

ADVANCING EMBEDDED AND EXTRINSIC SOLUTIONS FOR OPTIMAL
CONTROL AND EFFICIENCY OF ENERGY SYSTEMS IN BUILDINGS

A Dissertation

by

CHRISTOPHER JOSEPH BAY

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Bryan P. Rasmussen
Committee Members,	Daniel A. McAdams
	David E. Claridge
	Aniruddha Datta
Head of Department,	Andreas Polycarpou

August 2017

Major Subject: Mechanical Engineering

Copyright 2017 Christopher Joseph Bay

ABSTRACT

Buildings account for approximately 40% of all U.S. energy usage and carbon emissions. Reducing energy usage and improving efficiency in buildings has the potential for significant environmental and economic impacts. To do so, reoccurring identification of hardware and operational opportunities is needed to maintain building efficiency. Additionally, the development of controls that continually operate building systems and equipment at energy optimal conditions is required.

This dissertation provides contributions to both of the aforementioned areas, which can be divided into two distinct portions. The first presents the framework for the development of an automated energy audit process, termed Autonomous Robotic Assessments of Energy (AuRAE). The automation of energy audits would decrease the cost of audits to customers, reduce the time auditors need to invest in an audit, and provide repeatable audit processes with enhanced data collection. In this framework of AuRAE, novel, audit-centric navigational strategies are presented that enable the complete exploration of a previously unknown space in a building while identifying and navigating to objects of interest in real-time as well as navigation around external building perimeters. Simulations of the navigational strategies show success in a variety of building layouts and size of objects of interest. Additionally, prototypes of robotic audit capabilities are demonstrated in the form of a lighting identification and analysis package on a ground vehicle and an environmental baseline measurement package on an aerial vehicle.

The second portion presents the development and simulation of two advanced economic building energy controllers: one utilizes steady-state relationships for optimizing control setpoints while the other is an economic MPC method using dynamic models to optimize the same control setpoints. Both control methods balance the minimization of

utility cost from energy usage with the cost of lost productivity due to occupant discomfort, differing from standard building optimal control that generally addresses occupant comfort through setpoint limits or comfort measure constraints. This is accomplished through the development of component-level economic objective functions for each subsystem in the modeled building. The results show that utility cost and the cost of occupant productivity from optimal comfort can be successfully balanced, and even improved over current control methods. The relative magnitude of the cost of lost productivity is shown to be significantly higher than the cost of utilities, suggesting that building operators, technicians, and researchers should make maintaining occupant comfort a top priority to achieve the greatest economic savings. Furthermore, the results demonstrate that by using steady-state predictions, the majority of the performance gains produced with a fully dynamic MPC solution can be recovered.

ACKNOWLEDGMENTS

First I would like to express my profound gratitude to my advisor, Dr. Bryan Rasmussen, for his excellent mentorship and continued financial support. You have entertained all my questions and even supported my numerous, and sometimes divergent academic endeavors. Our conversations and your guidance are very much appreciated and will not soon be forgotten. I would also like to thank my committee members, Dr. Daniel McAdams, Dr. David Claridge, and Dr. Aniruddha Datta for their time and service on my committee.

Thank you to the many friends I have made in the lab: Chris Price, Rohit, Rawand, Trevor, Austin, Kaimi, Jorge, Chao, Erik, Edwin, Franco, and Priyam. Your willingness to discuss research and ideas will be sorely missed. As well as Philosophy Fridays and Terra Mystica any-days. You have each made your own impact in my life that I will carry with me. I would also like to thank the AggieE Challenge program and the undergraduate students that I had the opportunity to mentor.

Thank you to my mom and sisters for their never-ending support and contributions to the man that I am today. I love you all and I am grateful every day to call you family. Also, thank you to the countless others who have helped me along the way: educators, mentors, friends, and family-by-bond.

And last, thank you to my wife, Sydney. Without you this would all be for naught. You are my heart, my support, and I cannot express the extent of my gratitude for you. Thank you for joining me on this journey.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Dr. Bryan Rasmussen (advisor), Dr. Daniel McAdams of the Department of Mechanical Engineering, Dr. David Claridge of the Department of Mechanical Engineering, and Dr. Aniruddha Datta of the Department of Electrical Engineering.

Funding Sources

Graduate study was supported by a fellowship from Texas A&M University, funding from NIST Grant#60NANB12D209, funding from NSF Grant#CMMI-1563361, and funding from Aggie Challenge Program at Texas A&M University.

NOMENCLATURE

MPC	Model Predictive Control
DMPC	Distributed Model Predictive Control
E-MPC	Economic Model Predictive Control
LIDAR	Light Distance and Ranging
SLAM	Simultaneous Location and Mapping
PMV	Predicted Mean Vote
PPD	Predicted Percentage of Dissatisfied
LOP	Loss of Productivity
HVAC	Heating, Ventilation, and Cooling
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UV	Unmanned Vehicle
AuRAE	Autonomous Robotic Assessments of Energy
AHU	Air Handling Unit
VAV	Variable Air Volume
UBO	Utilities Business Office

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGMENTS	iv
CONTRIBUTORS AND FUNDING SOURCES	v
NOMENCLATURE	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	xii
LIST OF TABLES	xxi
 1. INTRODUCTION AND LITERATURE REVIEW	 1
1.1 Introduction	1
1.1.1 Advanced Management and Fault Detection	5
1.1.2 Advanced Building Controls	6
1.1.3 Contributions of This Dissertation	6
1.1.4 Chapter Outline	8
1.2 Background and Literature Review	8
1.2.1 Energy Assessments	8
1.2.2 Control of Energy Systems in Buildings	13
1.3 Outline of the Dissertation	20
 2. SUMMARY OF CONTRIBUTIONS	 24
2.1 Unifying Vision of Research	24
2.2 Energy Audits and Automation of the Audit Process	26
2.2.1 Navigational Strategies and Algorithms	29
2.2.2 Interior Navigational Algorithm and Simulation	31
2.2.3 Exterior Navigational Algorithm and Simulation	35
2.2.4 Development of Lighting Package	40
2.2.5 Development of Environment Sensing Package	43
2.3 Economic Control and Energy Optimization in Building Energy Systems	46
2.3.1 The Utilities Business Office Building	49

	Page
2.3.2 General Form of Component-Level Economic Objective Function	54
2.3.3 Economic Objective Function for Cost of Power Consumed by Fan	55
2.3.4 Economic Objective Function for Cost of Chilled Water Usage in AHU	57
2.3.5 Economic Objective Function for Cost of Discomfort in Zones . .	59
2.3.6 MATLAB and EnergyPlus Co-Simulation Design	66
2.3.7 Zone Temperature Models	69
2.3.8 Summary of Optimization	73
2.3.9 EnergyPlus and MATLAB Co-Simulation Results	74
2.3.10 Constant Occupancy Simulation	75
2.3.11 Random Occupancy Simulation	81
2.3.12 Random Occupancy with Optimal Setbacks Simulation	83
2.3.13 Priority Zone Case	86
2.4 Overall Comparison	89
 3. PAPER A: SIMULATION AND VALIDATION OF INTERIOR AND EXTE- RIOR NAVIGATIONAL STRATEGIES FOR AUTONOMOUS ROBOTIC AS- SESSMENTS OF ENERGY	 91
3.1 Synopsis	91
3.2 Introduction	91
3.3 Navigational Strategies	93
3.3.1 Interior Navigation	95
3.3.2 Exterior Navigation	97
3.4 Simulation Studies	100
3.4.1 Interior Navigation Simulation	101
3.4.2 Exterior Navigation Simulation	102
3.5 Algorithm Validation with Ground Robot Platform	103
3.5.1 Ground Robot Platform	103
3.5.2 Ongoing Efforts	105
3.6 Future Work	107
3.7 Conclusion	107
 4. PAPER B: AUTONOMOUS LIGHTING ASSESSMENTS IN BUILDINGS: PART 1 - ROBOTIC NAVIGATION AND MAPPING	 108
4.1 Synopsis	108
4.2 Introduction	108
4.3 Navigational Strategies	116
4.3.1 Interior Navigation	120

	Page
4.3.2 Exterior Navigation	122
4.4 Simulation Verification	129
4.4.1 Interior Navigation Simulation	129
4.4.2 Exterior Navigation Simulation	131
4.4.3 Transitioning Between Exterior and Interior Navigation	131
4.5 Future Work	132
4.6 Conclusion	134
5. PAPER C: AUTONOMOUS ROBOTIC BUILDING ENERGY AUDITS: DEMONSTRATED CAPABILITIES AND OPEN CHALLENGES	135
5.1 Synopsis	135
5.2 Introduction	135
5.3 Background	139
5.4 Overview of Autonomous Robotic Assessments of Energy	141
5.4.1 Navigational Strategies	143
5.4.2 Lighting Identification and Analysis	151
5.4.3 Baseline Environment Data Collection	160
5.5 Open Challenges	164
5.5.1 Open Challenges	164
5.5.2 Occupant Enumeration and Location	165
5.5.3 External Mapping and Identification	165
5.5.4 Cooperative Autonomous Audits	165
5.5.5 Automated Energy Simulation Software Support	166
5.5.6 Automated Report and Recommendation Generation	167
5.6 Conclusion	167
5.6.1 Contributors	168
6. PAPER D: STEADY-STATE PREDICTIVE OPTIMAL CONTROL OF BUILD- ING ENERGY SYSTEMS USING A MIXED ECONOMIC AND OCCUPANT COMFORT FOCUSED OBJECTIVE FUNCTION	169
6.1 Synopsis	169
6.2 Introduction	169
6.3 Background	174
6.4 Development of Economic Objective Function for Advanced Building Sys- tems Control	180
6.4.1 The Utilities Business Office	180
6.4.2 AHU Fan Economic Objective Function Development	185
6.4.3 AHU Chilled Water Economic Objective Function Development	188
6.4.4 Zone Occupant Comfort Economic Objective Function Development	191

	Page
6.5 Simulation Design	197
6.6 Results	203
6.6.1 UBO Simulation with Current Building Controls	203
6.6.2 Steady-State Optimal Control Simulation	206
6.6.3 Very Important Person (VIP) Simulation	208
6.7 Discussion	210
6.7.1 Future Directions	212
6.8 Conclusion	213
 7. PAPER E: DYNAMIC MODEL PREDICTIVE CONTROL VS STEADY-STATE PREDICTIVE OPTIMIZATION OF BUILDING ENERGY SYSTEMS	 214
7.1 Synopsis	214
7.2 Introduction	214
7.3 Background	218
7.4 Description of Simulated Design and Implementation of Control	224
7.4.1 The Utilities Business Office	224
7.4.2 AHU Fan Economic Objective Function	228
7.4.3 AHU Chilled Water Economic Objective Function	229
7.4.4 Zone Occupant Comfort Economic Objective Function	230
7.4.5 Simulation Design	232
7.4.6 Zone Temperature Models	234
7.4.7 Summary of Optimization	238
7.5 Results	239
7.5.1 Constant Occupancy Simulation	239
7.5.2 Random Occupancy Simulation	246
7.5.3 Random Occupancy with Optimal Setbacks Simulation	248
7.5.4 Priority Zone Case	250
7.6 Overall Comparison	252
7.7 Conclusions	253
 8. CONCLUSIONS	 256
8.1 Future Directions	257
 REFERENCES	 260
 APPENDIX A. UBO COMPONENT MODELS	 276
A.1 UBO Zone Models	276
A.2 UBO VAV Models	278
A.3 UBO Chilled Water Valve Fit	279

	Page
APPENDIX B. PMV APPROXIMATION	281
APPENDIX C. ZONE MODEL INPUT MANIPULATION	282
APPENDIX D. ENERGYPLUS INPUT FILE	283
APPENDIX E. MATLAB SIMULATION CODE	350
E.1 Steady-State Method Specific Code	350
E.2 Dynamic MPC Method Specific Code	364
E.3 Shared Code	386
APPENDIX F. MATLAB APOGEE INTERFACE	396
F.1 Interface Instructions	396
F.2 Datalogging Code	396

LIST OF FIGURES

FIGURE		Page
1.1	U.S. energy consumption and breakdown of energy source from 1908 to 2015 [1].	1
1.2	U.S. energy consumption and breakdown of energy used in different sectors [2].	2
1.3	Past and projected primary energy consumption for the U.S. buildings sector [2]. The vertical line represents when the data was collected.	3
1.4	Site energy consumption by end use for U.S. commercial buildings [2]. . .	4
1.5	The varying levels of energy audits as defined by the American Society of Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE) [3]. .	11
1.6	How MPC, or receding horizon control, typically functions.	16
2.1	Map created by ground robot platform during interior navigation.	30
2.2	The SRT-Target algorithm pseudo-code.	33
2.3	Results from a simulation of the SRT-Target algorithm. The UAV is shown as the star shape in blue, its path as the thin green line, objects of interest as red dots with their associated radius of discovery as rounded red rectangles, obstacles as black, open space as white, and unknown territory as grey [3].	34
2.4	The external navigation algorithm pseudo-code.	37
2.5	(a) The environment is divided into slices to decrease the amount of stored data. (b) An example of the robot's perception upon reaching an exterior corner. (c) A demonstration of how the robot navigates exterior corners. (d) An example of the robot's perception at a traversable gap.	38
2.6	Results from a simulation of the exterior navigational algorithm [3]. . . .	39
2.7	Ground robot platform prototype with lighting identification package installed.	41

FIGURE	Page
2.8 Preliminary results from the ground robot platform and lighting package prototype. The yellow spheres represent the centers of lights identified in the building (Evans Library at Texas A&M).	42
2.9 The UAV platform and sensing package prototype used to collect baseline environment measurements.	44
2.10 Preliminary mapping and CO ₂ measurement results from the environment sensing package and UAV platform.	45
2.11 Typical building HVAC system in commercial buildings.	47
2.12 How MPC, or receding horizon control, typically functions.	49
2.13 Zone layout for the Utilities Business Office (UBO) at Texas A&M University (figure from Apogee Software).	50
2.14 Rooftop Air Handling Unit (AHU) for the Utilities Business Office (UBO) at Texas A&M University (figure from Apogee Software).	52
2.15 Chilled water loop for the Utilities Business Office (UBO) at Texas A&M University (figure from Apogee Software).	52
2.16 Variable Air Volume (VAV) box in the Utilities Business Office (UBO) at Texas A&M University (figure from Apogee Software).	53
2.17 Process flow and current control implementation in the UBO.	54
2.18 Trend of the objective cost of the fan as the end static pressure changes.	57
2.19 Trend of the objective cost of the chilled water used by the AHU as the AHU discharge air temperature changes.	60
2.20 Relationship between PPD and PMV as calculated by Fanger [4].	61
2.21 Shape of Loss Of Productivity function.	63
2.22 The change in objective cost of the loss of productivity due to discomfort due to changes in zone air temperature during a 15 minute period.	65
2.23 SketchUp model of the Utilities Business Office (UBO) at Texas A&M University.	67

FIGURE	Page
2.24 Control hierarchy used in simulation of the UBO.	69
2.25 Zone 4 temperature prediction 30 minutes ahead using identified models. .	71
2.26 Zone temperatures for the steady-state method with constant occupancy. .	77
2.27 Zone temperatures for the MPC controller with constant occupancy. . . .	77
2.28 Zone 1 temperature for the MPC controller.	78
2.29 Mass flow of chilled water and the discharge air temperature for the steady-state method.	78
2.30 Mass flow of chilled water and the discharge air temperature for the MPC controller.	79
2.31 AHU air flow and end static pressure for the steady-state method.	79
2.32 AHU air flow and end static pressure for the MPC controller.	80
2.33 Zone temperatures for the steady-state method with random occupancy. .	82
2.34 Zone temperatures for the MPC controller with random occupancy.	82
2.35 Zone temperatures for the steady-state method with random occupancy and optimal zone setbacks.	84
2.36 Zone temperatures for the MPC controller with random occupancy and optimal zone setbacks.	84
2.37 Zone temperatures for the steady-state method with Zone 5 prioritized over other zones.	87
2.38 Zone 5 temperature with comfort limits as a prioritized zone using the steady-state method.	88
2.39 Zone temperatures for the MPC method with Zone 5 prioritized over other zones.	88
2.40 Zone 5 temperature with comfort limits as a prioritized zone using the MPC method.	89
3.1 The SRT-Target algorithm, previously presented in [5].	96

FIGURE	Page
3.2 The external navigation algorithm, based off of the use of instant goals described in [6]	98
3.3 The perceived environment divided up into sections for the exterior navigation algorithm.	99
3.4 Exterior navigation around a corner.	100
3.5 This figure shows the robot as it explores the interior and exterior of the building. The robot is shown in blue, the path of the robot in green, unexplored space in gray, obstacles in black, and free space in white. The tree-like nature of the interior navigational strategy can be seen in the top part of the figure.	101
3.6 Interior navigation simulations showing full exploration from different starting locations.	102
3.7 Exterior navigation simulations showing complete building navigation from different starting locations.	103
3.8 The ground robot platform used in experiment.	104
3.9 Exterior navigation around a building on Texas A&M's campus.	106
3.10 Map developed by ground robot platform for interior navigation.	106
4.1 Levels of energy audits, as defined by the American Society of Heating, Refrigeration and Air-Conditioning Engineers adapted from [7].	110
4.2 A diagram of the overall lighting audit information and process flow. The right portion of the figure is covered in this paper while the left greyed-out portion is covered in part 2.	114
4.3 Map created by ground robot platform during interior navigation.	117
4.4 The SRT-Target algorithm, previously presented in [5].	121
4.5 Flowchart showing the process of SRT-Target, the interior navigational algorithm.	122
4.6 The modified external navigation algorithm from [8], based off of the use of instant goals described in [6].	124

FIGURE	Page
4.7 The perceived environment divided up into sections for the exterior navigation algorithm [8].	125
4.8 Exterior navigation across a gap.	125
4.9 Performing an extended perception at a corner, not finding a gap. The grey slices indicate no obstacles detected.	127
4.10 Exterior navigation around a corner [8].	127
4.11 Flowchart showing the process of exterior navigational algorithm.	128
4.12 This figure shows the robot as it explores the interior of the building beginning in the top-left, then top-right, followed by bottom-left, and then bottom-right. The robot is shown in blue, the path of the robot in green, unexplored space in gray, obstacles in black, and free space in white. The tree-like nature of the interior navigational strategy can be seen as the exploration develops.	130
4.13 Exterior navigation simulation showing the progression as the robot travels around the building that has several large doors (gaps).	132
4.14 Ground robot currently under development.	133
4.15 UAV platform with lighting package currently under development.	134
5.1 Levels of energy audits, as defined by the American Society of Heating, Refrigeration and Air-Conditioning Engineers adapted from [7].	137
5.2 Pseudo-code for the SRT-Target indoor navigational algorithm [5].	145
5.3 Results from a simulation of the SRT-Target algorithm [3]. The UAV is shown as the star shape in blue, its path as the thin green line, objects of interest as red dots with their associated radius of discovery as rounded red rectangles, obstacles as black, open space as white, and unknown territory as grey.	146
5.4 Pseudo-code for the exterior navigational algorithm [3].	148

FIGURE	Page
5.5 (a) How the robot divides up the perceived environment. (b) An example of the robot reaching an exterior corner. (c) An example of the robot navigating an exterior corner. (d) An example of the robot determining there is a traversable gap [8].	149
5.6 Results from a simulation of the exterior navigational algorithm [3]. The UAV is shown as the star shape in blue, its path as the thin green line, obstacles as black, open space as white, and unknown territory as grey. . .	150
5.7 (a) The initial image of a light captured by the USB camera. (b) The thresholded image after applying Otsu's method. (c) The image after undergoing morphological operations. (d) Individual bulbs being identified using the contour detection function from OpenCV.	152
5.8 Results from lighting simulations showing illuminance levels depending on bulb type [9].	157
5.9 Ground robot platform prototype with lighting identification package installed.	158
5.10 Preliminary results from the ground robot platform and lighting package prototype. The yellow spheres represent the centers of lights identified in the building (large campus library).	159
5.11 The UAV platform and sensing package prototype used to collect baseline environment measurements.	161
5.12 Preliminary results from the environment sensing package.	163
6.1 U.S. energy consumption and breakdown of energy used in different sectors [2].	170
6.2 Past and projected primary energy consumption for the U.S. buildings sector [2]. The vertical line represents when the data was collected.	171
6.3 Site energy consumption by end use for U.S. commercial buildings [2]. . .	172
6.4 How MPC, or receding horizon control, typically functions.	176
6.5 Zone layout for the Utilities Business Office (UBO) at Texas A&M University.	181
6.6 Process flow and current control implementation for the UBO.	182

FIGURE	Page
6.7 Rooftop Air Handling Unit (AHU)) for the Utilities Business Office (UBO) at Texas A&M University.	183
6.8 Chilled water loop for the Utilities Business Office (UBO) at Texas A&M University.	184
6.9 Variable Air Volume (VAV) box in the Utilities Business Office (UBO) at Texas A&M University.	185
6.10 Trend of the objective cost of the fan as the end static pressure changes. .	188
6.11 Trend of the objective cost of the chilled water used by the AHU as the AHU discharge air temperature changes.	190
6.12 Relationship between PPD and PMV as calculated by Fanger [4].	191
6.13 Shape of Loss Of Productivity function.	193
6.14 The change in objective cost of the loss of productivity due to discomfort due to changes in zone air temperature during a 15 minute period.	196
6.15 SketchUp model of the Utilities Business Office (UBO) at Texas A&M University.	198
6.16 Control hierarchy used in simulation of the UBO.	200
6.17 Zone and outdoor air temperatures for the UBO using the current currently implemented methods with PMV optimal temperature setpoints.	204
6.18 Total air flow and end static pressure in the AHU for the UBO using the currently implemented control methods with PMV optimal temperature setpoints.	205
6.19 Chilled water flow and discharge air temperature for the UBO using the currently implemented control methods with PMV optimal temperature setpoints.	205
6.20 Zone and outdoor air temperatures for the UBO using the steady-state control method with PMV optimal temperature setpoints.	207
6.21 Total air flow and end static pressure in the AHU for the UBO using the steady-state control method with PMV optimal temperature setpoints. . .	207

FIGURE	Page
6.22 Chilled water flow and discharge air temperature for the UBO using the steady-state control method with PMV optimal temperature setpoints. . .	208
6.23 Zone and outdoor air temperatures for the UBO using the steady-state optimal control method with a VIP zone.	209
6.24 Zone and outdoor air temperatures for the UBO using the steady-state optimal control method with a VIP zone.	210
7.1 U.S. energy consumption and breakdown of energy used in different sectors [2].	215
7.2 Site energy consumption by end use for U.S. commercial buildings [2]. . .	216
7.3 How MPC, or receding horizon control, typically functions.	220
7.4 Zone layout for the Utilities Business Office (UBO) at Texas A&M University.	225
7.5 Rooftop Air Handling Unit (AHU)) for the Utilities Business Office (UBO) at Texas A&M University.	226
7.6 Chilled water loop for the Utilities Business Office (UBO) at Texas A&M University.	227
7.7 Variable Air Volume (VAV) box in the Utilities Business Office (UBO) at Texas A&M University.	228
7.8 Control hierarchy used in simulation of the UBO.	234
7.9 Zone 4 temperature prediction 30 minutes ahead using identified models. .	236
7.10 Zone temperatures for the steady-state method with constant occupancy. .	241
7.11 Zone temperatures for the MPC controller with constant occupancy. . . .	242
7.12 Zone 1 temperature for the MPC controller.	242
7.13 Mass flow of chilled water and the discharge air temperature for the steady-state method.	243
7.14 Mass flow of chilled water and the discharge air temperature for the MPC controller.	243

