator":
function u z supp = actuator(u)
u act = zeros(9,1);
u_max = zeros(9,1);
u_{min} = zeros(9,1);
u act(1:9,1) = u(1:9);
u \max(1:9,1) = u(10:18);
u \min(1:9,1) = u(19:27);
u z supp = u_act;
```

Appendix M Codes for performing the robustness analysis (section 6.5.2.3)

The steps to perform the robustness analysis are the same as that for optimality analysis (Appendix L), the only difference being that the line "meas_temp(1) = 0" in the function "temp_sensor" should not be commented out.