FIGURE	Page
7.15 AHU air flow and end static pressure for the steady-state method.	244
7.16 AHU air flow and end static pressure for the MPC controller.	244
7.17 Zone temperatures for the steady-state method with random occupancy. .	246
7.18 Zone temperatures for the MPC controller with random occupancy. . . .	247
7.19 Zone temperatures for the steady-state method with random occupancy and optimal zone setbacks.	248
7.20 Zone temperatures for the MPC controller with random occupancy and optimal zone setbacks.	249
7.21 Zone temperatures for the MPC method with Zone 5 prioritized over other zones.	251
7.22 Zone 5 temperature with comfort limits as a prioritized zone.	252
A.1 Fit of the data from the UBO for the discharge air temperature and chilled water valve position.	280
B.1 Fit of the PMV function with varying zone air temperature and relative humidity.	281

LIST OF TABLES

TABLE	Page
1.1 Breakdown of costs between traditional auditors and proposed Autonomous Robotic Assessments of Energy (AuRAE) system.	12
1.2 Common types and forms of MPC cost functions.	17
2.1 Summary of simulation results for five different layouts.	34
2.2 Regressions for Loss Of Productivity fit of PMV from [10].	62
2.3 Inputs and outputs for the generated models.	71
2.4 Annualized economic costs for constant occupation profiles.	81
2.5 Annualized economic costs for random occupation profiles and standard setback temperatures.	83
2.6 Annualized economic costs for random occupation profiles and optimal setback temperatures.	85
2.7 Summary of results of the simulations.	90
4.1 Breakdown of costs between traditional auditors and proposed AuRAE system.	112
4.2 Summary of simulation results for five different layouts. For each layout, five simulations were completed starting from five separate locations. . . .	130
5.1 Comparison of costs between auditors and automated audits [3].	139
6.1 Common types and forms of MPC cost functions.	176
6.2 Regressions for Loss Of Productivity fit of PMV from [10].	193
6.3 Inputs and outputs for the generated models.	202
6.4 Annualized economic costs of the different simulation scenarios.	211
7.1 Regressions for Loss Of Productivity fit of PMV from [10].	232
7.2 Inputs and outputs for the generated models.	236
7.3 Annualized economic costs for constant occupation profiles.	245

TABLE	Page
7.4 Annualized economic costs for random occupation profiles and standard setback temperatures.	247
7.5 Annualized economic costs for random occupation profiles and optimal setback temperatures.	250
7.6 Summary of results of the simulations.	253

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

Scarcity can be defined as the fundamental economic problem of having seemingly unlimited wants in a world of limited resources [11]. As humanity's population has continued to expand, scarcity of natural resources has become more and more of a pressing issue for current and future generations. Consider the area of energy: as society continues to develop, its energy requirements grow. In the U.S. alone, total energy consumption has nearly tripled over the last 65 years from 34.6 quadrillions Btus (quads) in 1950 to 97.4 quads in 2015 [12]. Of the energy consumed in the U.S., non-renewable energies still represent over 90% of energy sources, as can be seen in Figure 1.1. In response to growing energy needs and increased understanding of environmental impacts due to traditional energy sources, many nations have put forth specific renewable energy targets. These targets aim to reduce humanity's dependence on non-renewable energies and combat the growing

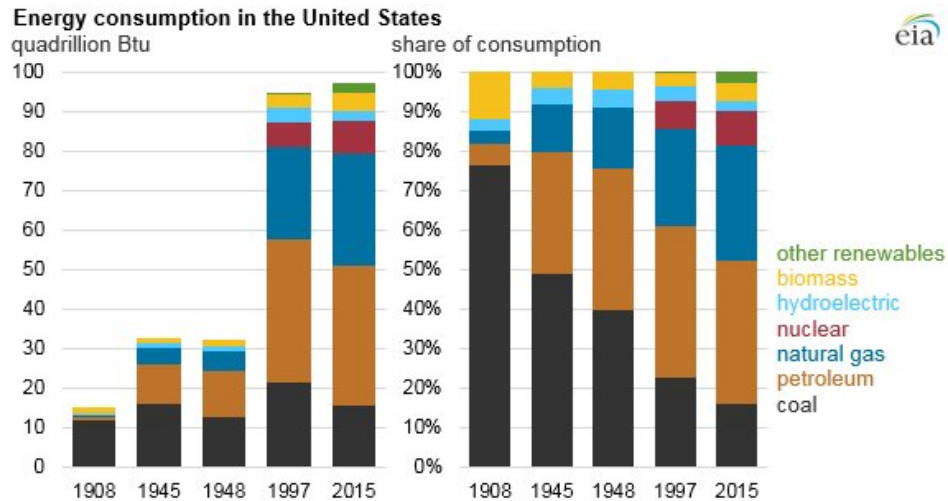


Figure 1.1: U.S. energy consumption and breakdown of energy source from 1908 to 2015 [1].

symptoms of climate change. For example, the European Union’s (EU) Renewable Energy Directive has established a binding goal of 20% final energy consumption from renewable sources by 2020 [13]. In the U.S., the Department of Energy has set a goal for the nation to have 20% of its electricity sourced from wind energy by the year 2030 [14]. While renewable sources will hopefully be increasing over the coming years, reductions in our energy usage can also play a significant role in alleviating the looming crisis of dwindling natural resources and global adverse environmental effects.

Focusing on the U.S., Americans consume 19% of the total world energy consumption, as shown in Figure 1.2. Delving into the energy consumption practices in the U.S., approximately 40% of all energy goes to building operations in the commercial and residential sectors [15]. Breaking the energy used in the commercial and residential building sectors down by source (right side of Figure 1.2), the data shows that approximately 75% comes from fossil fuels. As a result, energy usage in buildings account for 40% of the total U.S. carbon emissions [16]. Figure 1.3 shows that the buildings’ share of U.S. energy consumption has increased from approximately 34% in 1980 to the approximately 40% in 2010. Not only has it increased, but it is projected to continue growing over the

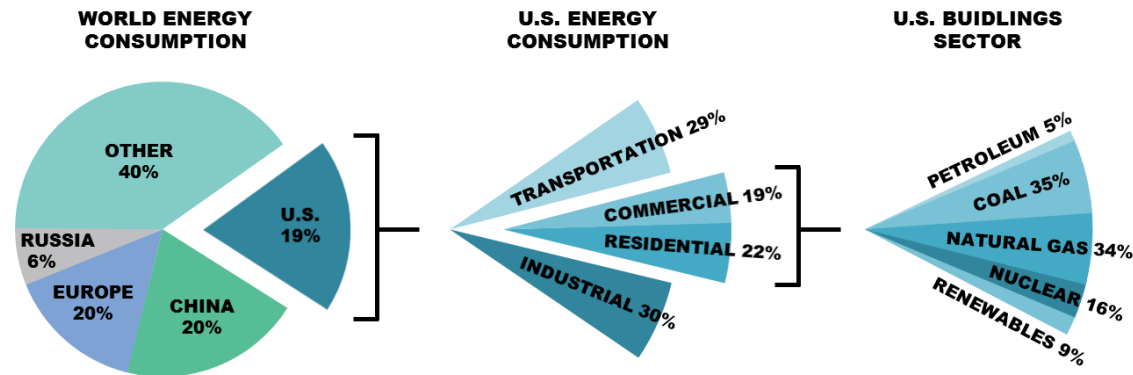


Figure 1.2: U.S. energy consumption and breakdown of energy used in different sectors [2].

next 20 years. Looking closer, Figure 1.3 shows that the commercial sector specifically has shown increased consumption percentage over time while the residential share has remained relatively constant. Using the right-side axis of Figure 1.3 along with the total U.S. consumption represented by the dashed line, one can calculate that the commercial buildings sector, at approximately 20%, consumes nearly 20 quads (20 quadrillion BTUs). For some perspective, one quad is approximately 293 billion kilowatt-hours, 8 billion gallons of gasoline, 38.5 million tons of coal, or nearly the energy used by 5.5 million U.S. homes at the time the data was taken (2010). Simply put, this is no small amount.

Investigating the commercial buildings sector further, the commercial site energy consumption can be broken down into specific end uses, as shown in Figure 1.4. Examining the top end uses reveals that space heating (27%), lighting (14%), space cooling (10%), water heating (7%), and ventilation (6%) are responsible for 64% of commercial building

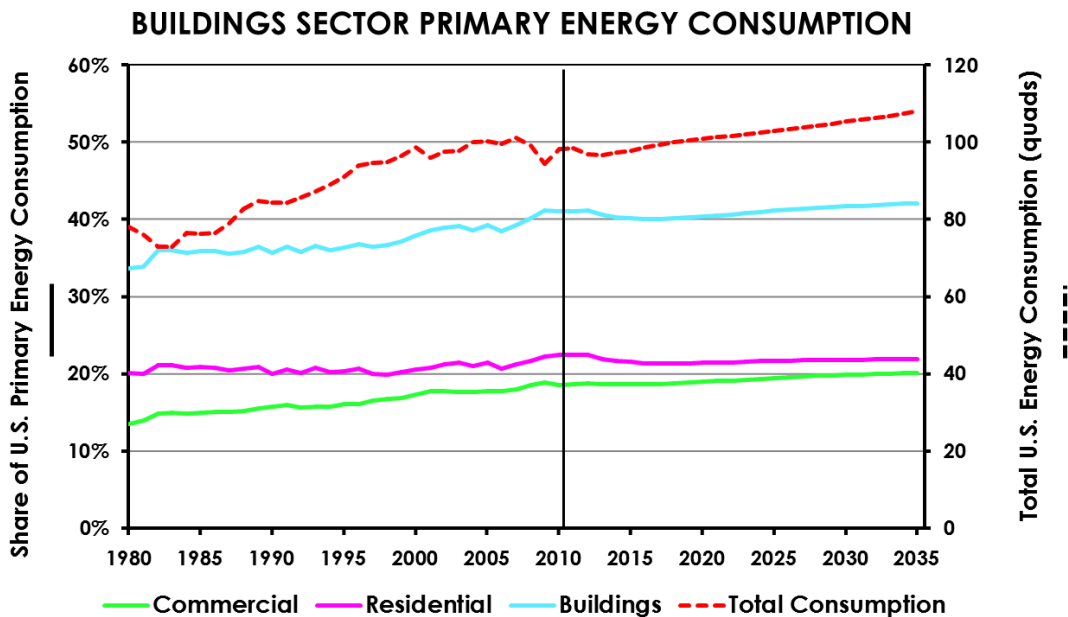


Figure 1.3: Past and projected primary energy consumption for the U.S. buildings sector [2]. The vertical line represents when the data was collected.

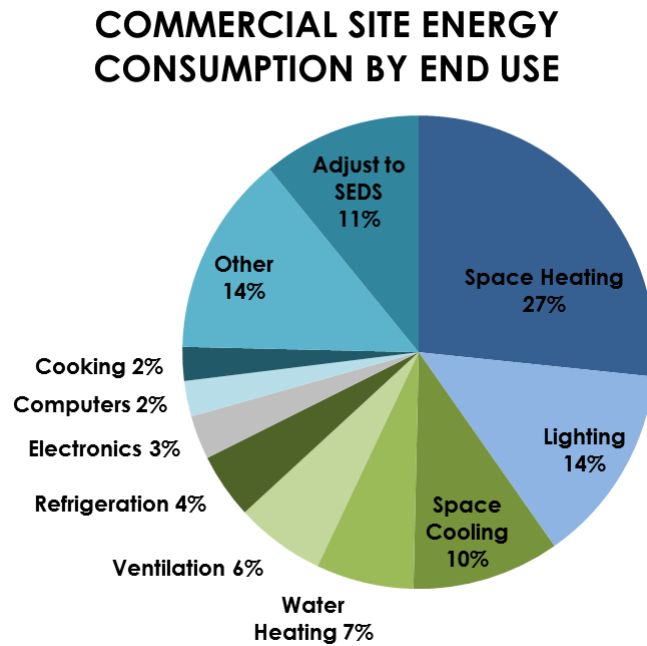


Figure 1.4: Site energy consumption by end use for U.S. commercial buildings [2].

energy usage. These categories can be combined more generally to refer to services required mostly when a building is occupied (space conditioning and lighting). The smaller categories and the all-encompassing “other” category represent 25%, while the other 11% is an adjustment that the Energy Information Administration uses to relieve discrepancies between data sources, specifically defined as “Energy attributable to the commercial buildings sector, but not directly to specific end-uses” [15]. To put this data in economic terms, in 2010 utilities cost businesses and building owners in the commercial sector \$179.4 billion. Not surprisingly, when this cost is broken down by use, the same five categories described at the beginning of this paragraph are at the top: lighting (\$35.4 billion), space heating (\$27.5 billion), space cooling (\$25.3 billion), ventilation (\$15.9) billion, and water heating (\$7.3 billion). Given the facts described above, buildings are a readily apparent and prime target for reductions in energy use and environmental impact.

Currently, there are three major trends in building energy research. One deals with advanced management and fault detection. Efforts include demand response and shedding, different occupant and reset strategies, as well as identifying equipment that is either tuned incorrectly or in need of repair or replacement. Another area of focus is on advanced building controls. Recent work has looked at coordinated control (centralized, decentralized, and distributed), predictive control, and cascaded control to improve a building's performance and operating efficiency. The third is the area of advanced hardware and technologies. Endeavors in thermo-electric heaters/coolers, micro-power generation at the residential and commercial level, as well as non-traditional de-humidification techniques are among the many proposed advancements and solutions. While there are many pathways for pursuing efficiency gains, this study focuses on two avenues that fall into the first and second described areas: energy audits/assessments and advanced controls of building energy systems.

1.1.1 Advanced Management and Fault Detection

To address the growing the needs of advanced building management and fault detection, energy audits and retro-commissioning have been used to help ensure buildings maintain efficient operation by identifying opportunities for equipment repair, upgrade, or changes in usage patterns, among other recommendations. These audits are performed by a team of trained energy auditors, and depending on the size of the facility being assessed and the thoroughness of the audit, the process can take several days to weeks. Recommendations given in a report at the end of an assessment serve to inform the owner/building manager about the cost and payback of specific retrofits and changes in operating behavior with the owner deciding which opportunities to pursue. These recommendations can vary with an auditor's experience and training, as well as a client's priorities. Although significant and worthwhile savings can be identified through an audit, majority of buildings

have not or will never receive an energy audit. One reason why companies don't perform audits is the cost, due to the significant number of personnel-hours required to complete the process. Through automating some or all of the audit tasks, the cost of an audit can be lowered thus encouraging more widespread use of audits as energy saving tools. Automation can also serve to decrease the variations in recommendations that arise from differing auditor experience and training.

1.1.2 Advanced Building Controls

As for control in building energy systems, current practice is to implement control with low-level controllers (proportional-integral or proportional-integral-derivative) in a decentralized fashion. In some cases, there may be some supervisory control, but mostly the systems operate independently. Because of the physically interconnected and complex nature of building systems, this uncoordinated control can often lead to inefficient solutions as controllers can compete with one another in achieving their desired outputs. Many advanced control strategies have been proposed, with Model Predictive Control (MPC) being a front-runner to address the challenges that building energy systems put forth. However, with MPC, the development of the objective function to be minimized is essential in the performance of the system. As of yet, an objective function that accounts for costs at the component level as well as the cost of occupant discomfort has not been proposed. Such objective functions would enable the identification of possible energy and cost savings at the component level, determining the economic importance of the different subsystems in a building. The results could serve as a guide to building energy managers and researchers as to where their efforts in research and development should be focused.

1.1.3 Contributions of This Dissertation

This dissertation provides contributions to both of the previously described areas which can be divided into two distinct parts. The first presents the framework for the develop-

ment of an automated energy audit process, termed Autonomous Robotic Assessments of Energy (AuRAE). The automation of energy audits would decrease the cost of audits to customers, reduce the time auditors need to invest in an audit, and provide repeatable audit processes with enhanced data collection. In this framework of AuRAE, novel, audit-centric navigational strategies are presented that enable the complete exploration of a previously unknown space in a building while identifying and navigating to objects of interest in real-time as well as navigation around external building perimeters. Simulations of the navigational strategies show success in a variety of building layouts and size of objects of interest. Additionally, prototypes of robotic audit capabilities are demonstrated in the form of a lighting identification and analysis package on a ground vehicle and an environmental baseline measurement package on an aerial vehicle.

The second part presents the development and simulation of two advanced economic building energy controllers: one utilizes steady-state relationships for optimizing control setpoints while the other is an economic MPC method using dynamic models to optimize the same control setpoints. Both control methods balance the minimization of utility cost from energy usage with the cost of lost productivity due to occupant discomfort, differing from standard building optimal control that generally addresses occupant comfort through setpoint limits or comfort measure constraints. This is accomplished through the development of component level economic objective functions for each subsystem in the modeled building. The results show that utility cost and the cost of occupant productivity from optimal comfort can be successfully balanced, and even improved over current control methods. The relative magnitude of the cost of lost productivity is shown to be significantly higher than the cost of utilities, suggesting that building operators, technicians, and researchers should make maintaining occupant comfort a top priority to achieve the greatest economic savings. Furthermore, the results demonstrate that by using steady-state predictions, majority of the performance gains produced with a fully dynamic MPC

solution can be recovered.

In these two efforts, building energy efficiency gains and economic savings are pursued. These two areas of research were chosen as no single topic can provide a complete solution to reducing energy usage in buildings; thus by investigating two distinct aspects, a greater impact may be had. Additionally, as these technologies become more fully developed, they have the potential to integrate into a more advanced solution as the robotic audit platforms can be used to continuously monitor structures, and through data collection, constantly improve the models of the advanced controls beyond the fixed measurement points that are currently available through building energy management systems. More details on the potential integration of these two research subjects can be found at the beginning of Chapter 2.

1.1.4 Chapter Outline

The rest of this chapter is organized as follows. First, background and an overview of the literature regarding energy assessments in buildings is given. Specifically, details about conducting energy audits will be discussed as well as the proven benefits. The case for automating energy audits will be made further. Then the general state of control in building energy systems will be presented, followed by an introduction to a leading candidate for advanced building controls, model predictive control (MPC). A summary of the relevant MPC in buildings literature will be given, focusing on the development and choice of objective functions. Lastly, an outline of the dissertation will be presented, as well as abstracts for the included research papers.

1.2 Background and Literature Review

1.2.1 Energy Assessments

Designed to operate efficiently when constructed, building performance can degrade over time. This degradation can be due to normal deterioration of systems, altered use or

misuse from the intended operation, or even gradual changes in climate and the surrounding environment. Moreover, new building technologies are being developed and made available at an increasing rate, but frequently these advances only appear in new construction and not existing buildings. As such, productivity losses and energy inefficiencies add up, resulting in increased cost to businesses and building owners. Specifically examining electricity, residential and commercial buildings accounted for 60.8%, or \$223.2 billion of the total U.S. electricity expenditures in 2010 [2]. With such a large financial factor, even small inefficiencies become significant. Beyond the economic cost resulting from these deficiencies, the energy used in buildings has a significant impact on the environment. As mentioned in the introduction, building operations account for approximately 40% of U.S. carbon emissions [16]. Commercial buildings alone account for half of the building energy usage [15] and building carbon emissions [17]. Any improvement in their energy consumption is welcome and worth pursuing.

Energy audits of buildings are a proven mechanism for achieving significant energy savings and reductions. Gaining initial popularity after the energy crises of the 1970s, energy audits have seen a recent resurgence in popularity as society has gained a better understanding of human impact on the environment as well as monetary incentives such as savings and recently created tax incentives. The savings from audits can be substantial; for example, small to medium sized manufacturers receiving one day audits and implementing suggested recommendations have resulted in an average of approximately 17% annual savings per client [18]. These facts notwithstanding, energy audits are still widely underused by commercial and industrial entities. While there is no national census detailing how many buildings could benefit from an energy audit, a U.S. industry survey estimated that below 5% of existing buildings have received audits [19]. Another study in the state of California estimated only 0.03% of existing buildings and 5% of newly constructed buildings have received audits [20].

One major barrier to the more widespread use of energy audits is cost. Assessments can be complex in nature and require significant personnel hours, extensive training of auditors over several years, and professional certification. Figure 1.5 shows a delineation of the varying levels of an energy audit, as defined by the American Society of Heating, Refrigeration and Air-Conditioning Engineers (ASHRAE). A preliminary energy-use analysis is the simplest in nature as the auditor examines utility bills and other energy use documents, performs some data analysis, and provides recommendations based on similar businesses or facilities. A Level 1 audit requires auditors to visit a site and perform a high-level walk through of the buildings, visually identifying energy saving measures and collecting limited data. This level of audit is usually limited to a day of on-site work, including travel. The next level audit, Level 2, is more involved, necessitating several auditors to gather many measurements and perform a complete, in-depth analysis of the energy usage. A Level 2 audit can last several days, depending on the size of the facility. The most comprehensive of all audits, a Level 3 audit, involves extensive measurement collection (both environmental and structural) to support software energy simulations of the buildings. These simulations, along with a more refined analysis and study of the data, produces the most tailored and inclusive set of possible energy saving recommendations. A Level 3 audit can last up to several weeks. With a significant portion of an auditor's time devoted to taking measurements, processing data, and choosing recommendations, any automation of these steps would lower the cost of an audit.

Another variable in the cost of energy audits is the experience training required of auditors. To become fully certified, auditors must attend specific training and courses, followed by a certification test, all which carry financial burdens. In addition to this official training, certified auditors are generally required to have a minimum of 3 years of experience before becoming certified [21]. The capital costs of auditor salary combined with training and certification constitute a substantial component of the cost of audits. Further-

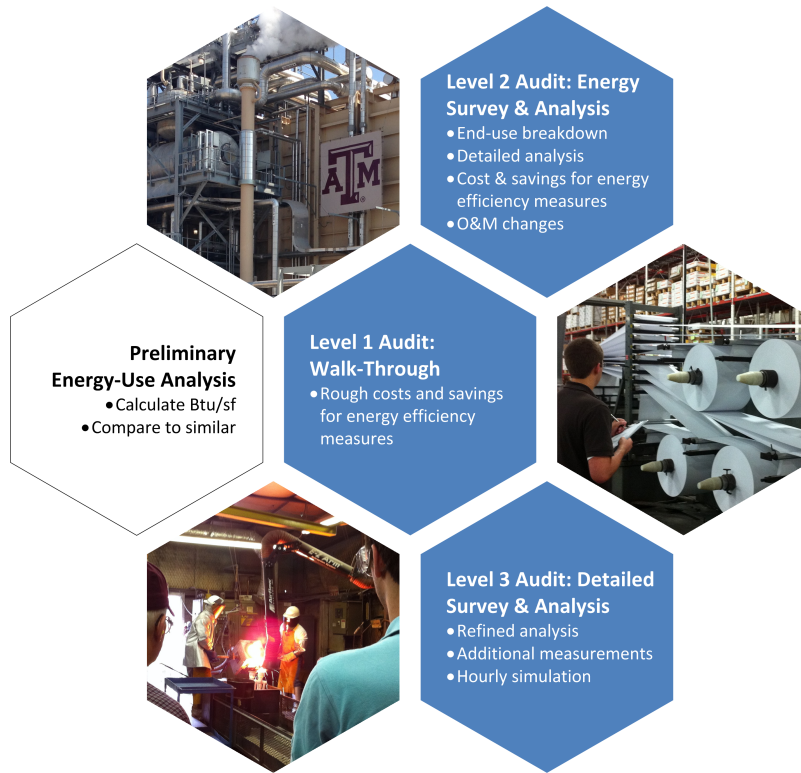


Figure 1.5: The varying levels of energy audits as defined by the American Society of Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE) [3].

more, experience can vary from auditor to auditor. Although auditors are subject to the same certification courses, in practice they can implement the audit procedures differently, resulting in variation of recommendations and potential savings. By automating the audit process, costs are greatly reduced and audits can be made more repeatable while providing increased richness of data. An estimation of the difference in costs is given in Table 1.1, below. An average annual salary of \$70,000 was used per discussion with auditors and auditing entities. The biggest savings can be seen in the capital investments made between auditors and automated audits. Additionally, the cost per audit is lower due to personnel and time savings.

Robots and prototype devices are increasingly being developed in research efforts as

Table 1.1: Breakdown of costs between traditional auditors and proposed Autonomous Robotic Assessments of Energy (AuRAE) system.

Auditors		AuRAE	
Training courses	\$15,000	UAV (x5)	\$6,000
Certification class	\$3,000	Technician training	\$1,500
Minimum 3 years experience (pay x2)	\$420,000	Sensors (x5)	\$15,000
Capital investments	\$438,000		\$22,000
Travel (2 auditors)	\$1,500	Travel (1 technician)	\$750
Cost of audit (20 hrs at \$60/hr)	\$2,400	Cost of audit (20 hrs at \$60/hr)	\$1,200
Cost per audit	\$3,900		\$1,950

tools for building inspection and auditing. Over recent years, the availability of economical robots and sensors has allowed for development of robotic solutions for new applications. Industrial inspection and maintenance has seen the development of serpentine robots [22] and climbing machines [23] that eliminate the need for workers to enter dangerous and difficult-to-reach areas. Much research has been completed concerning thermography of buildings. One group is developing a thermal indoor ground robot mapper [24]. Other researchers have attached thermal cameras to unmanned aerial vehicle (UAVs) to examine both the interior [25] and exterior [26, 27, 28] of buildings. Several groups have even worked to create 3D models suitable for energy simulation software by using a UAV equipped with a Lidar sensor [29] or just hand-held devices [30, 31]. Researchers in Europe have even proposed a model that simulates the behavior of a lighting assessment robot [32]. While research has been completed in automating building inspection tasks, much of it has centered on building thermography as that area has been the most accessible (relative low-cost and the availability of off-the-shelf components), leaving other tasks largely unexplored.

The first objective of this dissertation is to present the framework for an autonomous auditing platform, aimed at reducing the time and cost of energy audits. Prototypes are developed and presented, initially focused on performing lighting assessments and baseline measurement.

1.2.2 Control of Energy Systems in Buildings

Control of buildings presents several unique challenges. Buildings consist of numerous interconnected energy systems that affect and depend on one another. For example, in a large building there may be multiple chillers that are used to chill a secondary fluid, such as water. This chilled water is then pumped to various systems and areas of buildings where heat exchangers in air handling units (AHUs) use the chilled water to cool air streams. A network of fans and ducts then deliver the cooled air to the desired locations. The flow of this cooled air into the zones can be controlled by variable air volume (VAV) units, in which there may be another heat exchanger that utilizes heated water to warm the air, if necessary. The heated water for this process is provided by a different set of centralized pumps and heat exchangers. The zones themselves are connected to one another, either by conduction through barriers or shared doorways/open spaces. All these interconnections and couplings make for intriguing coordinated control problems with the potential for significant societal impact through increased energy efficiency.

Providing additional difficulties, nonlinear dynamics and multiple time scales occur across these various building systems. Equipment such as chilled water valves, fans, and dampers can have nonlinear behavior. For example, a nonlinear relationship can exist between air flow into a zone and damper position, dependent upon operating conditions and the damper used (single blade, opposed blade, etc.). End static pressure, often regulated by fan speed, depends on duct efficiencies and losses as well as zone damper positions. As for time scales, changes in damper position in a VAV or fan speed in an AHU are relatively fast

(on the order of seconds) compared to changes in desired chilled water temperatures (minutes) or changes in zone air temperature (hours). There are also slow, overarching changes to take into consideration such as the shift in solar loads as the sun moves throughout the day, gradual changes in outdoor air temperature due to change in weather or seasons, and the deterioration of equipment through use over time. These all contribute to shifting systems behaviors and disturbances that can cause undesired performance in systems. There are also discrete changes to take into account, such as whether an area/room is occupied, how many people are in said room, or changes in real-time pricing of utilities. In addition, the sensors of the system are distributed (not always equally), centralized monitoring is being performed, and devices are driven by localized controls. In addition to all of this, the systems must operate within constraints due to hardware limitations, limited resources, and/or issues of health and comfort.

Standard practice in building energy systems is to implement control with low-level controllers (proportional-integral, PI, or proportional-integral-derivative, PID) in a decentralized fashion. These controllers work to maintain building operator selected setpoints such as zone temperatures, AHU discharge air temperatures, duct pressures, chilled water supply and return temperatures, among others. In some cases, there may be supervisory control, but mostly the systems operate independently. Because of the physically interconnected and complex nature of building systems, this uncoordinated control can often lead to inefficient solutions as controllers can compete with one another in achieving their desired outputs. Additionally, building technicians are required to tune each loop, which can amount to several thousands of loops in some cases. For example, Texas A&M University has over 200,000 sensors and actuators distributed throughout approximately 200 buildings [33]. Because this number can be so large means that in most cases gains for the loops are either never changed from factory settings or marginally tuned at best. Current state-of-the-art buildings can include an energy management system (EMS). An EMS

can provide for more advanced control by utilizing additional sensors and communicating measurements to a centralized controller. They may provide some level of weather compensation resulting in reduced energy usage. Heating and cooling requirements can be coupled among buildings to produce more savings than optimizing just one building. They are also connected, enabling remote access and notifications for building technicians to deal with issues more quickly. While EMS allow for more coordination and make it easier to review performance, for the most part they still employ PI and PID controllers, requiring operators to tune systems. Additionally, EMS require regular adjusting of operating points to optimize performance and place the job of figuring out optimal control strategies on building operators.

While the challenges of building control are numerous, one control method that has emerged as a capable solution is model predictive control (MPC). MPC has been chosen as the most appropriate control method in various building thermal control research projects such as Opti-Control in Switzerland [34], Intelligent Buildings and Rational Management of Renewable Energy (MIGRER) in France [35], and MPC for UC Merced Campus in the U.S. [36]. MPC, or receding horizon control, predicts the change in the dependent variables of a modeled system by changing independent variables. Using the current state information, dynamic models of the system, and an objective function, MPC will determine the changes in the independent variables that will minimize the user-defined objective function while honoring given constraints on both dependent and independent variables. Once this series of changes is determined, the controller will apply the first determined control action, and then repeat the calculations for the next time step. Figure 1.6 displays how a typical reference tracking MPC implementation would behave. At time k the controller determines what the predicted output would be along with its optimal control trajectory. After completing the computations, the control would be applied and the system would move on to time $k + 1$, repeating the predictions and optimization with

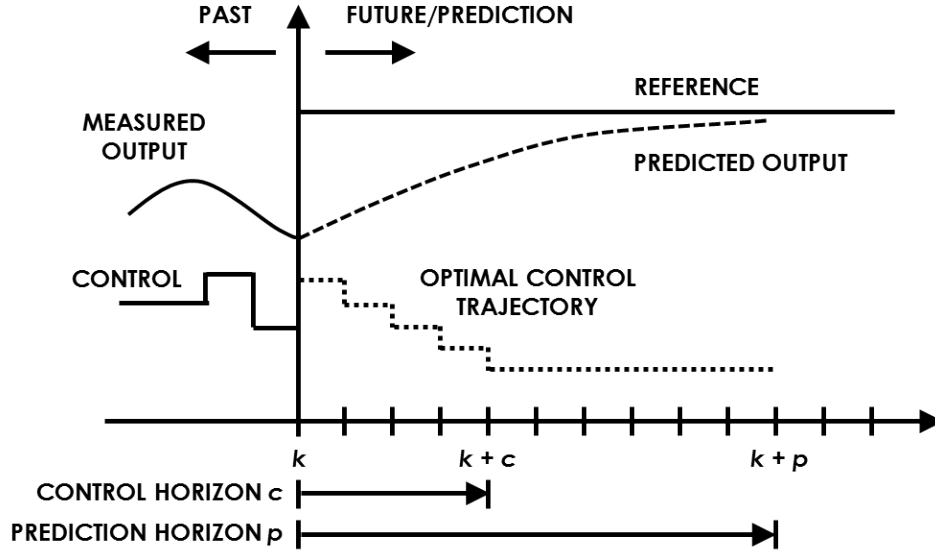


Figure 1.6: How MPC, or receding horizon control, typically functions.

the new measurements. Worth noting are the two horizons within MPC: the prediction horizon, which is the length of time for which the system outputs are predicted, and the control horizon, which is the number of control inputs that are determined in the prediction computation. The prediction horizon is often limited by computational capabilities, while the control horizon is selected so that the system dynamics are allowed to diminish [37]. Specific details about the derivation and function of MPC can be found in [38].

MPC in buildings has been studied mostly in simulation [39, 40, 41, 42, 43, 44, 45, 46, 47, 48], with a few experimental efforts [34, 35, 36, 49, 50, 51]. In simulation, MPC has been adapted for controlling building systems such as floor heating [52], water heating [53], cooling [36], and ventilation [54], among others. As MPC has been applied to such varied systems and structures, a variety of cost functions have been developed. The most common types of cost functions include quadratic cost, linear cost, and probabilistic cost [40]. The forms of these functions are shown in Table 1.2.

For most applications of MPC for buildings that have been reported in the literature,

Table 1.2: Common types and forms of MPC cost functions.

Cost Function Type	Mathematical Representation
Quadratic	$l_i(x_i, u_i) = x_i^T Q x_i + u_i^T R u_i$
Linear	$l_i(x_i, u_i) = c^T u_i$
Probabilistic	$l_i(x_i, u_i) = \mathbb{E}[q_i(x_i, u_i)]$

the objective function that is minimized focuses on energy usage, occupant comfort, or some combination of the two. This can take the form of economic MPC (E-MPC), where the objective function is a linear combination of the monetary cost of building energy consumption [43]. Generally in the applications where E-MPC is used, the amount of energy consumed is minimized while occupant comfort is maintained between upper and lower limits by constraints on the variables. In [46], a linear E-MPC cost function was used that had a time-varying cost of electricity vector for several actuators, which included positioning of blinds, the level of electrical lighting, chiller production, operation of the cooling tower, and heat from radiators. Occupant comfort was maintained by constraints placed on the room temperatures and the lighting levels. This proposed control strategy is able to respond to real-time changes in utility pricing; however, occupant comfort becomes a second priority temperature and lighting levels are constrained within specified limits as opposed to optimizing occupant comfort. Avci, et al. [55], also simulated a MPC controller that determined on/off control actions as well as reference temperatures for AC units to different zones. Their optimization leveraged real-time pricing of utilities and occupant preference to minimize usage of utilities and the difference between reference temperatures and room temperatures (error). In this effort, the authors developed an algorithm to choose the temperature reference setpoint based on the current utility cost and the occupants desired temperature range. While this helps to minimize cost to the user as well as maintain comfort within the user's defined range, this method does not provide a

minimal economical solution in that the cost depends on the user's own comfort decisions, which can either vary greatly or require updating with weather or seasonal changes.

The experiment detailed in [49] had a similar cost function, except that it was optimizing the cost of energy usage by manipulating the temperature setpoint for the water coming from the cooling tower as well as the temperature and mass flow setpoints of the water coming from the chillers. Again, occupant comfort was just maintained between upper and lower limits and not optimized. Papers [39, 41, 43, 50] all employed the linear E-MPC objective function while actuating heat flux, input power to a heat pump compressor, indoor air temperature setpoint and a thermal energy storage system, and temperature setpoints, respectively. The 4 previous works place constraints on the MPC optimization with temperature limits relating to comfort ranges, placing the optimization focus on control action and utility price. Oldewurtel performed simulations on six different combinations of subsystems including actuated blinds, electric lighting, radiators, mechanical ventilation, floor heating, evaporative cooling, and chilled ceilings while maintaining comfort between temperature limits [44]. Corbin, *et al.* [45] detailed two case studies in which the first used a linear E-MPC cost function while the second minimized the sum of electricity used by all HVAC equipment with a comfort penalty. This comfort penalty was defined as an area-weighted sum of the number of zone occupied hours outside of a predicted mean vote (PMV) threshold of ± 0.5 . PMV is an index that determines the thermal comfort of an average individual dependent upon a variety of factors, including air temperature, relative humidity, relative air velocity, metabolic rate, clothing insulation level, work output, and several other variables [4]. Corbin, *et al.* worked to optimize occupant comfort in their second case-study, but do so with the focus of reducing the cost of energy usage, neglecting the economic aspect associated with occupant comfort and productivity.

The authors of [47] also included a discomfort cost with their monetary energy cost that was based on different lower and upper thermal limits; however, the physical mean-

ing of this discomfort cost is arbitrary as the cost increases to unity until the temperature limits are exceeded and then becomes significantly large, not following any physical or measured relationship. The cost function in [48] included regulation of occupant comfort based on PMV, though it took the quadratic cost form, with the first term being the difference between the predicted PMV of the zone and the PMV setpoint for the zone, quantity squared, multiplied by a weighting factor. The second term consisted of the square of the change in control action, or increment, multiplied by a weighting factor. With this form, the MPC will balance maintaining the desired zone PMV while limiting large control action rates, in this case changes in water flow and air velocity. This will help keep occupants comfortable but the economic cost of the control actions is not accounted for. Morosan, *et al.* [42] used a linear objective function that penalizes the error between the predicted room temperature and the future room temperature reference as well as the energy usage to condition the room. In their efforts, comfort is accounted for as a comfort index that acts as a penalty when the room temperature does not meet its setpoint. However, in the presented simulations the temperature setpoints are arbitrarily chosen and not dependent on any comfort information. Additionally, this method, like the previous ones mentioned, does not account for the economic aspect of occupant comfort. One objective function from the literature that appears more unique than others was used in [51]. This objective function consisted of three different linear terms: 1) a weighting coefficient multiplied by the predicted percentage of dissatisfied (PPD) people, which PPD can be calculated from PMV, 2) a weighting coefficient multiplied by the summation of cost of energy consumed by the heating and cooling devices, and 3) a weighting coefficient multiplied by the summation of the green house gas intensities of the various energy sources (electricity and natural gas). This objective function displays the power of MPC to determine optimal control actions with respect to a user's desired metrics, in this case occupant comfort, monetary cost of energy, and environmental impact of energy sources. While providing great

flexibility in allowing the building operator to prioritize the three metrics with the weighting factors, the economic impact of the three areas can not be optimized due to differing units and arbitrary weights. Overall, previous methods have accounted for the economic cost of energy usage and/or attempted to maintain occupant comfort through optimization constraints or optimizing comfort itself, but none have accounted for the economic aspect of comfort on occupant productivity alongside utility costs.

To address this void, the second objective of this dissertation is to develop and present a control method that minimizes the economic cost of both utilities and occupant comfort. The monetization of occupant comfort occurs through the connection of an occupant's discomfort to loss of productivity. By evaluating energy cost and occupant comfort monetarily, the effect of each on overall cost can be identified and help guide future research and development efforts. In addition to the development an economic control method, the effectiveness of using steady-state predictions versus dynamic models is examined.

1.3 Outline of the Dissertation

So far the motivation behind this research effort has been presented in the introduction, Section 1.1, along with the relative background and general literature review in Section 1.2. The rest of this dissertation is written in such a format that includes peer-reviewed conference and journal papers. Some of these papers have been published while others will be submitted for publishing. These papers cover the main contributions relative to this research and supply additional details in explaining the techniques and results. The remainder of this dissertation will be organized as follows. Chapter 2 will give a summary of the contributions of this research. They have been grouped into the following categories: (i) autonomous robotic assessments of energy and (ii) economic control of building energy systems. Chapters 3 through 7 are the included research articles. And lastly, Chapter 8 provides conclusions about the research as well as a discussion of identified gaps for fu-

ture work. To aid the reader, an overview of each included research article is given below through a brief abstract.

Chapter 3 - Paper A [Bay et al., 2015]

Simulation and Validation of Interior and Exterior Navigational Strategies for Autonomous Robotic Assessments of Energy, published at the Dynamic Systems and Control Conference

Approximately 40% of all US energy usage and carbon emissions are attributed to buildings. Energy audits of buildings are an effective way to identify significant energy savings, but the extensive training of auditors and cost of the audits result in only a very small fraction of buildings receiving an audit. Automation of the audit process using robots can offer more detailed information for better recommendations and greatly reduce the cost of audits. This paper discusses navigational strategies that would be used by ground and aerial robots as they conduct automated energy audits. The strategies are split into two scenarios: interior and exterior navigation. Simulations for both the interior and exterior navigational algorithms are presented. Lastly, a ground robot platform is developed to validate the navigational strategies.

Chapter 4 - Paper B [Bay et al., 2016]

Autonomous Lighting Assessments in Buildings: Part 1 - Robotic Navigation and Mapping, published in Advances in Building Energy Research

Approximately 40% of all US energy usage and carbon emissions are attributed to buildings. Energy audits of buildings are an effective way to identify significant energy savings, but the extensive training required by auditors and cost of the labor intensive audits result in only a small fraction of buildings receiving an audit. Automation of the audit process using robots can offer more detailed information for better recommendations, greater consistency in analysis and recommendations, and greatly reduce the cost of audits. This paper introduces such a system and proposes navigational strategies that would be used by ground and aerial robots as they conduct automated energy audits. The strategies are divided into the interior and exterior environments. Simulations for both the interior and exterior navigational algorithms are presented, showing success in completely exploring previously unknown areas, identifying and maneuvering to objects of specific interest to energy audits, and circumnavigating open exterior perimeters of buildings.

Chapter 5 - Paper C [Bay et al., 2017]

Autonomous Robotic Building Energy Audits: Demonstrated Technology and Open Challenges, accepted for ASHRAE Transactions

Building operations are a significant consumer of energy and contributor to carbon emissions in the U.S. and around the globe. Energy audits offer significant potential in reducing building energy use by providing tailored recommendations involving equipment upgrades, operational adjustments, and building recommissioning. However, energy audits are a time intensive process that requires significant experience and training. This causes high costs related to performing an audit and prevents many businesses from having an audit completed. Automating the audit process will not only reduce the cost of audits, but clients will be provided with more repeatable and accurate recommendations based on improved data collection and analysis. Previous work and the current state of the art of robotic auditing tools are discussed in this paper, followed by open challenges and future possibilities of autonomous vehicles for conducting audits.

Chapter 6 - Paper D [Bay et al., 2017]

Steady-State Predictive Optimal Control of Building Energy Systems Using a Mixed Economic and Occupant Comfort Focused Objective Function, to be submitted

Control of energy systems in buildings is an area of increasing interest as the importance of energy efficiency and occupant comfort grows. The objective of this study is to demonstrate the effectiveness of a novel steady-state optimal control method in minimizing the economic costs associated with operating a building. Specifically, the cost of utility consumption and the cost of loss productivity due to occupant discomfort are minimized. This optimization is achieved through the use of steady-state predictions and component level economic objective functions. Specific objective functions are developed and linear models are identified from data collected from a building on Texas A&M University's campus. The building consists of multiple zones and is serviced by a variable air volume (VAV), chilled water air handling unit (AHU). The proposed control method is then co-simulated with MATLAB and EnergyPlus. Simulation results show improved comfort performance and decreased economic cost over the currently implemented controller, minimizing productivity loss and utility consumption. The potential for more serious consideration of the economic cost of occupant discomfort in building control design is discussed.

Chapter 7 - Paper E [Bay et al., 2017]

Dynamic Model Predictive Control vs Steady-State Predictive Optimization of Building Energy Systems, to be submitted

Optimal control of energy systems in buildings is a current area of interest; however, full implementation of Model Predictive Control (MPC) with dynamic prediction models becomes increasingly difficult as larger and more complex systems are controlled. The purpose of this paper is two-fold: 1) to introduce an economic MPC strategy that minimizes utility cost and the cost of loss productivity due to occupant discomfort, and 2) investigate the performance increase gained by using dynamic models over steady-state relationships to help determine when the added difficulty, communication burden, and computational cost is justified. A model is developed of a real building on Texas A&M University's campus in EnergyPlus. Dynamic MPC and steady-state optimization are implemented on the simulated building and co-simulation with the control implemented in MATLAB is performed. Operational cases including constant and random occupancy as well as standard and optimal setback temperatures are investigated. The proposed algorithm's ability to determine optimal setback temperatures as well as prioritize certain zone's comfort over others is demonstrated. It is found that the performance increase from dynamic models is marginal in some of the proposed cases and that steady-state predictions can provide similar performance results as a fully dynamic solution.

2. SUMMARY OF CONTRIBUTIONS

This chapter summarizes the different contributions from the five papers included in this dissertation (Papers A to E). Papers A and B have been published through peer-reviewed channels, Paper C is currently under peer-review, and Papers D and E will be submitted to appropriate journals. The papers are included in chronological order within the two research areas. This chapter is organized as follows. An overview of energy audits and automating the audit process will be given, followed by details regarding the development of Autonomous Robotic Assessments of Energy, including new navigational algorithms and preliminary results from audit package prototypes. Then a continuation of the discussion on control of energy systems in buildings will be presented. Economic model predictive control will drive the development of specific economic objective functions for systems used in a building on Texas A&M University's campus. These objective functions will be used in a steady-state optimal control method as well as a Model Predictive Control implementation. The chapter ends with results of simulations using the steady-state and MPC methods as well as a comparison in performance of the supervisory controls between using steady-state and dynamic models for predictions.

2.1 Unifying Vision of Research

The Building Services Research and Information Administration (BSRIA) recognized a need for identifying pathways for buildings to improve the productivity and well-being of occupants and the respective changes required in the industry [56]. To accomplish this, the BSRIA held a forum inviting approximately 30 senior executives that represented a mix of manufacturers, specifiers, and end users at an Air-Conditioning, Heating, Refrigeration (AHR) Exposition in Chicago in 2015. Discussions were held about the “changing landscape around the design and use of buildings, their evolution, and to uncover how

the HVAC and building controls industry should be shaped in the future in order to best respond to client needs” [56]. Some of the main conclusions were that there will be continued progress in and requirements for green construction and sustainability, driven by both environmental and economic factors. Also, buildings will need to adapt to people, and not the other way around as it has traditional been. There will be the desire for more automation. Driven by the Internet-of-Things and smart technology in general, more self-learning and self-identifying systems will become available. Lastly, products of the future should be designed with the medium and small building retrofit market in mind, as it is these markets that represent the vast majority of buildings and floor space.

The results of this forum are presented not to isolate the direct source from which the areas of research in this dissertation were derived, but rather to reinforce the predominant arc towards the future of building energy systems and the ideas behind the motivation of the following efforts. As mentioned in Chapter 1, although buildings are initially setup to run efficiently, their use, equipment, and environment change over time. To keep up with these changes, continuous identification of hardware and operational opportunities are required. While energy audits can be performed to accomplish this identification, they are time consuming and under-utilized. To address these issues, an automated robotic assessment platform is proposed. By its very nature the platform is an advancement in automated building technology and falls right in line with enabling small and medium building retrofits, reducing the cost and making the process easier to perform and provide results to customers. Furthermore, as buildings are consistently changing, there is a need for controls that continually operate the system at energy and economic optimal conditions. Current standard practice with PID control loops is static and does not update over time. Thus, an advanced control strategy that uses continually updating models along with economic objective functions is developed and presented. This strategy not only optimizes the economic cost of the energy consumed, but optimizes the economic implications

of loss of productivity due to occupant discomfort as well. This capability of the building being able to respond adaptively to its occupants is a step along the arc towards future building energy systems. While providing near-term benefits in allowing for the identification of cost saving areas for research and development focus, it is also part of a larger development of an advanced distributed controller that automatically identifies models and system interactions and allows for optimal operation of large-scale systems with reduced information and communication burdens compared to traditional methods. But these two research efforts are not disjoint; rather, as the two technologies become more refined, they can be integrated to provide a more comprehensive and effective solution. The automated assessment platform can be part of the building energy management (BEM) system. Performing monthly, weekly, or even daily audits, the platform can assist in fault detection and diagnosis as well as provide enriched data for the advanced distributed controller, beyond the data from statically located building sensors. As the advanced controller continues its model iterations over time, this enriched data will enable more accurate predictions and thus a more comfortable, environmentally friendly, and cost saving building control strategy, benefiting occupants, building owners, and the world at large. This is the vision that was the driving force behind these two areas of research, and is the vision for the future of the presented contributions.

2.2 Energy Audits and Automation of the Audit Process

As discussed, a major opportunity in improving building energy efficiency is the timely identification of potential equipment retrofits and changes to operational settings. Buildings are generally commissioned when first constructed, meaning the systems and equipment installed in buildings are setup and tuned to operate as intended in the building design. This means maintaining the desired environmental factors of the building in a cost and energy efficient manner. However, as buildings age and these systems aren't updated

with newer technologies or re-tuned to operate with new conditions (both functional and environmental), their efficiency and effectiveness can decrease, resulting in both undesirable conditions and increased cost. One of the first steps in correcting these situations is identification of the issues which can be performed through the use of energy audits. As previously mentioned, energy audits are time intensive and under-utilized. Automation of this auditing process would decrease the number of required personnel as well as the time to complete an audit. Additionally, an automated solution would provide for more dense data sets of environmental measurements and increase repeatability of audit outcomes, reducing the variance that can come from auditor experience.

Thus we proposed a platform termed Autonomous Robotic Assessments of Energy (AuRAE) [5, 8, 3]. Considered as state of the art, these AuRAE will be performed by a cooperative fleet of ground and aerial vehicles, though singular units could be used in smaller facilities. Upon arriving at an audit site, the auditor would turn on the robots which would begin gathering data and measurements as they navigate through and around buildings, freeing the auditor to perform other tasks such as interviewing building operators or discussing audit priorities with the business owner. Ground vehicles would serve as mobile bases for the aerial vehicles, relaying and coordinating information as well as providing additional power in the form of swappable battery packs as needed. Robots could explore cooperatively combining their maps in real-time, further reducing the duration of an audit. The aerial vehicles would be equipped with specific sensor packages enabling certain objects of interest to be identified (lights, HVAC ducts/terminals, windows, etc.) and correlating measurements to be taken.

The collected data would be leveraged for the purposes of energy modeling and making savings recommendations. Some measurements of interest are temperature, relative humidity, human occupancy, CO₂ levels, lighting levels, light locations and types, missing/-faulty insulation (both in walls and around windows), ducting leaks, thermal reflectivity of

the exterior, and overall dimensions of the building (interior and exterior). This information would ideally be processed in real-time and displayed on a tablet or mobile device for the auditor to examine in numerical and map form. The vehicles would have two modes of operation: 1) fully autonomous exploration, and 2) a follow-me mode. The follow-me mode would cause the robot to identify and follow the auditor. This would be beneficial in multi-floor facilities, where the vehicles may need to use elevators to transition between floors. This could also be useful in situations where certain areas may only be audited during a specific time. The robot would follow the auditor into the space, transition to fully autonomous to explore and measure the space, and then transition back to follow-me mode to exit the area, following the auditor. From the collected data, many of the typical recommendations for energy and cost savings can be made. Additionally, 3-dimensional models can be automatically generated in such a format that they can be imported into energy simulation software so that buildings with varying conditions and equipment can be simulated. The combination of these capabilities will result in an in-depth report given to the building owner/operator with recommendations for energy savings.

However, several unique challenges arise due to the automation of these tasks. When navigating in an indoor building environment, robots are not able to utilize current positioning technologies, such as the global positioning system (GPS). While some facilities may have blueprints available, translating these plans into a sufficiently accurate map for the robots navigational use is not trivial; therefore, the robots must fully explore an unknown building environment autonomously, tracking their location as they travel. Another layer of complexity is added in that specific objects and related data must be identified and measured. Thus, the robots must be able to identify these objects of interest, alter their exploration path, navigate to the targets, and make the required measurements and observations. To solve the challenges of navigating indoors, an integrated mapping, localization, exploration, and target identification solution is needed. The next sections present

a proposed solution to the aforementioned challenges.

2.2.1 Navigational Strategies and Algorithms

Solutions are needed for both the indoor and outdoor navigation problems that come with operating in and around buildings. Mobile robotic mapping has been an area of active research for many years. To successfully create a map of an area, a robot must be able to sense its environment and determine its location with respect to its surroundings. This problem is known as the Simultaneous Location and Mapping (SLAM) problem, which has been a large area of research in autonomous robotics. An example of what SLAM provides is shown in Figure 2.1. Obstacles and boundaries are shown in black, observed space is shown in light grey, and the robot's path is shown in red. Much research has been done on autonomous UVs in an outdoor environment which include GPS reliance to estimate the robots position and odometry [57, 58]; however, an increase in focus on SLAM systems that can perform indoors in GPS-denied environments has occurred over the last decade [59, 60, 61]. Different sensors have been utilized in these studies, such as 2D LASER range finders [62, 63, 64], monocular vision cameras [65, 66, 67, 68], stereo camera systems [69], and more recently, 3D depth cameras [70, 71, 72]. While SLAM solutions exist for operating indoors, robotic systems still require navigational strategies to determine where to move to and how to get there.

Navigational strategies and path-planning for autonomous robots have been researched for quite some time. Many algorithms are available with a recent focus on using rapidly-exploring random trees (RRTs), originally proposed by LaValle in [73]. In LaValle's paper, the RRT algorithm is described as starting at an initial point and growing "branches" in random directions whose selection is biased by their Voronoi diagrams. Essentially, branches are biased to grow towards regions that do not currently contain nodes. With enough iterations, the branches fill the space and there will exist a path from the starting

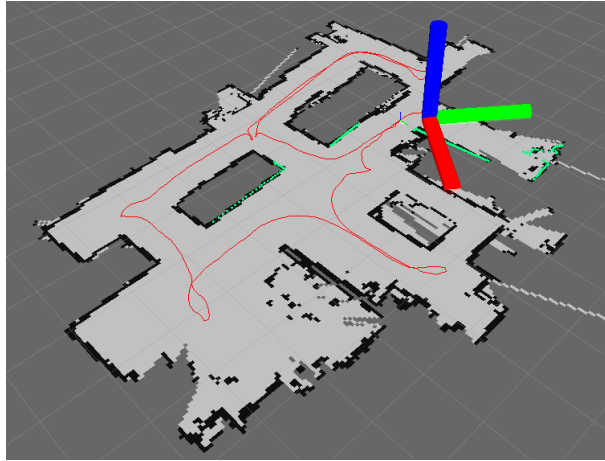


Figure 2.1: Map created by ground robot platform during interior navigation.

point to any space (within the resolution of the tree). Later, LaValle built upon this work to propose RRT-Connect [74] which incrementally builds two RRTs. One originates from the starting point and the other from the goal. They expand until they connect with one another. Both of these strategies work in a known environment with a known goal, but do not provide a solution for a robot continually exploring an unknown environment. However, the random nature of the RRT algorithm provides for probabilistic complete exploration of an enclosed space, which is useful for interior exploration.

A data structure based on RRT was proposed by Oriolo in [75], called Sensor-based Random Tree (SRT). As the name states, this method adapts the RRT strategy to the characteristics of typical hardware used in robotic sensing. The sensor perceives the surrounding and an associated safe region is determined. The tree is then incrementally built based on the generation of random points within this safe region. The resulting data structure represents a global map of safe regions which expands as the robot navigates. In [75], two exploration techniques are detailed: SRT-Ball and SRT-Star. Both methods assume 360° of perception, requiring a complete rotation sensor or integrating multiple devices

that overlap to provide full coverage. SRT-Ball's radius of perception is limited by the nearest sensed obstacle. SRT-Star's area of perception is divided up into conical areas with each cone extending as far as the nearest obstacle in the cones region. SRT-Star is more confident as it uses more of the information available while SRT-Ball provides more conservative navigation. With SRT-Star, additional computation compared to SRT-Ball is required in determining each cone from one another as perception and navigation takes place. A method that builds on SRT-Ball and SRT-Star, SRT-Radial, was proposed by Espinoza in [76]. In the improved SRT-Radial method, the perceived safe region contours to obstacles, making complete use of the information available. In [76], 360° of perception is assumed, and with all the SRT strategies, the authors assume that perfect localization is provided by a SLAM method. Achieving 360° of perception can be expensive and difficult. Also, these strategies do not allow for identifying and navigating to targets as they are discovered in real-time.

2.2.2 Interior Navigational Algorithm and Simulation

To address the aforementioned issues, the interior building navigation algorithm termed SRT-Target was proposed. SRT-Target is an algorithm capable of completely exploring a previously unknown, enclosed space while navigating to objects that are identified in real-time by a sensor package. Pseudo-code for the algorithm is shown on the left-hand side of Figure 2.2. Complete details of the algorithm and how it works can be found in Chapter 3. A brief summary of the algorithm follows. The robot first initializes its current position. Then an initial perception of the surrounding environment is taken and stored in variable S . Next, S is added to the global map T and $S \cdot \beta$ is added to the map U , a reduced global map. $S \cdot \beta$ is a reduced version of S that facilitates the robot physically visiting each space to ensure all objects of interest are identified. This is necessary because of the hardware limitations of the sensing package that require the robot to be within a certain distance of

any target. A loop is then entered to find a valid candidate position to move to. The robot first checks if there is an object of interest within range. If so, the object's location becomes the candidate location, and the robot proceeds to those coordinates and the loop is exited. If there is no target, a random direction is chosen. The distance along this direction to the nearest obstacle is determined, within the limits of the range sensor. Worth noting is that the obstacle boundaries are expanded in such a way to ensure the robot will never come close enough to contact them. A candidate position that is a variable distance (dependent on α) along that direction is generated and then checked to be valid, ensuring it exceeds a certain distance threshold and is within a region that hasn't previously been sensed twice. The robot moves to the location if it is valid; otherwise, the robot continues the search for a valid location candidate, up to a fixed number of iterations. If no valid candidate location can be found, then the robot returns to the last, or parent location and starts the process over. In this fashion, the robot completely explores an enclosed space, ensuring discovery of all objects of interest. Upon finding no new valid candidate locations, the robot returns to its starting point.

Simulations of the interior navigational algorithm SRT-Target have been performed. Results from some of the simulations are shown in Figure 2.3. The UAV is the star shape in blue, its trajectory the thin green line, objects of interest (in this case, lights) and their associated radius of discovery are represented by the rounded rectangles in red, obstacles are shown in black, open space is shown in white, and unknown territory is shown in grey. The layout shown is representative of an actual office space, hallway, and laboratory. As the iterations increase, beginning in the top left and finishing in the bottom right, the random navigational direction of the UAV can be seen. Upon entering the radius of discovery for a light, the UAV navigates to the center of the light to facilitate data measurement (spectrometer and light level). This behavior can be repeated for any object of interest, provided there is a sensor package to find the object. Additional visual results of interior

```

SRT_TARGET( $q_{init}, K_{max}, I_{max}, \alpha, d_{min}$ )
1:  $q_{curr} = q_{init}$ ; # Initialize current position.
2: for  $k = 1$  do  $K_{max}$ 
3:    $S \leftarrow \text{PERCEPTION}(q_{curr})$ ; # The perceived environment is stored in S.
4:    $\text{ADD}(T, U, (q_{curr}, S, \beta))$ ; # Add the current location and S to the global
5:   # tree T and add  $S \cdot \beta$  to the tree U.
6:    $i \leftarrow 0$ ; # Initialize the count.
7:   loop # Loop until a valid candidate is found
8:   # or max iterations are reached.
9:      $(target, q_{target}) \leftarrow \text{TARGET\_CHECK}$ ; # Check for target at current location.

10:    if ( $target = true$ ) then
11:       $q_{cand} = q_{target}$ ; # Make the candidate location the target location.
12:      break; # Stop searching for candidate.
13:    else
14:       $\theta_{rand} \leftarrow \text{RANDOM\_DIR}$ ; # Pick random direction within sense range.
15:       $r \leftarrow \text{RAY}(S, \theta_{rand})$ ; # Find greatest distance  $r$  along
16:      #  $\theta_{rand}$  in the local safe region.
17:       $q_{cand} \leftarrow \text{DISPLACE}(q_{curr}, \theta_{rand}, \alpha \cdot r)$ ; # Calculate candidate location
from
18:      # current location,  $r$ , and  $\theta_{rand}$ .
19:       $i \leftarrow i + 1$ ;
20:    until ( $\text{VALID}(q_{cand}, d_{min}, U)$  or  $i = I_{max}$ )
21:    if ( $\text{VALID}(q_{cand}, d_{min}, U)$ ) then
22:       $\text{MOVE\_TO}(q_{cand})$ ; # Move to candidate.
23:       $q_{curr} \leftarrow q_{cand}$ ; # Update current location.
24:    else # If valid candidate can't be found move to parent location.
25:       $\text{MOVE\_TO}(q_{curr}.parent)$ ;
26:       $q_{curr} \leftarrow q_{curr}.parent$ ; # Update current location.
27: return  $T$ ; # The result is given in the global map T.

```

Figure 2.2: The SRT-Target algorithm pseudo-code.

simulations can be found in Chapter 3.

A series of interior simulations were conducted, consisting of five similarly sized layouts of different physical configurations and light locations. For each room, five simulations were performed with the robot starting from a different point each time. The results

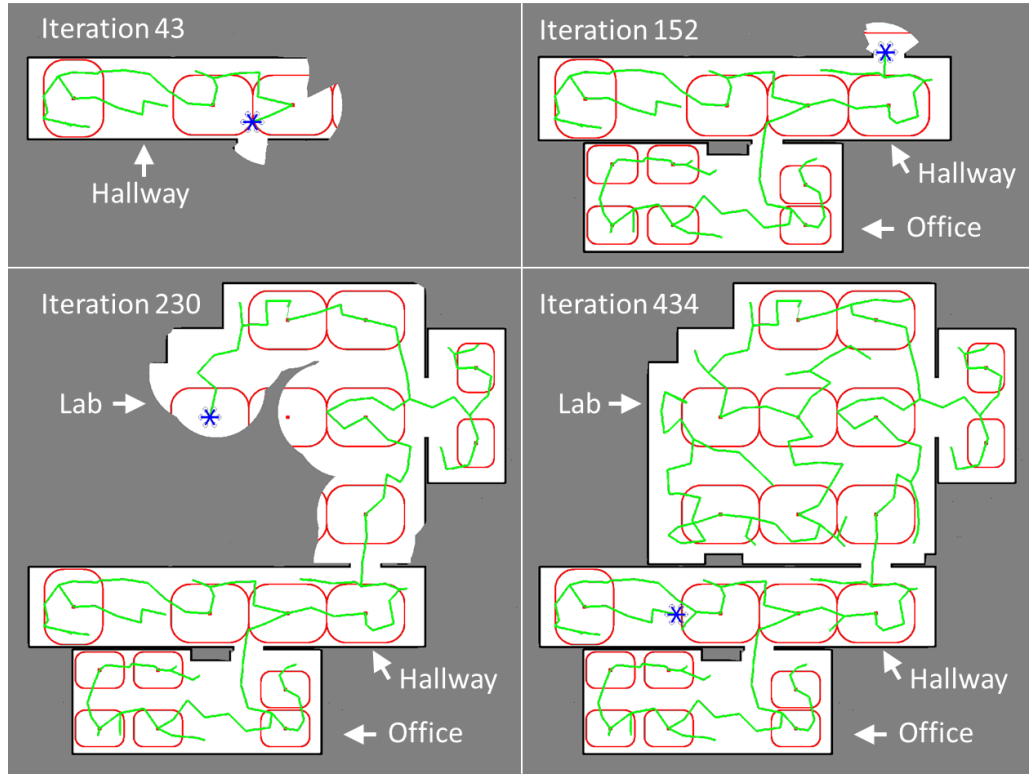


Figure 2.3: Results from a simulation of the SRT-Target algorithm. The UAV is shown as the star shape in blue, its path as the thin green line, objects of interest as red dots with their associated radius of discovery as rounded red rectangles, obstacles as black, open space as white, and unknown territory as grey [3].

of these simulations are summarized in Table 2.1. In each case of the 25 different simulations the robot fully explored the layout and found 100% of the lights.

Table 2.1: Summary of simulation results for five different layouts.

	Layout 1	Layout 2	Layout 3	Layout 4	Layout 5
Average Iterations	112.6	105.8	127.8	97.8	149.4
Avg. Distance Traveled (ft.)	130.3	125.0	152.5	111.6	175.2
Avg. Exploration Completion (%)	100.0	100.0	100.0	100.0	100.0
Correct Light Detection (%)	100.0	100.0	100.0	100.0	100.0

2.2.3 Exterior Navigational Algorithm and Simulation

The SRT-Target method works for interior navigation of unknown environments that have a closed perimeter; however, for performing various audit tasks on the exterior of a building with potential openings, a different approach is required. Numerous boundary following procedures and techniques have been proposed over the years. Some have been based on fuzzy logic controllers [77], Lyapunov functions [78], or even the sensing mechanism of cockroaches [79]. Many use single or multiple ultrasonic sensors [80, 81, 82] and are meant to operate in known environments. These methods were designed to perform wall-following, but lack the capability to circumnavigate perimeters that are not fully closed. One boundary following method [6] uses instant goals to provide path planning. Instant goals allows for a robot to follow arbitrarily shaped shapes of simple or complex geometries. Additionally, the robot can encounter disturbing obstacles and continue moving along the previous boundary, subject to the robots geometry being able to pass between the two obstacles. An ideal strategy would be able to follow arbitrarily shaped buildings, with both convex and concave corners, and be able to traverse open bay doors, common with commercial and industrial buildings. This would simplify the audit process by not requiring the customer to close all exterior openings that may be open during normal use.

For these reasons an algorithm based on the instant goals method presented in [6] was developed for external navigation. This proposed algorithm can follow convex and concave perimeters while being able to traverse potential openings in the boundary (open doors, passageways, etc.), which is not currently achievable with reported algorithms. This is ideal for navigating the exterior profile of various industrial and commercial buildings; however, the proposed strategy can also be used in other areas where non-closed boundary following is required. The exterior perimeter can then be used a boundary for the interior navigation algorithm, allowing for interior exploration to occur without requiring the

closure of building doors and openings which may interrupt business productivity.

Pseudo-code of the exterior algorithm is shown in Figure 2.4. Details of the algorithm can be found in Chapter 4, but a brief summary is given here. The robot begins with an initialization of its current position and initial perception of the surrounding environment, dividing up the area of perception into slices, as shown in Figure 2.5(a). Finding the nearest obstacle, the robot moves to a user-defined, perpendicular distance away. The robot then perceives the space in front of and alongside it, determining if it has reached a corner. If the robot has reached what it determines to be a corner, it will extend its perception range out to a user-defined value dependent on the largest exterior gap that should be traversed, shown in Figure 2.5(b). If no additional obstacle is found, the robot will move around the corner in an arc as shown in Figure 2.5(c). If an additional obstacle is found, the robot determines that there is a traversable gap and will navigate to the other side, as shown in Figure 2.5(d). This process of sensing the wall and determining the distance to travel along the perimeter is repeated until the robot reaches its starting location.

Figure 2.6 shows results from the simulation of the exterior navigational algorithm. Again, the UAV is shown as a blue star shape, its path as a thin green line, obstacles as black, open space as white, and unknown territory as grey. The simulations show the UAV successfully navigate convex and concave features as well as open gaps in the perimeter. Additional visual results can be found in Chapter 3.

```

EXT_NAV( $q_{init}$ ,  $srch_{from}$ ,  $srch_{to}$ ,  $srch_{dir}$ ,  $n_{slices}$ ,  $d_{gap}$ )
1:  $q_{curr} = q_{init}$ ; # Initialize current position.
2:  $S \leftarrow \text{PERCEPTION}(q_{curr})$ ; # The perceived environment is stored in S.
3:  $q_{wall} = \text{NEAR\_OBS}(q_{curr}, S)$ ; # Find nearest point on obstacle.
4: if ( $\text{DIST}(q_{curr}, q_{wall}) \neq \perp_{dist}$ ) then
5:    $q_{\perp} = \text{PERP}(q_{curr}, q_{wall}, \perp_{dist})$ ; # If the robot is not the user-defined distance
6:    $\text{MOVE\_TO}(q_{\perp})$ ;  $q_{curr} \leftarrow q_{\perp}$ ; # away from the wall, move to the correct
7:    $S \leftarrow \text{PERCEPTION}(q_{curr})$ ; # position and update  $S$  and  $q_{wall}$ .
8:    $q_{wall} = \text{NEAR\_OBS}(q_{curr}, S)$ ;
9: if ( $\text{CORNER}(S, q_{wall}) = \text{true}$ ) then # If at a "corner", extend the perception
10:   $S_{extended} \leftarrow \text{PERCEPTION}(q_{curr}, srch_{dir}, d_{gap})$  # to determine if there is a gap.
11:  if ( $S_{extended} \neq \text{empty}$ ) then # Divide  $S_{extended}$  into sections.
12:     $D \leftarrow \text{SLICES}(S_{extended}, n_{slices}, srch_{from}, srch_{to}, srch_{dir})$ ;
13:     $q_{dist} = \text{DIST}_{last}(D, q_{wall})$  # Find distance to last obstacle sensed.
14:  else
15:     $D \leftarrow \text{SLICES}(S, n_{slices}, srch_{from}, srch_{to}, srch_{dir})$ ; # Divide  $S$  into sections.
16:     $q_{dist} = \text{MAX}(\text{DIST}(D, q_{wall}))$ ; # Find the max. distance between the nearest
    # wall point and the obstacles sensed in  $D$ .
17: else
18:   $D \leftarrow \text{SLICES}(S, n_{slices}, srch_{from}, srch_{to}, srch_{dir})$ ; # Divide  $S$  into sections.
19:   $q_{dist} = \text{MAX}(\text{DIST}(D, q_{wall}))$ ; # Find the max. distance between the nearest
    # wall point and the obstacles sensed in  $D$ .
20:  $[targ_{dist}, targ_{head}] = \text{LINE}(q_{dist}, q_{wall})$ ; # Draw a line between the points
21:  $q_{targ} = \text{TARGET}(q_{curr}, targ_{dist}, targ_{head})$ ; # and find the target coordinate.
22: if ( $targ_{dist} = 0$ ) then
23:   $\text{MOVE}(q_{curr}, targ_{dist-prev}, targ_{head-prev})$ ; # If only one obstacle is sensed
24: else # use previous motion.
25:  if ( $\text{DIST}(q_{targ}, q_{wall}) < \perp_{dist}$ ) then # If  $q_{targ}$  is too close to the obstacle, revise
26:     $[targ_{dist}, targ_{head}] = \text{LINE}(q_{targ}, q_{wall})$ ; #  $q_{targ}$  to be farther from obstacle.
27:     $q_{targ} = \text{TARGET}(q_{wall}, \perp_{dist}, targ_{head})$ ;
28:     $\text{MOVE\_TO}(q_{targ})$ ;
29:  else
30:     $\text{MOVE}(q_{curr}, targ_{dist}, targ_{head})$ ; # Move to target location.
31:  $[targ_{dist-prev}, targ_{head-prev}] = [targ_{dist}, targ_{head}]$ ;
32: while  $q_{curr} \neq \text{THRESH}(q_{init})$  do Repeat lines 2-40;

```

Figure 2.4: The external navigation algorithm pseudo-code.

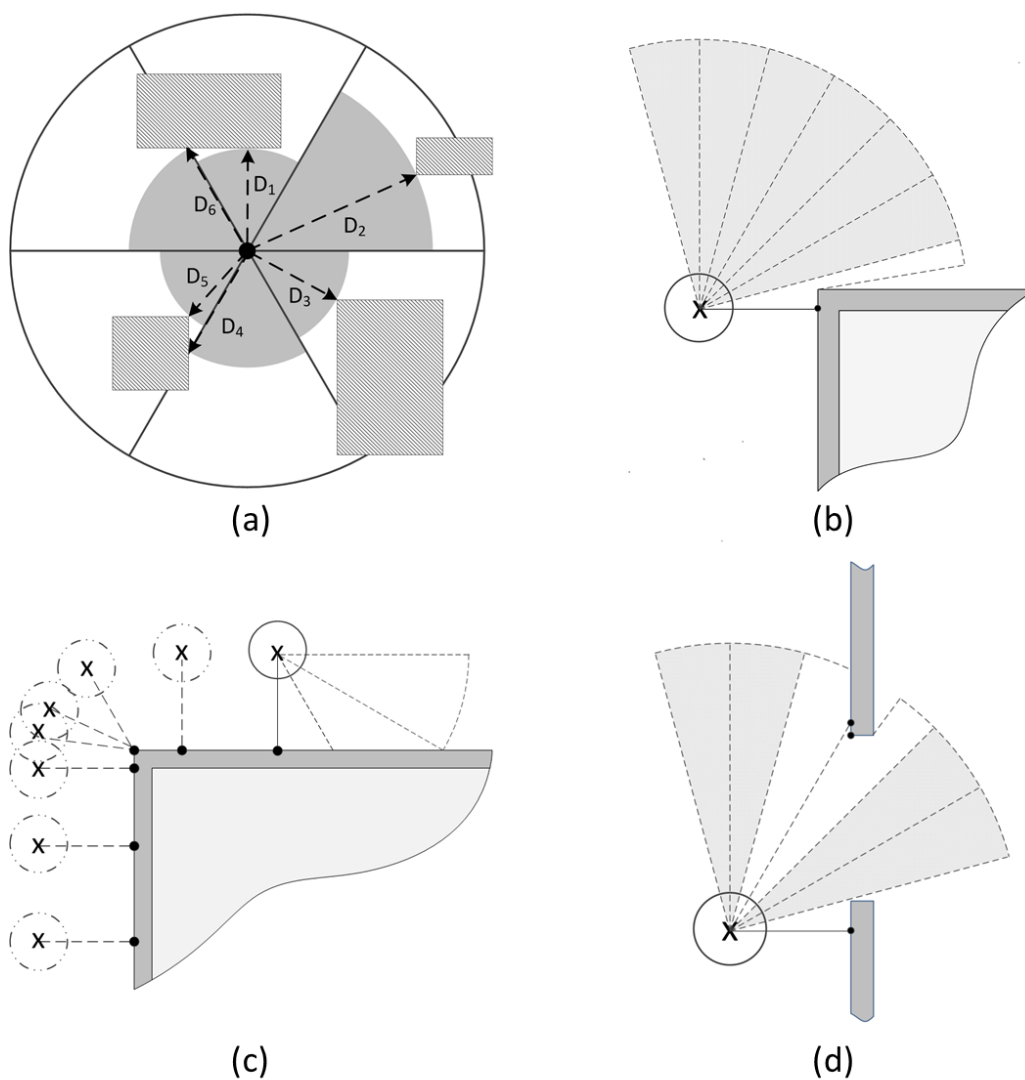


Figure 2.5: (a) The environment is divided into slices to decrease the amount of stored data. (b) An example of the robot's perception upon reaching an exterior corner. (c) A demonstration of how the robot navigates exterior corners. (d) An example of the robot's perception at a traversable gap.

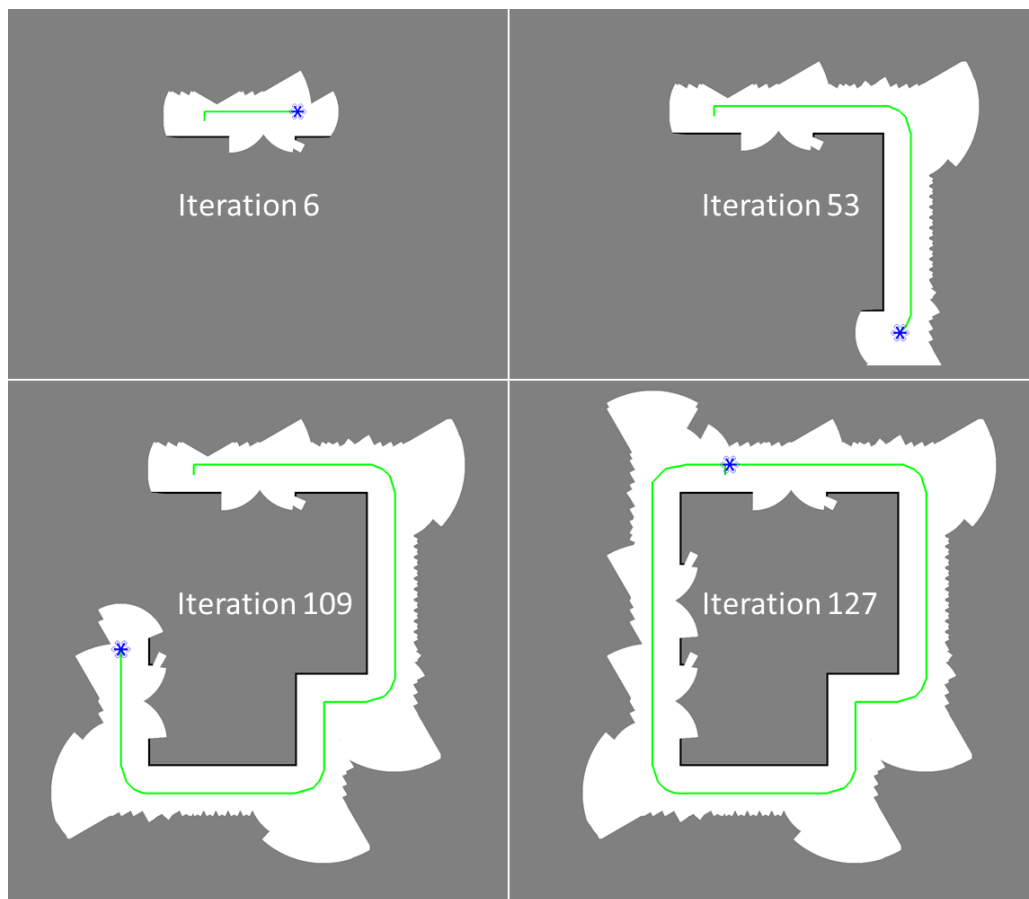


Figure 2.6: Results from a simulation of the exterior navigational algorithm [3].

2.2.4 Development of Lighting Package

Lighting is a key consumer of electricity in buildings. In the U.S., approximately 12%, or 765 TWh, of the total electricity consumption is due to lighting [83]. Other estimates of the global electricity usage put lighting at 20% [84]. Having such widespread use, there is constant development of new and more efficient lighting technologies. Upgrades and changes to building lighting are also easier to implement than several other building upgrades, often requiring less time and money to implement. Consequently, lighting is one of the first assessment tasks that auditors consider. Examining the Industrial Assessment Center database [85], which contains over 16,000 assessments across 40 years, shows that of the over 120,000 recommendations made, more than 20% pertain to lighting. As a key consumer in building electricity and a frequent recommendation given by auditors, lighting assessments were a logical technology to develop first for AuRAE.

To correctly identify and analyze lights, several sensors are needed. The lighting sensing package for AuRAE contains a USB camera, a light level sensor, a USB laser rangefinder, and a spectrometer. The USB camera captures images of the ceiling. These images are used to identify the presence of lights, their shape, and their location through some morphological processes and coordinate transformations, detailed in 5. The light level sensor records lighting information to be used later in analysis and simulation. The USB laser rangefinder is key in determining the locations of the lights relative to the robot. The spectrometer is used to analyze the lighting spectrum to determine the type of bulb in use. After all the data are collected, lighting maps and simulations can be generated.

A novel lighting package prototype has been developed and installed on a ground robot platform, as shown in Figure 2.7. The ground robot uses Hector SLAM and a 2D scanning laser rangefinder to generate a map and locate the robot's position. The lighting package identifies ceiling lights in real-time as the robot navigates a space. Currently the ground

robot is controlled by a user via a base station computer or mobile tablet. In place of the navigational strategy informing the lighting package to take a measurement upon arriving at the light location, the user currently sends a command to record lighting data.

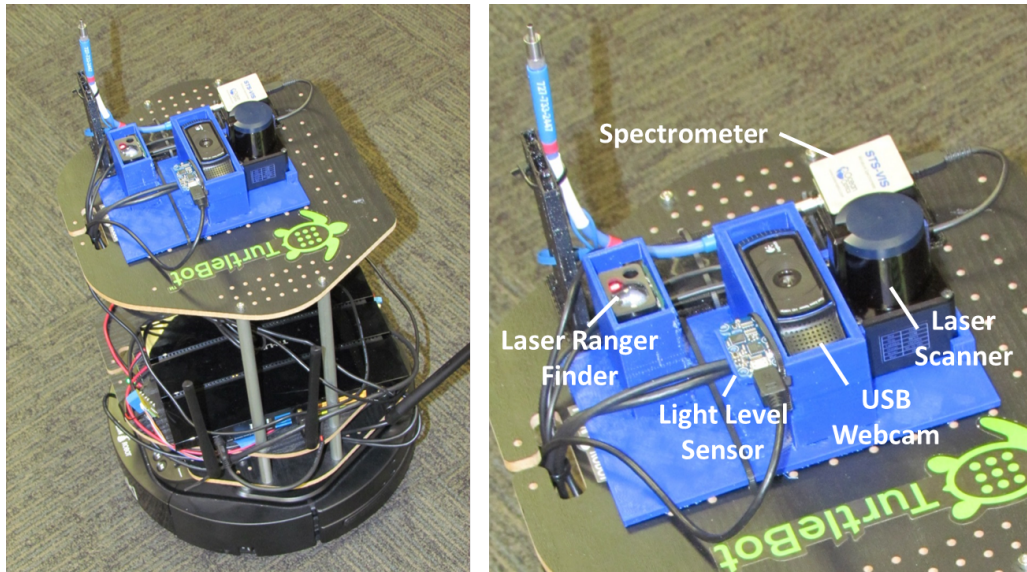


Figure 2.7: Ground robot platform prototype with lighting identification package installed.

Preliminary experiments with the ground robot platform have been performed, with novel results shown in Figure 2.8. The figure shows real light detection and facility map generation using the methods described previously. For reference, the black shown in the images represents obstacles, the light gray represents open space, the medium grey represents unknown space, and the dark grey represents space outside of the mapping domain. The yellow spheres represent the centers of identified lights and the faint, green line shows the ground robots path. Using a 2D scanning laser rangefinder, Hector SLAM produces a 2D map of the building. The lighting package is able to determine the height of the lights with its laser rangefinder. Overall, the results are accurate and very promising. The preliminary tests were performed in Evans library on Texas A&M's campus. The rows

of obstacles observed in Figure 2.8 are the book shelves. In the lower part of the image, the lights can be seen to be in two straight rows along the wider path way between two sections of the library. Some variance in the lighting locations can be seen above the book shelves. Because the shelf height is so close to the ceiling that parts of the lights were actually obstructed from the view of the ground robot, causing the robot to only identify the center of the visible light. The size of the library and the number of lights show that the developed methods are able to handle larger commercial and industrial spaces.

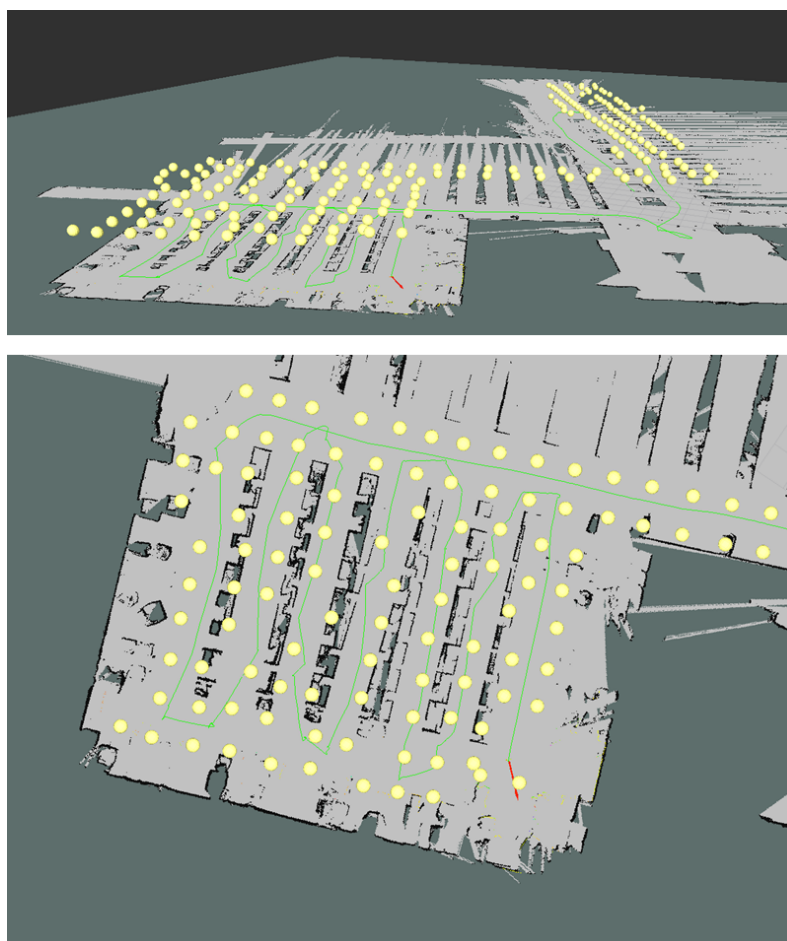


Figure 2.8: Preliminary results from the ground robot platform and lighting package prototype. The yellow spheres represent the centers of lights identified in the building (Evans Library at Texas A&M).

2.2.5 Development of Environment Sensing Package

Establishing an environment baseline of the site being assessed is useful in ensuring that comfort and health guidelines and regulations are being met. The collected data can also be used in the fault detection of existing building sensors. Duct balancing problems can be found and addressed, as well as stratification, poor ventilation, and other location dependent problems. An environment baseline consists of measurements of temperature, relative humidity, CO₂ levels, and lighting levels throughout the building. Additionally, the data gathered provide key information for performing the more advanced energy simulations and analysis that come with more comprehensive audits. By freeing the technician from having to collect these measurements, their time can be spent elsewhere interviewing staff or assessing other equipment.

To gather the required data, an environment baseline package prototype has been developed. It consists of a microcontroller, combined temperature and relative humidity sensor, a CO₂ sensor, and a light level sensor. Data is written to a SD memory card for easy access during the analysis phase. In order to gain the best representation of the building environment while working to automate the process, a UAV platform was chosen for the sensing prototype. The UAV allows the package to take measurements at various heights and locations throughout the building. A Google Project Tango tablet was used to determine the UAV's 3D position. An app running Real-Time Appearance-Based Mapping, or RTAB-Map, is used [86] to solve the SLAM problem. RTAB-Map is a RGB-D Graph SLAM approach that uses a Bayesian loop closure detector. More details can be found in [86]. The UAV platform with the sensing package prototype can be seen in Figure 2.9.

Preliminary results from taking measurements from a local church are shown in Figure 2.10. The top portion of Figure 2.10 shows an oblique view of the areas of the church that were mapped. The UAV was maintained at roughly the same height through the

hallways and classrooms; thus, there was little temperature variation in the vertical direction. However, in the gym, the UAV performed multiple passes at different heights. Even with the approximately 20 foot ceiling, the temperature and humidity showed very little variation. The bottom portion of Figure 2.10 shows an overlay of interpolated CO₂ concentrations throughout the building. Not all areas of the building were visited, as is visible by the trajectory of the UAV shown in red. The highest levels appear in the gym, near the divider that was partially open. This could have resulted from the researchers' extended time in this area during data collection. To verify the 3D position determination and mapping algorithms, and the multi-sensor system, the UAV was piloted manually in this example.

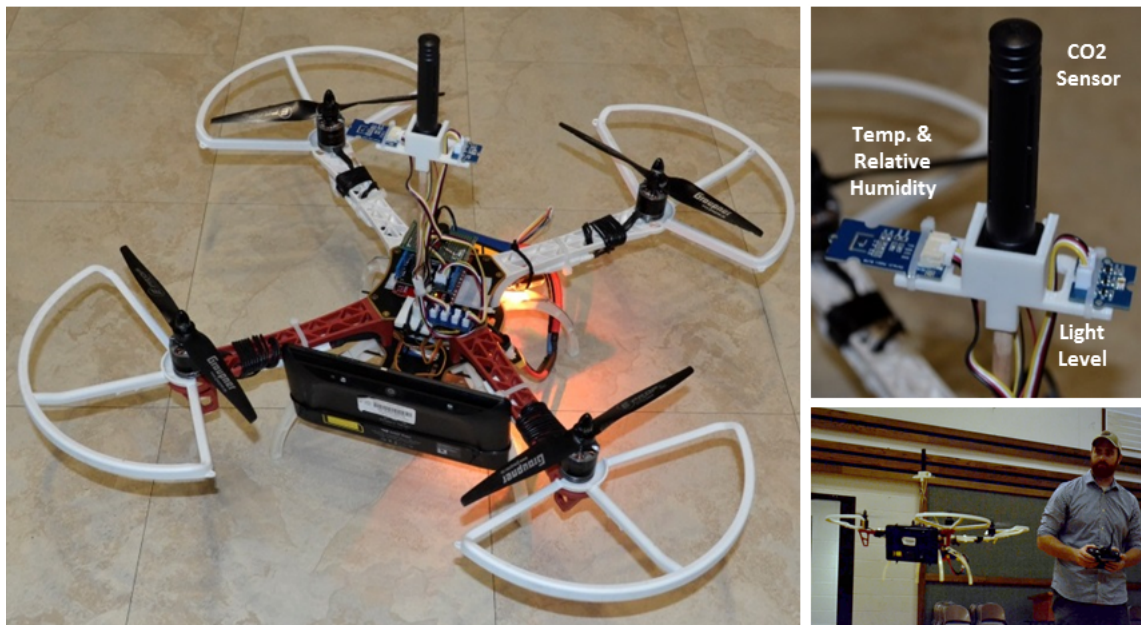


Figure 2.9: The UAV platform and sensing package prototype used to collect baseline environment measurements.

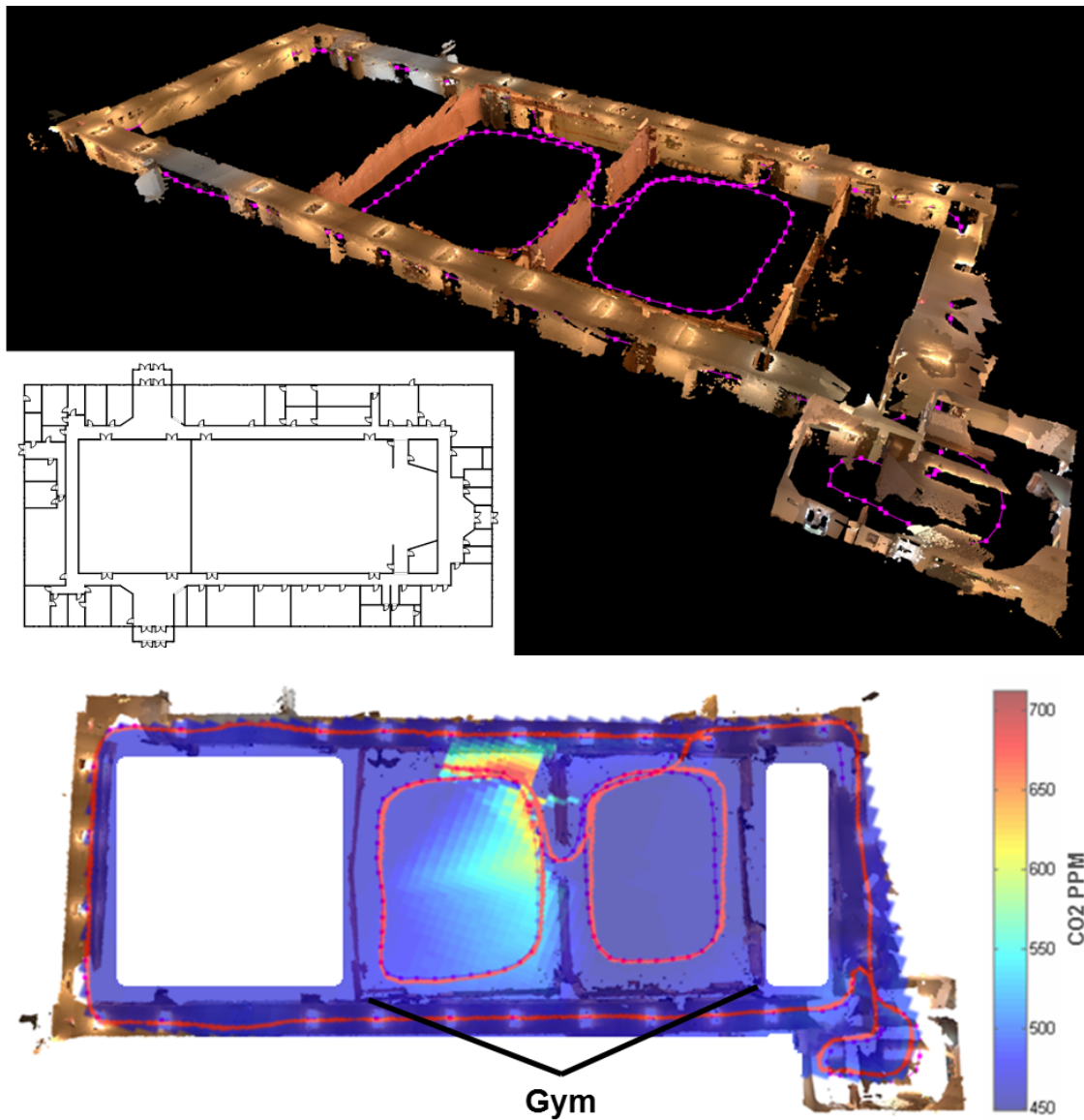


Figure 2.10: Preliminary mapping and CO₂ measurement results from the environment sensing package and UAV platform.

2.3 Economic Control and Energy Optimization in Building Energy Systems

Control of energy systems in buildings presents several unique challenges. Buildings consist of numerous interconnected energy systems that affect and depend on one another. Figure 2.11 shows a typical HVAC system setup for a commercial building. In this system there is a chiller that chills a secondary fluid, such as water. This chilled water can then be pumped to various systems and areas of buildings where heat exchangers in air handling units (AHUs) use the chilled water to cool air streams. In this case, it is pumped to a singular AHU. A network of fans and ducts then deliver the cooled air to the desired locations. The flow of this cooled air into the zones can be controlled by variable air volume (VAV) units, in which there may exist an additional heat exchanger that utilizes heated water to reheat the air, if necessary. In the displayed system, the heat exchanger responsible for reheating the air is located in the main AHU. The heated water for this process is provided by a different set of centralized pumps and heat exchangers, shown as the boiler. The zones themselves are connected to one another, either by conduction through walls or by shared air through doorways/open spaces. While the example in Figure 2.11 shows a typical system for one building, a campus can have tens to hundreds of these systems, each with unique effects and connections with other systems. All these interconnections and couplings can make for highly complex coordinated control problems.

Providing additional difficulties, nonlinear dynamics and multiple time scales occur across these various building systems. Equipment such as chilled water valves, fans, and dampers can have nonlinear behavior. For example, a nonlinear relationship can exist between air flow into a zone and damper position, dependent upon operating conditions and the damper used (single blade, opposed blade, etc.). End static pressure, often regulated by fan speed, depends on duct efficiencies and losses as well as zone damper positions. As for time scales, changes in damper position in a VAV or fan speed in an AHU are relatively fast

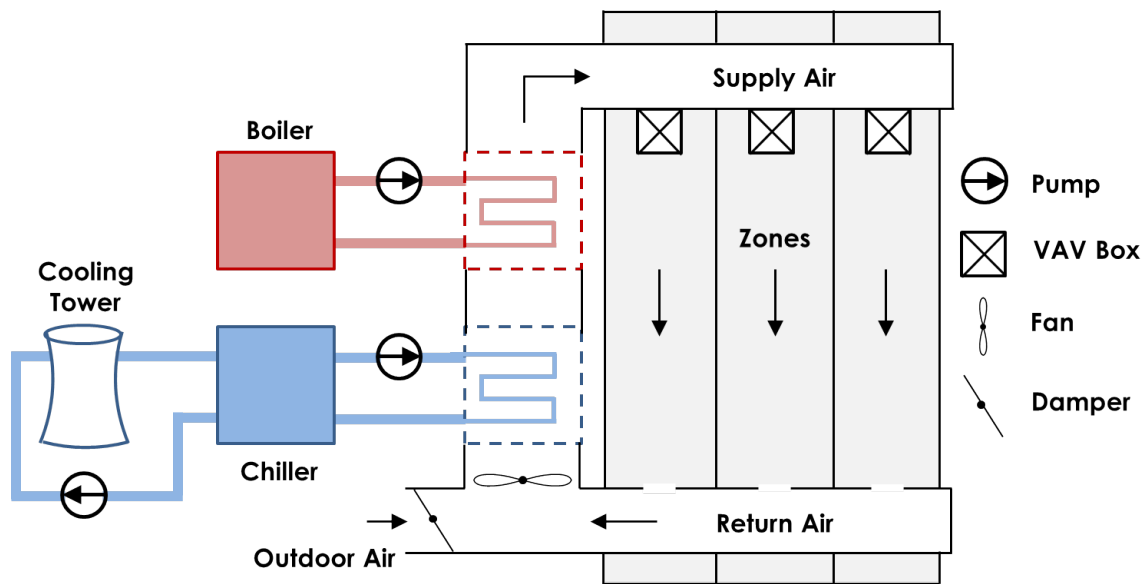


Figure 2.11: Typical building HVAC system in commercial buildings.

(on the order of seconds) compared to changes in desired chilled water temperatures (minutes) or changes in zone air temperature (hours). There are also slow, overarching changes to take into consideration such as the shift in solar loads as the sun moves throughout the day, gradual changes in outdoor air temperature due to change in weather or seasons, and the deterioration of equipment through use over time. These all contribute to shifting systems behaviors and disturbances that can cause undesired performance in systems. There are also discrete changes to take into account, such as whether an area/room is occupied, how many people are in said room, or changes in real-time pricing of utilities. In addition, the sensors of the system are distributed (not always equally), centralized monitoring is being performed, and devices are driven by localized controls. In addition to all of this, the systems must operate within constraints due to hardware limitations, limited resources, and/or issues of health and comfort.

While the challenges of building control are numerous, one control method that has emerged as a capable solution is model predictive control (MPC). MPC has been chosen as

the most appropriate control method in various building thermal control research projects such as Opti-Control in Switzerland [34], Intelligent Buildings and Rational Management of Renewable Energy (MIGRER) in France [35], and MPC for UC Merced Campus in the U.S. [36]. MPC, or receding horizon control, predicts the change in the dependent variables of a modeled system by changing independent variables. Using the current state information, dynamic models of the system, and an objective function, MPC will determine the changes in the independent variables that will minimize the user-defined objective function while honoring given constraints on both dependent and independent variables. Once this series of changes is determined, the controller will apply the first determined control action, and then repeat the calculations for the next time step. Figure 2.12 displays how a typical reference tracking MPC implementation would behave. It can be seen at time k that the controller determines what the predicted output would be along with its optimal control trajectory. After completing the computations, the control would be applied and the system would move on to time $k + 1$, repeating the predictions and optimization with the new measurements. It is important to note the two horizons within MPC: the prediction horizon, which is the length of time for which the system outputs are predicted, and the control horizon, which is the number of control inputs that are determined in the prediction computation. The prediction horizon is often limited by computational capabilities, while the control horizon is selected so that the system dynamics are allowed to diminish [37]. An extensive overview of the recent advances of MPC can be found in [38].

To address the aforementioned challenges of controlling building systems, a large-scale project to develop a MPC control algorithm has been pursued. This project includes investigating cascaded control loops to recover linear behavior of nonlinear systems, developing an automated black-box modeling algorithm that identifies and recursively updates system models from building data, and expanding a steady-state prediction, distributed control method to utilize dynamic models and MPC horizon predictions. However, in

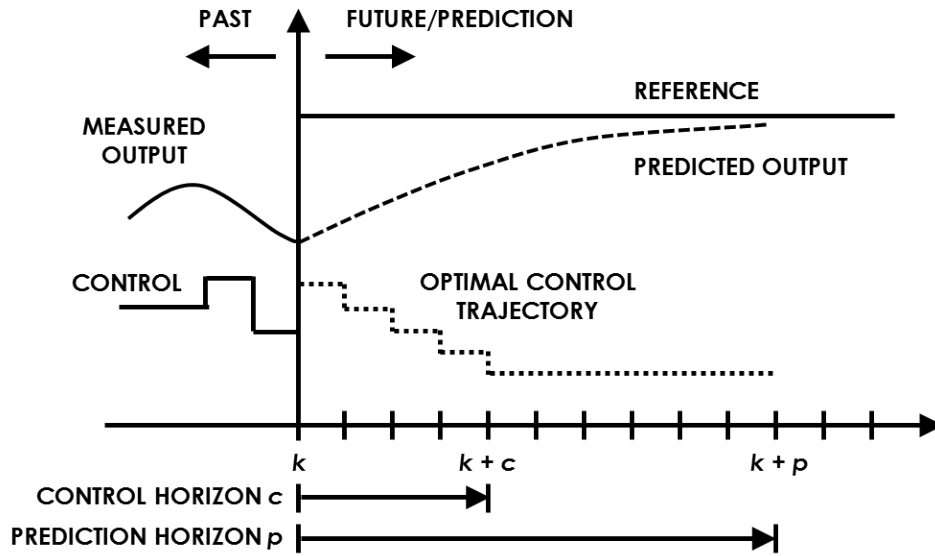


Figure 2.12: How MPC, or receding horizon control, typically functions.

MPC a major portion of the controller design is in determining the structure of the cost function. In this research, the focus was to develop an economic objective function that minimizes the monetary cost of energy usage by the individual components as well as the monetary cost of loss productivity due to discomfort. This choice was made for two reasons: 1) to put occupant comfort on equal footing, in economic terms, with energy usage as opposed to arbitrary weights or temperature limits that are normally used, and 2) to identify the relative monetary cost between different components to inform building engineers and managers about areas of economic importance.

2.3.1 The Utilities Business Office Building

Working with the Utilities Energy Management staff at Texas A&M University, our lab has been granted limited access to the Utilities Business Office (UBO) on campus to gather data from and eventually implement our developed control algorithm on. A specific implementation of the general economic objective function developed for the UBO is de-

scribed in the following subsections, but first, some relevant information about the UBO will be given.

The UBO is a rectangular, single-story building consisting of 11 zones, 10 of which are actively controlled. The general layout can be seen in Figure 2.13. In this initial development, the focus was limited to the cooling aspects of the system because: 1) due to the location and climate of the building, the majority of the year is spent in cooling mode, and 2) simplified operating conditions will help to validate the developed control strategy in its first implementation.

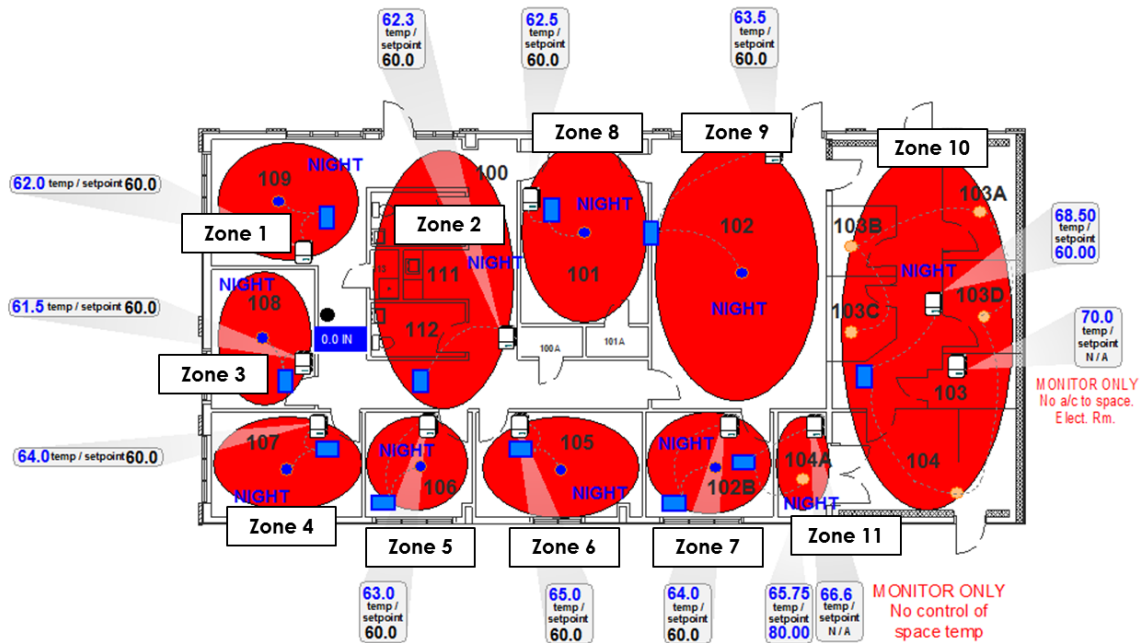


Figure 2.13: Zone layout for the Utilities Business Office (UBO) at Texas A&M University (figure from Apogee Software).

The UBO is serviced by a single, rooftop air handling unit (AHU). The AHU consists of a variable air volume (VAV) fan, a chilled water coil, an outdoor air damper, a return air damper, a discharge air temperature sensor, and an end static pressure sensor. The

organization of these components can be seen in Figure 2.14. During normal operation, the VAV fan works to maintain an end static pressure given by a pressure demand calculation dependent upon the individual damper positions of the zone terminal boxes, moving the air through the system. The pressure demand calculation is the feedback signal for a proportional-integral-derivative (PID) loop that produces a reference equal to the desired end static pressure. This reference is fed to another PID loop that actuates the fan speed to maintain the given end static pressure. For conditioning the air, chilled water provided by central plant is passed through the AHU's chilled water coil to lower the temperature of the moving air as well as help reduce the air's humidity. The amount of chilled water passing through the coil is controlled by a valve which is actuated by a PID control loop. This control loop is driven by a difference in cooling demand setpoint and a cooling demand calculation that is a weighted combination of the average and maximum cooling loopout values from the individual zone control loops. This loop's output is the input (the desired discharge air temperature) for a second loop that actuates the chilled water valve.

Figure 2.15 details the chilled water loop. A chilled water pump works to maintain a specific supply pressure to provide the required chilled water flow for the AHU. Worth noting is that the chilled water supply and return temperatures are available to measure within the energy management system.

The UBO consists of 11 zones, 10 of which are actively controlled. Each controllable zone is serviced by a VAV terminal box equipped with hot water reheat capabilities, an example of which is shown in Figure 2.16. The flow of conditioned air into the room is regulated by a damper in the terminal box whose position is determined by a PID control loop. The error signal for the control loop is the difference between the respective room temperature setpoint and the measured room temperature. The room temperature setpoint is determined by whether the room is occupied or unoccupied as well as whether the zone is in heating mode or cooling mode. A deadband control method is employed such that if

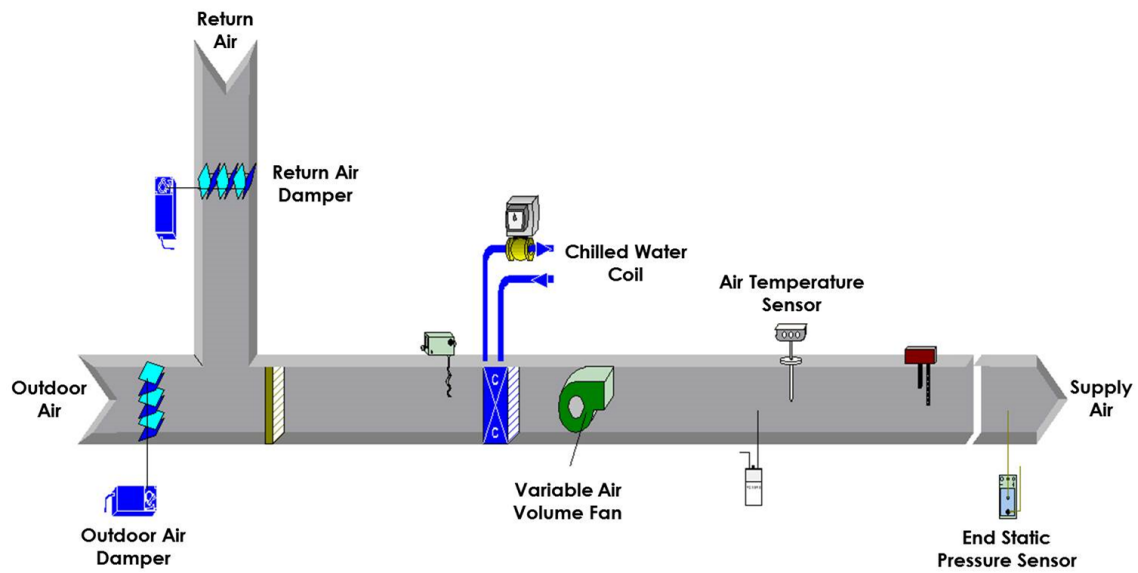


Figure 2.14: Rooftop Air Handling Unit (AHU) for the Utilities Business Office (UBO) at Texas A&M University (figure from Apogee Software).

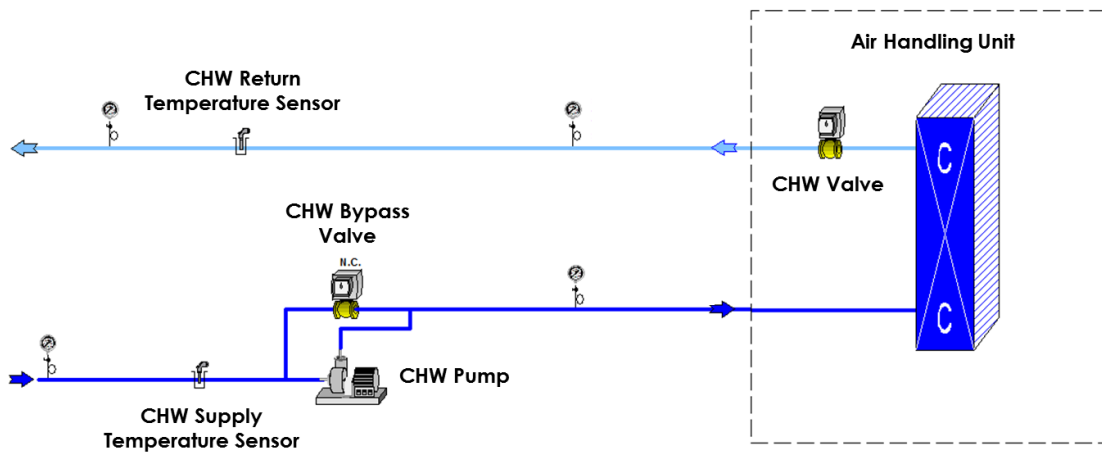


Figure 2.15: Chilled water loop for the Utilities Business Office (UBO) at Texas A&M University (figure from Apogee Software).

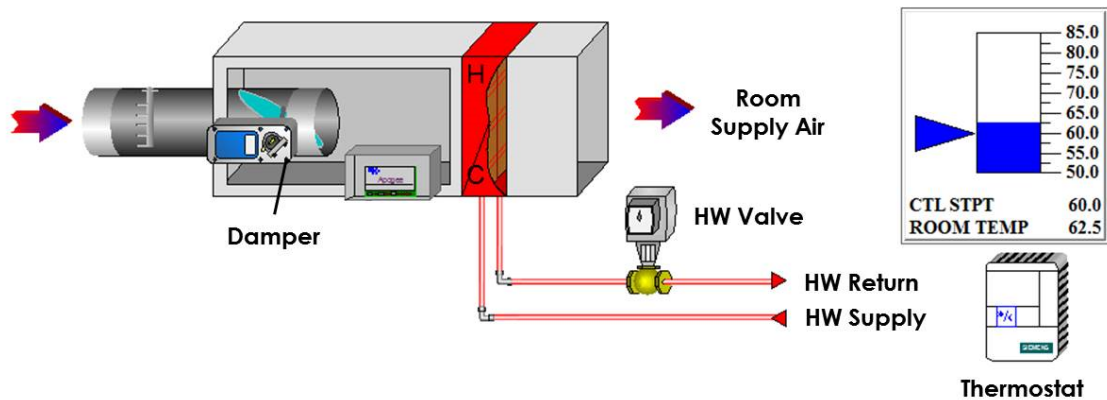


Figure 2.16: Variable Air Volume (VAV) box in the Utilities Business Office (UBO) at Texas A&M University (figure from Apogee Software).

the room is occupied, the zone VAV will heat the room to 70°F or cool the room to 74°F. If the room is unoccupied, the VAV will heat the room to 60°F or cool the room to 85°F. If the occupancy sensors in all the rooms read unoccupied, then the main AHU will turn off and the room temperatures will freely fluctuate.

Figure 2.17 visually displays the process flow and current control implementation described above. The proposed control methods presented in this dissertation operate as supervisory controllers, replacing the current setpoint calculations that need to be tuned by building operators.

The scope of this research effort was limited to developing economic cost functions for the AHU and the individual zones. For the AHU, two sources of cost were identified: 1) the cost of electricity used by the VAV fan to move the air, and 2) the cost associated with the production of chilled water for cooling of the air. At the zone level, there is no equipment that consumes a significant amount of power or resources. The damper in the VAV terminal box is the only actuated component and the power required to move it is negligible. As mentioned in Chapter 1, select previous building energy optimizations have

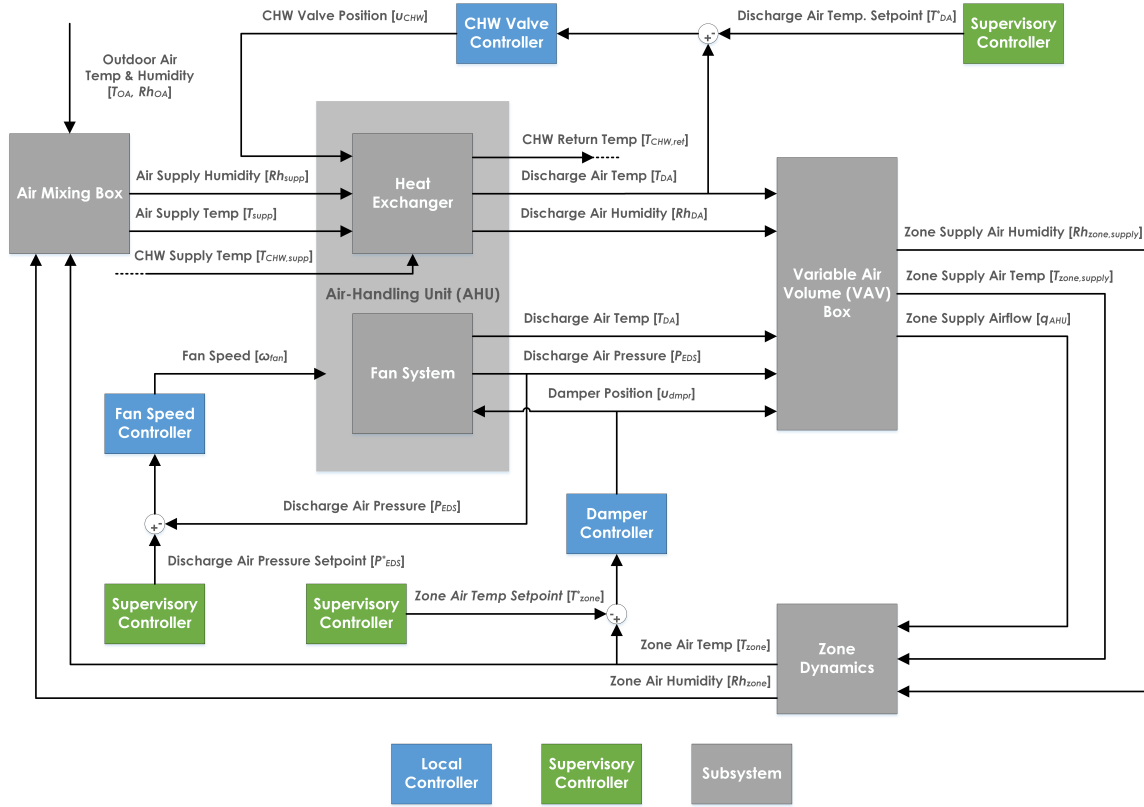


Figure 2.17: Process flow and current control implementation in the UBO.

included some form of occupant comfort, whether simply as constraints on the optimization, or as a measure to be minimized; however, to the author's knowledge, no previous effort has been reported in the literature that optimizes the economic cost of occupant discomfort in simulation or implementation. Thus, an economic objective function that minimizes the cost of occupant discomfort as a measure of the loss of productivity was developed at the individual zone levels. The following sections give an overview of the development of the three objective functions.

2.3.2 General Form of Component-Level Economic Objective Function

Considering the literature and previous works, a general component-level objective function of the form shown in Equation 2.1 was chosen. The quadratic terms (first and

third) were included to provide for standard convex optimization when desired, where e represents the error for the system, Q the weighting placed upon the error, u the control action, and S the weighting placed on control action. The linear terms (second and fourth) were included to facilitate calculating cost purely in economic terms (dollars), as opposed to nonsensical units such as dollars squared. R and T can be formulated in such a manner to transform e and u into economic cost, as will be demonstrated in the subsequent sections.

$$J_{component} = e^T Q e + e^T R + u^T S u + u^T T \quad (2.1)$$

2.3.3 Economic Objective Function for Cost of Power Consumed by Fan

As described above, the fan in the AHU works to maintain an end static pressure in the duct to move the required amount of air to condition the individual zones. To accomplish this, the fan motor requires electricity and there is a cost associated with the power used. A simple way to measure the power consumed by a fan motor would be use a power meter on the electrical lines to the fan and measure the power consumed; however, power meters are not often included in existing building energy systems and integrating a new one has its own associated costs to install and connect the device to the network. Therefore, the power consumed by the fan was instead determined by other available data, specifically the change in pressure across the fan and the volume flow rate of air being moved by the fan. With these two values, the work being performed by the fan on the air can be determined and converted into a measure of power, as power is work over time. The equation for work performed by a fan is given in Equation 2.2, where P_{fan} is the power consumed by the fan [W], 0.1175 is a conversion factor for imperial units, q is the air flow rate through the AHU [cfm], ΔP is the change in pressure across the fan [in. H₂O], and μ_f , μ_b , and μ_m are efficiencies for the fan blade, the fan belt, and the fan motor, respectively. The change

in pressure across the fan was taken from the end static pressure sensor in the AHU. As for the volume flow rate, the flow rate meters included in each of the VAVs were used in summation to calculate the total air flow rate through the AHU, assuming minimal duct losses and leaks.

$$P_{fan} = \frac{0.1175 \cdot q_{AHU} \cdot \Delta P}{\mu_f \cdot \mu_b \cdot \mu_m} \quad (2.2)$$

Recalling the need for the supervisory controller to overlay the local controllers for the most practical implementation as well as the general form of the proposed economic objective function in Equation 2.1, the most appropriate control action u for the AHU fan objective function would be the end static pressure setpoint. Assuming the local PID controllers have zero steady state error and sample significantly faster than the supervisory controller, it can be assumed that the end static pressure will equal the end static pressure setpoint. As such, an equation relating the end static pressure setpoint to economic cost is needed so that the cost of power for the fan can be minimized. Realizing that multiplying Equation 2.2 by an electricity cost rate [\$/kWh] and a duration of time, as well as recognizing that ΔP is equal to the end static pressure setpoint given enough time, then Equation 2.3 can be derived:

$$J_{fan} = \frac{0.1175 \cdot C_{elec} \cdot q_{AHU} \cdot P_{EDS}^* \cdot t_s}{1000 \cdot \mu_f \cdot \mu_b \cdot \mu_m} \quad (2.3)$$

where C_{elec} is the rate of electricity cost [\$/kWh], P_{EDS}^* is the end static pressure setpoint, and t_s is the sampling time of the supervisory controller. Reformulating Equation 2.3 to fit Equation 2.1, the result is given by Equation 2.4:

$$\begin{aligned}
J_{fan} &= \$ \text{ cost of eletricity used by fan} \\
J_{fan} &= e^T Q e + e^T R + u^T S u + u^T T \\
Q &= 0 \quad R = 0 \quad S = 0 \\
u &= P_{EDS}^* \quad T = \frac{0.1175 \cdot C_{elec} \cdot q_{AHU} \cdot t_s}{1000 \cdot \mu_f \cdot \mu_b \cdot \mu_m}
\end{aligned} \tag{2.4}$$

thus, establishing an objective function that can minimize the cost of electricity used by the AHU's VAV fan by optimizing the end static pressure setpoint. The overall trend of the cost of electricity used by the fan as end static pressure is varied is shown in Figure 2.18.

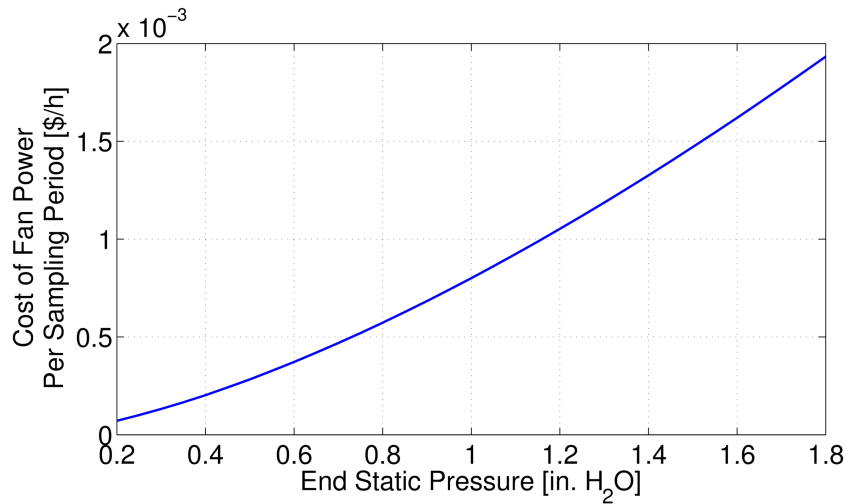


Figure 2.18: Trend of the objective cost of the fan as the end static pressure changes.

2.3.4 Economic Objective Function for Cost of Chilled Water Usage in AHU

In order to condition the air to provide cooling to the zones, chilled water is pumped through a chilled water coil over which air is passed in the AHU. To determine the economic cost of using the chilled water, the associated production cost of said chilled water was chosen. The UEM Office at Texas A&M University maintains utility usage data,

specifically the cost per unit of energy of the respective utility. For chilled water, this is the dollar cost associated with producing one mmBtu of chilled water at the campus wide chilled water temperature of 45°F. Starting with this value, a relationship is needed between the cost of chilled water in [\$/mmBtu] and a relevant setpoint in the AHU that can be optimized. The most sensible setpoint with regard to chilled water usage in the AHU is the discharge air temperature setpoint. This setpoint, as described above is given by a demand calculation to a PID loop that actuates the valve metering how much chilled water flows through the chilled water coil. To connect this setpoint with the relative cost term, the Equation 2.5:

$$\dot{Q} = \dot{m} \cdot c \cdot \Delta T \quad (2.5)$$

is used, where \dot{Q} is the rate of change of heat, or energy, \dot{m} is the mass flow rate, c is the specific heat of the respective fluid, and ΔT is the change in temperature of the fluid. If a mass flow rate sensor is available on the AHU for the chilled water, then this measurement can be used to calculate the rate of change of energy of the chilled water and thus the overall cost; however, mass flow rate sensors are not usually installed on chilled water lines at the AHU level. If there is not a mass flow sensor, then the mass flow rate can be converted to a volume flow rate by multiplying by the fluid's density, and it is this volume flow rate that can be determined for the chilled water in the AHU cooling coil. With the UBO, the approximate maximum flow rate through the chilled water valve was determined from previously recorded data. Assuming a linear valve/flow relationship, the chilled water valve opening (in percent) multiplied by the maximum possible flow will give the current volume flow rate of the chilled water. Using data values for the discharge air temperature setpoint and the chilled water valve position, a fit was generated to transform the setpoint to a valve position. This fit can be found in Appendix A. Combining the relationships

gives Equation 2.6

$$J_{CHW} = \alpha \cdot q_{max} \cdot \rho \cdot c \cdot \Delta T_{h2o} \cdot t_s \cdot 0.00341214 \left[\frac{mmBtu}{kWh} \right] \cdot \frac{3682}{241} \left[\frac{\$}{mmBtu} \right] \quad (2.6)$$

$$\alpha = (3.021T_{DA}^{*2} - 109.4T_{DA}^* + 1002)$$

where α is the fitted relationship between the discharge air temperature setpoint (T_{DA}^*) and the chilled water valve position, q_{max} is the maximum flow through the valve (at 100% opening), ρ is the density of water, c is the specific heat of water, ΔT_{h2o} is the change in temperature of the supply and return chilled water, t_s is the sample time of the supervisory controller, 0.00341214 is a conversion factor from kWh to mmBtu, and the last term is the economic cost for the UBO of the consumed chilled water. Reformulating Equation 2.6 to fit Equation 2.1 gives:

$$J_{CHW} = \$ \text{ cost of energy to produce chilled water}$$

$$J_{CHW} = e^T Q e + e^T R + u^T S u + u^T T$$

$$Q = 0 \quad R = 0 \quad S = 0 \quad (2.7)$$

$$u^T T = \alpha \cdot q_{max} \cdot \rho \cdot c \cdot \Delta T_{h2o} \cdot t_s \cdot 0.00341214 \left[\frac{mmBtu}{kWh} \right] \cdot \frac{3682}{241} \left[\frac{\$}{mmBtu} \right]$$

$$\alpha = (3.021T_{DA}^{*2} - 109.4T_{DA}^* + 1002)$$

where the entire linear cost term $u^T T$ is equal to Equation 2.6. This result actually lends itself well to traditional convex optimization of the setpoint T_{DA}^* , due to the quadratic nature of the fit. The overall shape of the objective function can be seen as the AHU discharge air temperature is varied in Figure 2.19.

2.3.5 Economic Objective Function for Cost of Discomfort in Zones

The last economic objective function that was developed for the UBO is the cost of occupant discomfort within each zone. In order to measure an occupants level of discomfort,

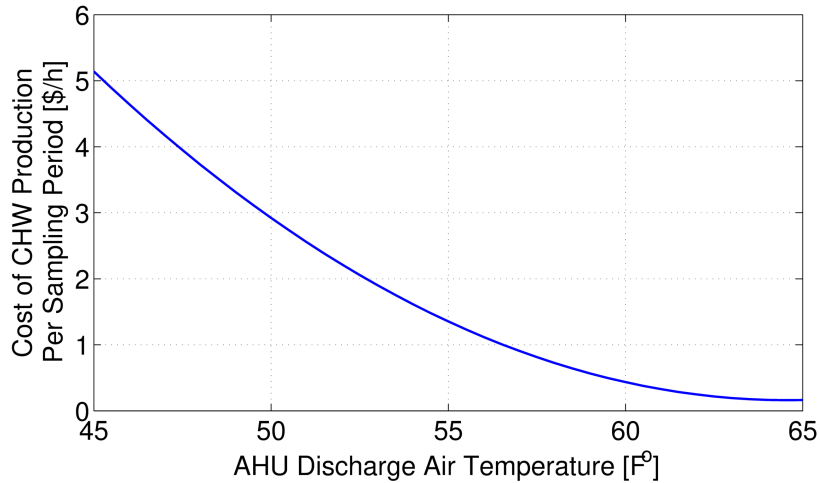


Figure 2.19: Trend of the objective cost of the chilled water used by the AHU as the AHU discharge air temperature changes.

many have relied on the use of predicted mean vote (PMV), developed by Fanger in the 1970's [4]. Briefly, PMV is a measure on the American Society of Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE) thermal sensation scale of -3 to 3, where negative numbers represent being too cold, positive numbers represent being too warm, and a value of zero represents being comfortable. Performing a large study of people, Fanger collected data regarding occupants votes and created an equation to determine the PMV across a variety of environmental factors. These factors include air temperature, air relative humidity, relative air speed, mean radiant temperature, an occupant's insulation level due to clothing, an occupant's metabolic rate, an occupant's work output, among several other variables. The details of the equation can be found in [4]. From PMV, Fanger determined the Predicted Percentage of Dissatisfied (PPD) of people. The relationship of PPD to PMV can be seen in Figure 2.20.

It can be seen that at -3 and 3 PMV that approximately 100% of the population would be dissatisfied. Worth noting is that even at zero PMV, 5% of the population, on average,

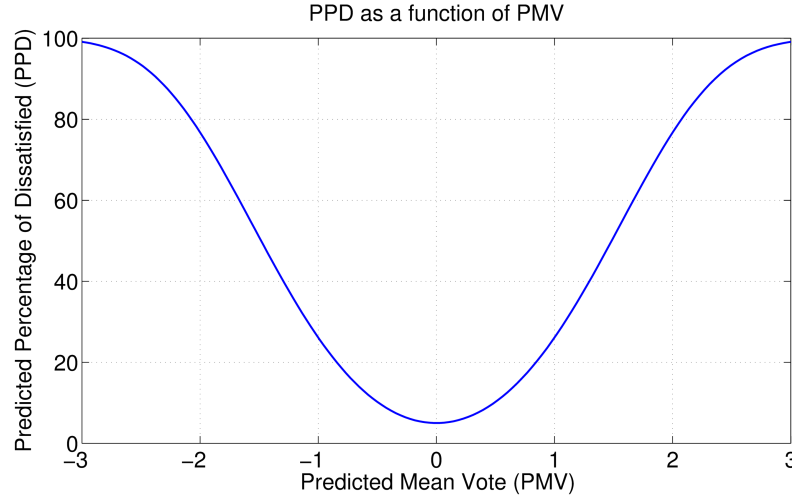


Figure 2.20: Relationship between PPD and PMV as calculated by Fanger [4].

will still be dissatisfied, attesting to the fact that each individual has specific preferences. While PMV will tell you how many people will be dissatisfied, a relationship to tie this to an economic cost is still needed. Fortunately, researchers have investigated the effect of occupant comfort on worker productivity, as mentioned in the literature review. One effort in particular tied PMV to a measure of Loss Of Productivity (LOP). By using regression analysis, a direct relation can be calculated between a worker's loss of performance and the PMV of an indoor climate by including the calculations of equivalent thermal situations from Gagge's two-layer human model [87] and Fanger's comfort equation [4]. For a detailed explanation, see [10]. The results of Roelofsen's work [10] are two sets of coefficients for a regression fit for the cold side of the PMV comfort zone and for the warm side of the PMV comfort zone. The regression is a 6th-order fit shown in Equation 2.8:

$$LOP = b_0 + b_1 PMV + b_2 PMV^2 + b_3 PMV^3 + b_4 PMV^4 + b_5 PMV^5 + b_6 PMV^6 \quad (2.8)$$

where LOP is the loss of productivity and b_0, \dots, b_6 are the regression coefficients. The

value of the Roelofsen's coefficients are repeated in Table 2.2, for reference. Roelofsen constructed the regression on the cold side to be zero at -0.5 PMV and for the warm side to be zero at 0 PMV, leaving a region between -0.5 to 0 PMV where LOP is zero. This is because several studies found a region of conditions near and below 0 PMV that had a negligible effect on productivity. Figure 2.21 shows the change in LOP as PMV varies.

Table 2.2: Regressions for Loss Of Productivity fit of PMV from [10].

Regression Coefficients	Cold Side of PMV Comfort Zone	Warm Side of PMV Comfort Zone
b_0	1.2802070	-0.15397397
b_1	15.995451	3.8820297
b_2	31.507402	25.176447
b_3	11.754937	-26.641366
b_4	1.4737526	13.110120
b_5	0.0	-3.1296854
b_6	0.0	0.29260920

To tie LOP to an economic cost, a simple multiplication of the lost productivity percentage by the amount of salary and employed earns over the sample period was used, shown in Equation 2.9 where β [\$] is the productivity lost in wages:

$$\beta = LOP \cdot \left(\frac{p_{year} \cdot t_s}{52\text{wks} \cdot 40\text{hrs}} \right) \quad (2.9)$$

Considering the individual zones with respect to Equation 2.1, it is noted that while there is no control action u at the zone level that consumes energy, there is an error signal present. That is the error e in Equation 2.1 is given by:

$$e = T_{zone} - T_{zone}^* \quad (2.10)$$

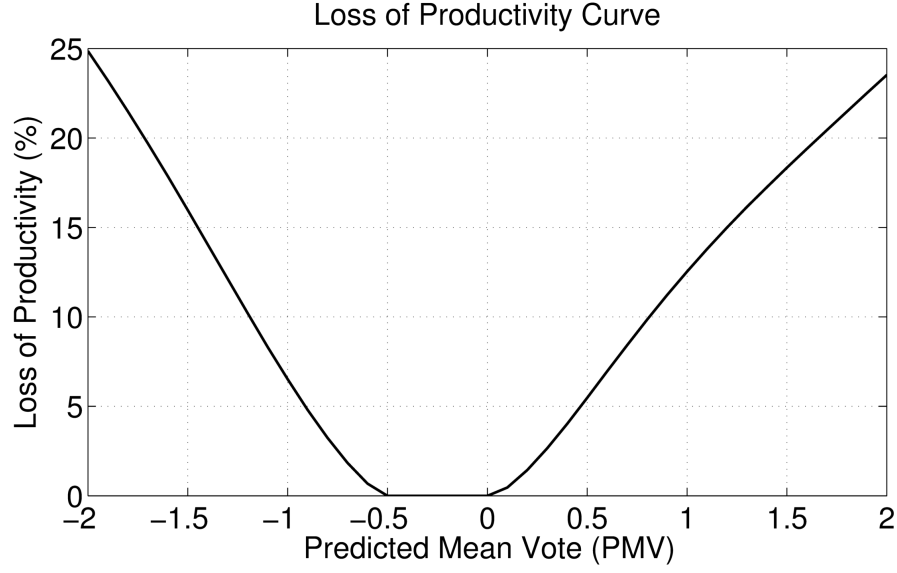


Figure 2.21: Shape of Loss Of Productivity function.

where T_{room} and T_{zone}^* are the zone temperature and zone temperature setpoint, respectively. The error is defined in this manner as this research focuses on the systems when in cooling mode; thus, the error will be mostly positive during cooling mode as the zone temperature will generally be above or at the setpoint temperature. If the zone were in heating mode, the sign of the error term would need to be reversed. Recognizing that the zone temperature setpoint can be optimized by the supervisory controller to minimize the LOP by optimizing the zone's PMV, and that the error signal is a difference of temperatures, the sensitivities of the above relationships can be determined and combined to give an economic objective function. In determining PMV, only the zone air temperature and zone air relative humidity are changing. As such, an accurate fit of the PMV equation was created and used to reduce computation time and complexity. The details of this fit can be found in Appendix B. The fit is defined as:

$$PMV = 0.5542 \cdot Rh_{zone} + 0.23 \cdot T_{zone} - 5.44 \quad (2.11)$$

Equation 2.11 is used to determine the sensitivity of PMV to changes in air temperature. The resulting combination of sensitivities is shown in Equation 2.12:

$$\begin{aligned}
J_{room} &= e^T Q e + e^T R + u^T S u + u^T T \\
Q &= 0 \quad S = 0 \quad T = 0 \\
R &= \left[\frac{\partial PMV}{\partial T_a} \right] \cdot \left[\frac{\partial LOP}{\partial PMV} \right] \cdot \left[\frac{\partial \beta}{\partial LOP} \right]
\end{aligned} \tag{2.12}$$

where the sensitivities are determined to be:

$$\left[\frac{\partial PMV}{\partial T_a} \right] = 0.23 \tag{2.13}$$

$$\begin{aligned}
\left[\frac{\partial LOP}{\partial PMV} \right] &= b_1 + 2b_2 PMV + 3b_3 PMV^2 + \\
&4b_4 PMV^3 + 5b_5 PMV^4 + 6b_6 PMV^5
\end{aligned} \tag{2.14}$$

$$\left[\frac{\partial \beta}{\partial LOP} \right] = \left(\frac{p_{year} \cdot t_s}{52\text{wks} \cdot 40\text{hrs}} \right) \tag{2.15}$$

where PMV is the predicted mean vote, p_{year} is an occupant's annual salary, and t_s is the sampling time of the supervisory controller, in hours. The cost calculation assumes that if more than one occupant is in a zone, the sum of the occupant's salaries is used for p_{year} and that a standard 40 hour work week is used. For Equation 2.14, the coefficients vary as described by Table 2.2. If the zone's PMV falls within the range of -0.5 to 0, then Equation 2.14 is equal to zero. Thus, an economic objective function that minimizes the cost of productivity loss by optimizing the zone temperature setpoint was developed.

The overall trend of the objective function as the zone air temperature varies is shown in Figure 2.22. The abrupt changes occur at the PMV values of -0.5 and 0. This is due to the fit of the LOP equation. It is worth noting that this curve will shift depending

on other variables in the objective function that are varied, such as annual salary and relative humidity of the zone. Also, depending on the value of the user-defined setpoint, the objective cost can become negative. This does not mean that the zone is earning money, but occurs due to the structure of the objective function. An optimization competition can occur between the error term and the coefficient R . As the error is defined as the difference in the zone temperature and zone temperature setpoint, the objective cost will be zero when the zone reaches this defined setpoint; however, if this setpoint is not equal to the optimal comfort temperature, as determined by the PMV function, then the objective cost will also be zero if the zone air temperature is equal to the optimal comfort temperature. This behavior presents an interesting control question. To operate at the most cost effective point, as defined by the objective function, it would require the zone temperature setpoint to be set to the PMV optimal temperature, but in doing so, the ability for occupants or building managers to provide the system feedback about their comfort is removed. So which is more important: the ability for occupants to choose their zone temperatures or

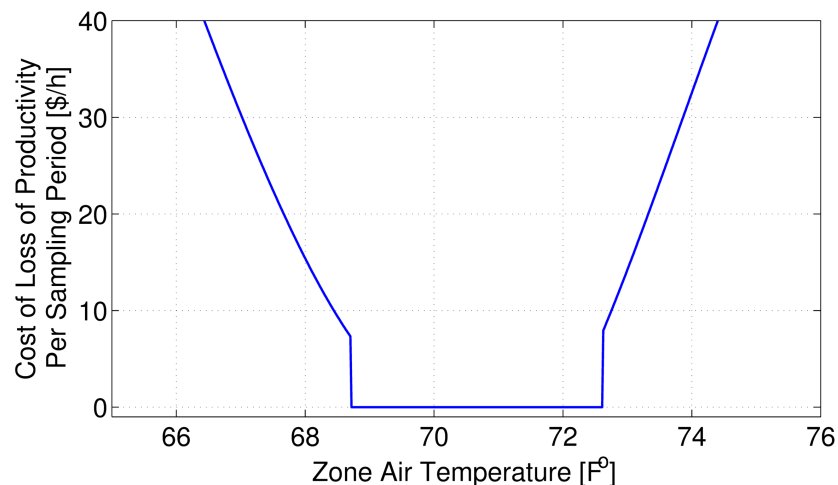


Figure 2.22: The change in objective cost of the loss of productivity due to discomfort due to changes in zone air temperature during a 15 minute period.

allow the system to determine what is best for the occupants? This question deserves additional investigation but is beyond the scope of this publication. Fortunately, with how the objective function is structured currently, the system will propose a compromise: a temperature between the user-defined setpoint and the PMV optimal comfort temperature.

For a centralized implementation, the objective function then becomes:

$$J_{total} = J_{fan} + J_{CHW} + \sum_{i=1}^n J_{zone,i} \quad (2.16)$$

where n is the number of zones in the system. Equation 2.16 is the system objective function that will be minimized in the optimizations.

2.3.6 MATLAB and EnergyPlus Co-Simulation Design

A model of the UBO was first created with the aid of SketchUp [88], a 3D modeling software. Using dimensions taken from the building, the single story layout was replicated. Utilizing the plug-in from OpenStudio, thermal zones, boundaries, and interactions were defined. This model was then exported as an input file (IDF) for EnergyPlus [89], an open-source energy simulation program that has been developed by the Building Technologies Office (BTO) within the U.S. Department of Energy (DOE). Using EnergyPlus, HVAC equipment was added to the model based on the equipment present in the UBO. The IDF file for EnergyPlus can be viewed in Appendix D. While EnergyPlus excels at modeling and simulation, it is not immediately accessible for controller development and implementation. As such, controllers for the AHU and the zone level VAVs were created in MATLAB. To enable co-simulation between EnergyPlus and MATLAB, two programs were used. The first was MLE+ [90], an open-source MATLAB toolbox for creating the necessary configuration files and providing functions to connect EnergyPlus and MATLAB with an easy-to-use graphical interface. MLE+ utilizes the Building Controls Virtual Test Bed (BCVTB) [91] as the communication backend between EnergyPlus and MAT-

LAB, providing the co-simulation functionality. The MATLAB code for the simulations can be found in Appendix E.

The model developed in SketchUp is shown in Figure 2.23. The individual zones were setup as defined in the building layout and exterior doors & windows were placed as accurately as possible. The AHU and zone VAVs were added using EnergyPlus' HVACTemplate objects. A central electric chiller was also added to supply chilled water to the AHU. The chilled water output temperature is regulated to 45°F, the same as the supply chilled water temperature for the UBO. Currently, EnergyPlus does not offer modeling of pressure with variable air volume systems, thus another solution was necessary to simulate the UBO's control and physical limitations of end static pressure and flow in the AHU. In order to accomplish this, it was assumed that the dynamics of the fan speed and end static pressure were fast enough compared to the simulation timestep (1 minute) to be considered instantaneous. Additionally, it was assumed that the fan would supply the requested end static pressure, constrained by the physical limitations of the AHU and ducting. To

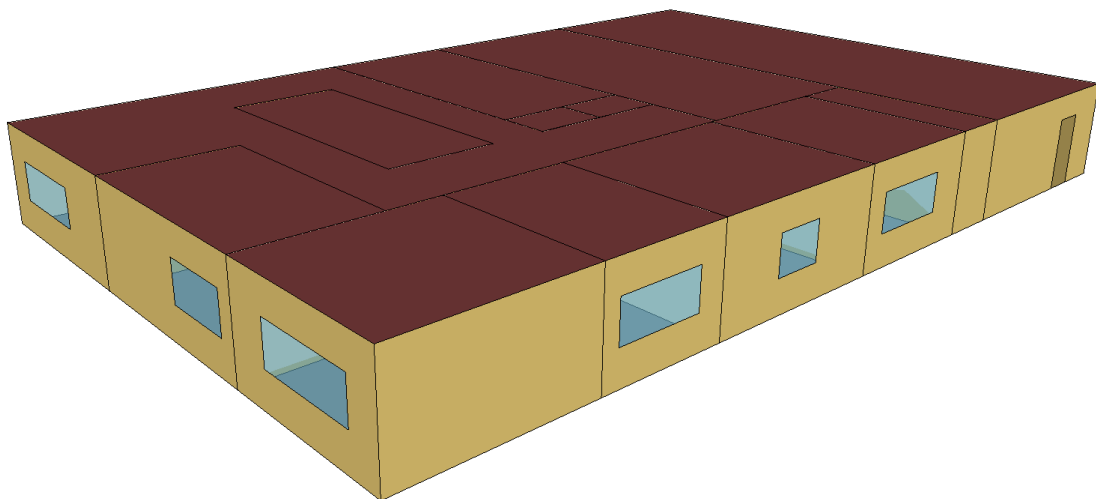


Figure 2.23: SketchUp model of the Utilities Business Office (UBO) at Texas A&M University.

determine this constraint, data from the real UBO building was analyzed and a maximum possible work performed by the fan was calculated (910 W). During the optimizations, the constraint is calculated by using Equation 2.18. To determine the total volume flow through the AHU, individual models of the zone VAVs were generated from data based on VAV damper position and end static pressure. A fit of the form shown in Equation was used for the VAV models, taken from [92].

$$q_{VAV} = (a \cdot u_d^2 + b \cdot u_d) \cdot \sqrt{P_{EDS}} \quad (2.17)$$

where q_{VAV} is the air flow through the VAV [cfm], u_d is the damper position [%], P_{EDS} is the end static pressure [in. h₂O], and a, b are identified coefficients. All of the VAV models can be found in Appendix A. The effect of outdoor air temperature was also considered on the VAVs as the AHU draws in outdoor air, but was shown to be minimal. This is most likely due to the fact that AHU is able to meet its discharge air temperature setpoint, even at varying flows, effectively isolating the VAV supply air from the outdoor air conditions. The flows from the VAV models are summed to obtain the total flow through the AHU, assuming minimal duct losses. The constraint can then be written as follows:

$$910[W] \geq \frac{0.1175 \cdot q_{AHU} \cdot P_{EDS}}{\mu_f \cdot \mu_b \cdot \mu_m} \quad (2.18)$$

Thus, the optimization will only ever choose an end static pressure setpoint that is physically achievable by the system. This end static pressure is then passed to the VAV models which, along with the commanded damper positions, produce individual zone air volume flows.

The overall control hierarchy can be seen in Figure 2.24. The supervisory controller supplies the zone temperature setpoints (T_{ZONES}^*) to the respective PID reference inputs and the end static pressure setpoint (P_{EDS}^*) to the VAV models. The discharge air tem-

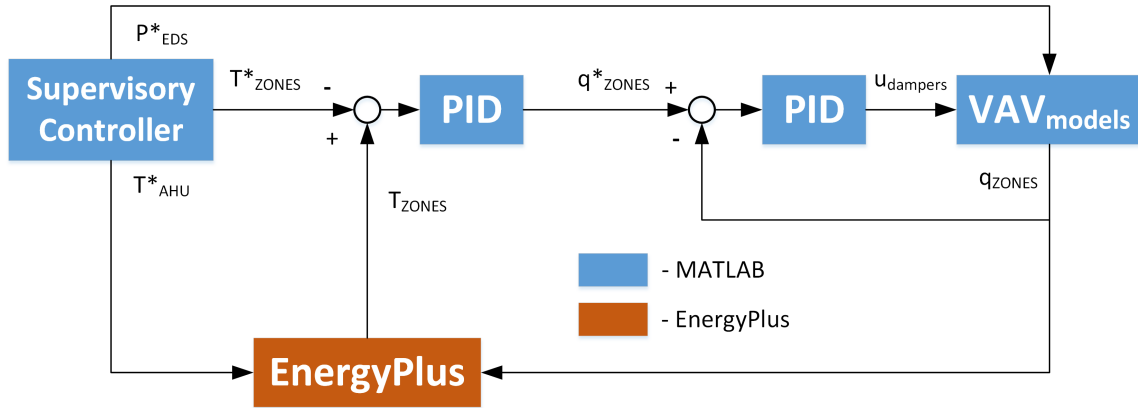


Figure 2.24: Control hierarchy used in simulation of the UBO.

perature setpoint (T^*_{AHU}) is supplied directly to EnergyPlus as the control for the chilled water valve is implemented using an appropriate setpoint manager within EnergyPlus. The zone temperature error is fed to the cooling PID controller which outputs a desired percentage of maximum flow setting (0% to 100%). This signal is passed through block G which maps the maximum flow percentage to the range of minimum flow to maximum flow. If the minimum flow setting for a zone is zero, then the signal remains unchanged. The desired flow percentage is then passed to the flow PID controller which then produces a desired damper position. This damper position is then used in the VAV models as previously described. The output from the VAV models, desired zone air volume flows, are converted to air flow fractions and then sent to zone VAVs in EnergyPlus where the room dynamics are simulated for one timestep.

2.3.7 Zone Temperature Models

For the supervisory controllers, models are necessary to predict the room temperatures as the setpoints are optimized. A previously developed modeling algorithm was employed to generate models for the individual zones. The full details of this process can be found in [93]. An overview of the process is given here. Data from the EnergyPlus simulation is

analyzed to determine significant coupling interactions between zones. The temperature of these zones along with other inputs such as outdoor air temperature, outdoor air relative humidity, AHU discharge air temperature, end static pressure, and zone temperature setpoints are used as inputs to ARX, ARMAX, and Box-Jenkins modeling methods with the output being the respective zone temperatures. The best fitting model is selected and then the individual models are combined into a centralized model of the entire system. Predictions from this centralized model are then used by the supervisory controller to optimize the UBO's 12 setpoints (discharge air temperature setpoint, end static pressure setpoint, and 10 zone temperature setpoints) to minimize the economic cost functions previously described. The identified models can be written in a discrete, linear state-space form as follows:

$$\begin{aligned}x_i(k+1) &= A_i x_i(k) + B_i u_i(k) + K_i e_i(k) \\ y_i(k) &= C_i x_i(k)\end{aligned}\tag{2.19}$$

where x_i is the state vector at a given timestep k , u_i is the input vector corresponding to each model, and y_i denotes the zone air temperature. Normally $e_i(k)$ is considered as a random component representing a white-noise disturbance; however, in the simulations $e_i(k)$ represents the error between the previous prediction and the current measured value, relying on the assumption that the error is approximately Gaussian. Using the error as $e_i(k)$ improves the models ability to predict the zone temperatures and thus improves control performance. The identified matrices for all the zone models can be found in Appendix A. The prediction results from the dynamic model for Zone 4 of the UBO are shown in Figure 2.25. The 30 minute ahead prediction recreates the measured temperature fairly accurately, with only minor differences. The other zone models show similar predictive capabilities.

The inputs and outputs that were provided to the modeling algorithm for each zone are listed in Table 2.3. Because the last input includes the output of the models, the identified

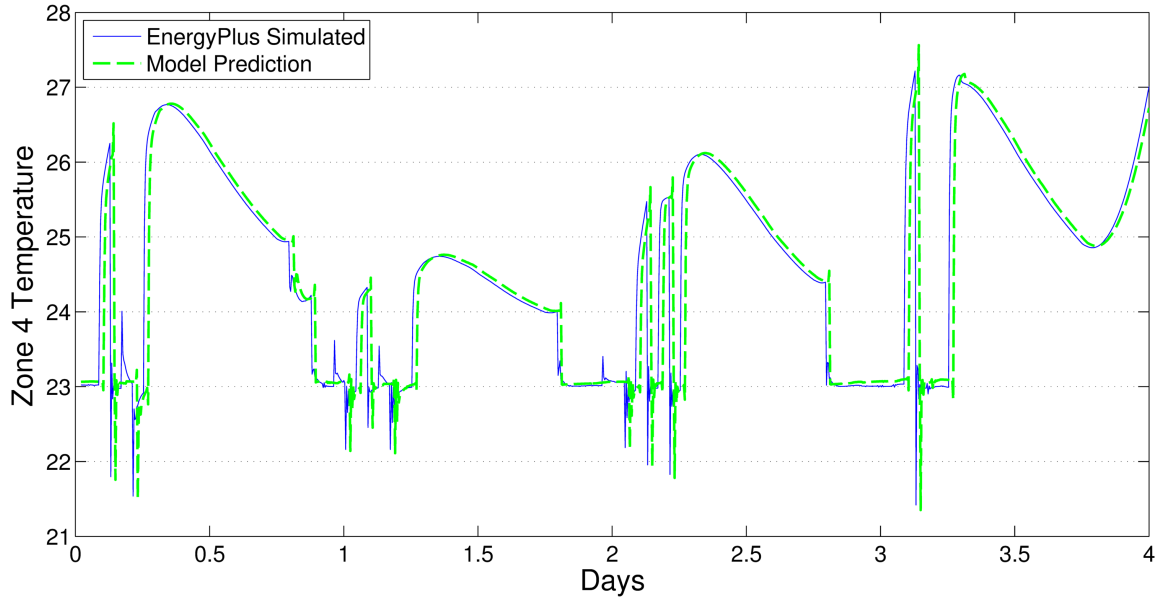


Figure 2.25: Zone 4 temperature prediction 30 minutes ahead using identified models.

models required modification so that the output is not part of the input vector. This is accomplished by subtracting the model output from the input vector and substituting that input vector back into the identified state-space model, resulting in modifications to the state matrix A and the error matrix K . This manipulation is detailed in Appendix C.

Models were identified for two operational cases: 1) where the zone VAV damper still had actuator range (i.e. the damper was not fully open), and 2) where the zone VAV

Table 2.3: Inputs and outputs for the generated models.

Model Input	Model Output
T_{OA}	$T_{zone,i}$
Rh_{OA}	
$T_{zone,i}^*$	
T_{AHU}^*	
P_{EDS}^*	
$T_{zone\ dist.,j} - T_{zone,i}$	

damper is fully open. The separate models were necessary as the effect of the model inputs varies greatly between the two cases. In case 1, the effect of T_{AHU}^* and P_{EDS}^* are minimal compared to $T_{zone,i}^*$. This is due to the fact that the VAV damper is actuated by a PID controller with $T_{zone,i}^*$ as the reference. The PID controller is able to reject changes in T_{AHU}^* and P_{EDS}^* by changing the damper position to achieve $T_{zone,i}^*$. However, in case 2 when the damper is fully open, T_{AHU}^* and P_{EDS}^* become the inputs of significance as they directly effect the zone temperature, determining the amount and the temperature of the incoming conditioned air. For the dynamic MPC controller, the models were used in the state-space form described by Equation 2.19. For the steady-state optimal control method, the steady-state outputs were calculated as:

$$y_i(k) = [C_i(I - A_i)^{-1}B_i] u_i(k) + e_i(k) \quad (2.20)$$

It was then necessary to determine when each model should be used in the simulations. This decision was made based off of two conditions. The first determined if the predicted zone temperature using the case 1 models was greater than the prediction from the case 2 models. This condition served to verify whether the predicted temperature from the case 1 models was currently achievable with the state of the AHU. If the case 1 predicted temperature was lower than the case 2 predicted temperature, then the VAV wouldn't be able to achieve the case 1 predicted temperature with the current T_{AHU}^* and P_{EDS}^* values. The second condition checked if the current VAV damper position was less than 95%, or in other words if the VAV still had actuator range of the damper. The value of 95% was used as opposed to 100% to serve as a threshold and help prevent the system from oscillating between cases. If both these conditions were true, then the case 1 models were used; otherwise, the case 2 models were used. The EnergyPlus and MATLAB code used in the simulations can be found in Appendices D and E.

2.3.8 Summary of Optimization

For the proposed steady-state controller, the centralized objective function that is minimized can be written as:

$$\min_{[T_{zone,i}^*, T_{DA}^*, P_{EDS}^*]} = J_{fan}(P_{EDS}^*) + J_{CHW}(T_{DA}^*) + \sum_{i=1}^n J_{zone,i}(T_{zone,i}^*) \quad (2.21)$$

where n is equal to the number of zones. For the proposed MPC controller, the centralized objective function to be minimized can be written as:

$$\min_{[T_{zone,i}^*, T_{DA}^*, P_{EDS}^*]} = \sum_{j=1}^m \left(J_{fan,j}(P_{EDS,j}^*) + J_{CHW,j}(T_{DA,j}^*) + \sum_{i=1}^n J_{zone,i,j}(T_{zone,i,j}^*) \right) \quad (2.22)$$

where m is equal to the length of the prediction horizon and n is equal to the number of zones. Both optimizations are subject to the constraints:

$$\begin{aligned} \frac{0.1175 \cdot q_{AHU} \cdot P_{EDS}^*}{\mu_f \cdot \mu_b \cdot \mu_m} &\leq 910 \\ \Delta T_{DA}^* &\leq 0.5 \\ \Delta P_{EDS}^* &\leq 0.15 \end{aligned} \quad (2.23)$$

where the first inequality is the constraint on the work done by the fan described previously in Equation 2.18, and the second and third inequalities are constraints placed on the change in the AHU discharge air temperature setpoint and the end static pressure setpoint. The second and third constraints are meant to prevent drastic changes that could place undue stress on the equipment. For the EnergyPlus simulations where the mass flow rate of the chilled water is available, the chilled water cost function was modified slightly. A specific fit for the simulated chilled water valve was generated between the discharge air temperature setpoint and the chilled water mass flow rate. This fit was used in the chilled

water cost function, redefined from Equation 2.7 as:

$$J_{CHW} = \alpha \cdot c \cdot \Delta T_{h_2o} \cdot t_s \cdot 0.00341214 \left[\frac{mmBtu}{kWh} \right] \cdot \frac{3682}{241} \left[\frac{\$}{mmBtu} \right] \quad (2.24)$$

$$\alpha = \left(\frac{-332.9 \cdot T_{DA}^{*2} + 1.089e4 \cdot T_{DA}^* - 6.802e4}{T_{DA}^{*2} + 7787 \cdot T_{DA}^* - 7.606e4} \right)$$

The MPC simulations used prediction horizons of 30 minutes, computed in 5 minute intervals. The steady-state control method simulations performed the optimization every 15 minutes, a time that allows for most of the dynamics to dissipate. The EnergyPlus simulation for both methods is run at 1 minute timesteps, which is the smallest timestep that the program allows. The MPC simulations assumed perfect knowledge of future outdoor weather conditions and occupancy profiles. This data could be replaced with estimates from weather forecasts and occupant's schedules in a practical application.

2.3.9 EnergyPlus and MATLAB Co-Simulation Results

Simulations were performed with both the steady-state optimal control method and the MPC controller in several different operational cases. These cases included constant occupancy throughout the workday, random occupancy with standard zone temperature setbacks (to simulate a more realistic working environment), random occupancy with optimal temperature setbacks computed by the minimization function, and a situation in which one room is prioritized over others, or a Very-Important-Person (VIP) case. The performance of both the algorithms are compared to the that of the previously described control strategy implemented in the UBO with temperature setpoints that produce a PMV equal to -0.25. A value of PMV = -0.25 was chosen to be the LOP optimal value as it is in the middle of the loss of productivity range defined to be zero, between PMV = -0.5 and PMV = 0.

2.3.10 Constant Occupancy Simulation

In this simulation the zones are assumed to be constantly occupied between the hours of 7 AM and 6 PM. The user-defined temperature setpoints are equal to the air temperature that will give a PMV of zero based on the current zone's relative humidity. Figure 2.26 shows how the zone temperatures vary throughout the day using the steady-state control method. Initially, the temperatures drop drastically as the rooms become occupied at 7 AM and the setpoints are lowered. Then majority of the rooms come up to a temperature of approximately 22.5 °C, which is associated with a PMV of slightly less than zero. Some variations in the room temperatures are observable, though the zones approximately stay within the -0.5 to 0 PMV range. These variations are due to the couplings that were identified in the modeling process between the different rooms. The optimization may lower some room setpoints to assist with the cooling of other rooms. This is most notable after approximately 2:30 PM when the greatest cooling demand occurs during the day. Specifically, zone 1's damper has become fully open. In order to ensure Zone 1 receives the cooling it needs, Zone 6's temperature is lowered. The discharge air temperature is also lowered significantly after 4 PM (shown in Figure 2.29). In response to the cooler discharge air temperature, many of the other zones reduce their damper openings resulting in an increase in the end static pressure (shown in Figure 2.31).

Compared to the steady-state control method, the MPC controller exhibited fewer variations in zone temperatures throughout the day, shown in Figure 2.27. The zone temperatures begin to drop at 6:30 AM as the prediction horizon of the MPC controller enables automated pre-cooling of the building. This pre-cooling action allows for the zones to be much closer to their optimal loss of productivity values at the start of occupancy at 7 AM. Again, around 3 PM during the highest cooling demand of the day, Zone 1 can be seen experiencing a slight rise in temperature. During this period, Zone 1's damper is fully open,

relying on the surrounding zones as well as the temperature and flow rate of the incoming conditioned air to maintain the zone's temperature. Figure 2.40 shows the increase in Zone 1's temperature (solid blue) as well as the 0 and -0.5 PMV thresholds corresponding to the limits of zero loss of productivity (dashed red and dashed green, respectively). Zone 1's temperature rises slightly above the 0 PMV limit, with the greatest difference equaling approximately 0.3 °C. This time above the 0 PMV threshold is due to the MPC controller determining the optimal setpoints that minimize the trade-off between increased comfort and increased cooling costs.

Overall, the two methods show similar strategies with the discharge air temperature (Figures 2.29 and 2.30) and the end static pressure (Figures 2.31 and 2.32). One main difference is that the MPC controller employs a higher end static pressure in the beginning of the simulation while the steady-state method has a rise in end static pressure near the end of the workday. Also, the steady-state method tends to display larger changes in the control variables compared to the MPC controller, which could be due in part to the difference in optimization timesteps (15 minutes vs 5 minutes, respectively).

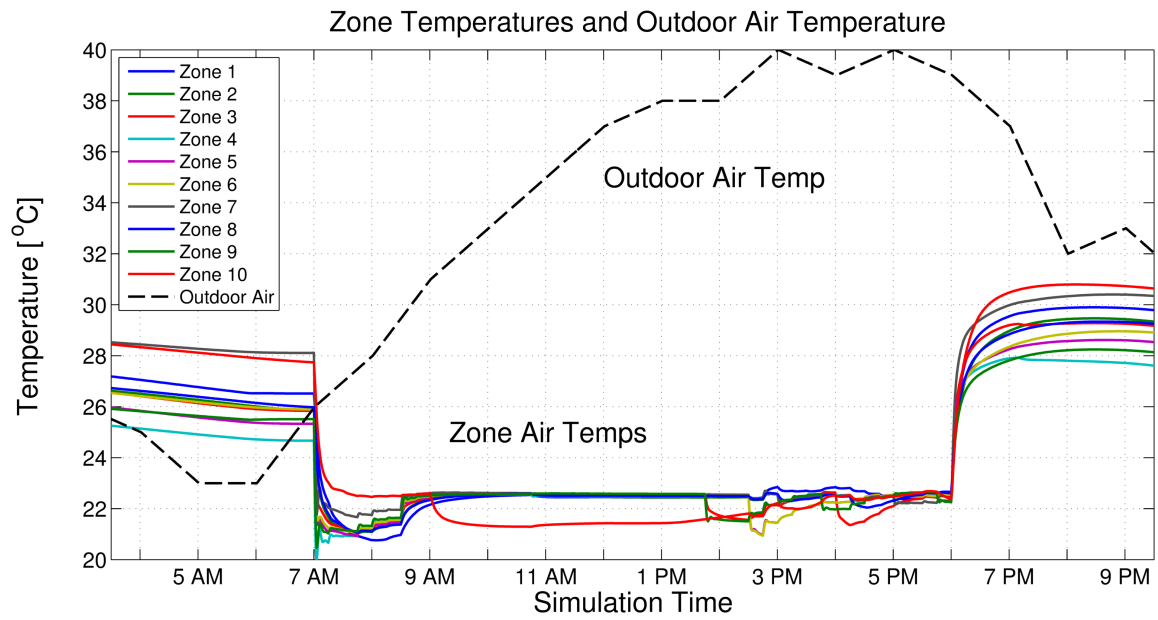


Figure 2.26: Zone temperatures for the steady-state method with constant occupancy.

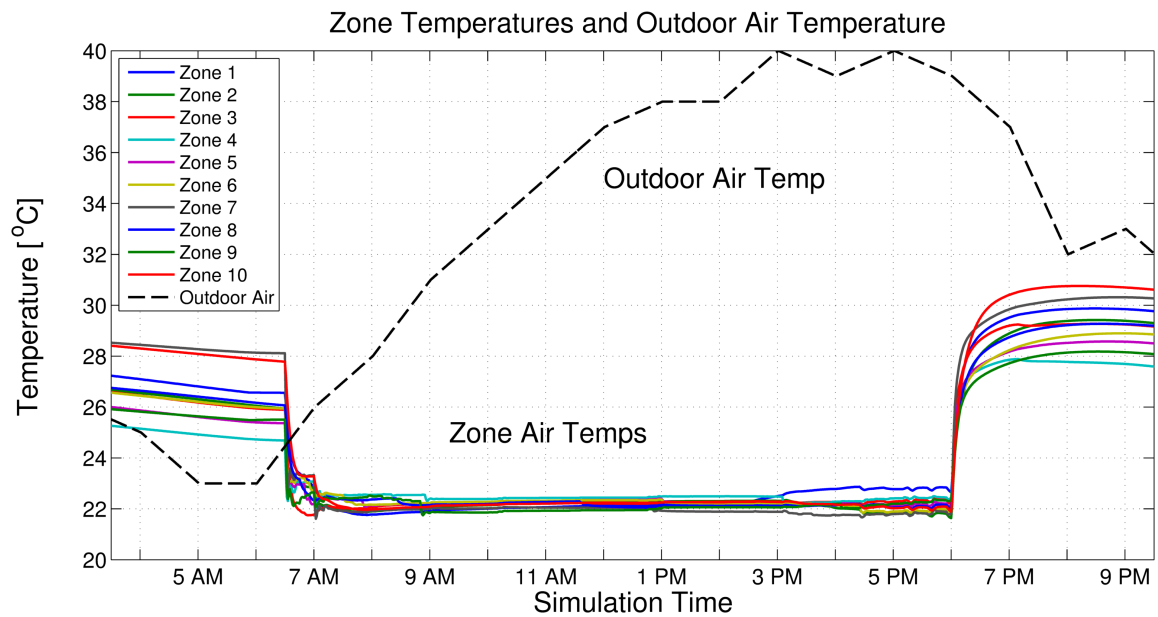


Figure 2.27: Zone temperatures for the MPC controller with constant occupancy.

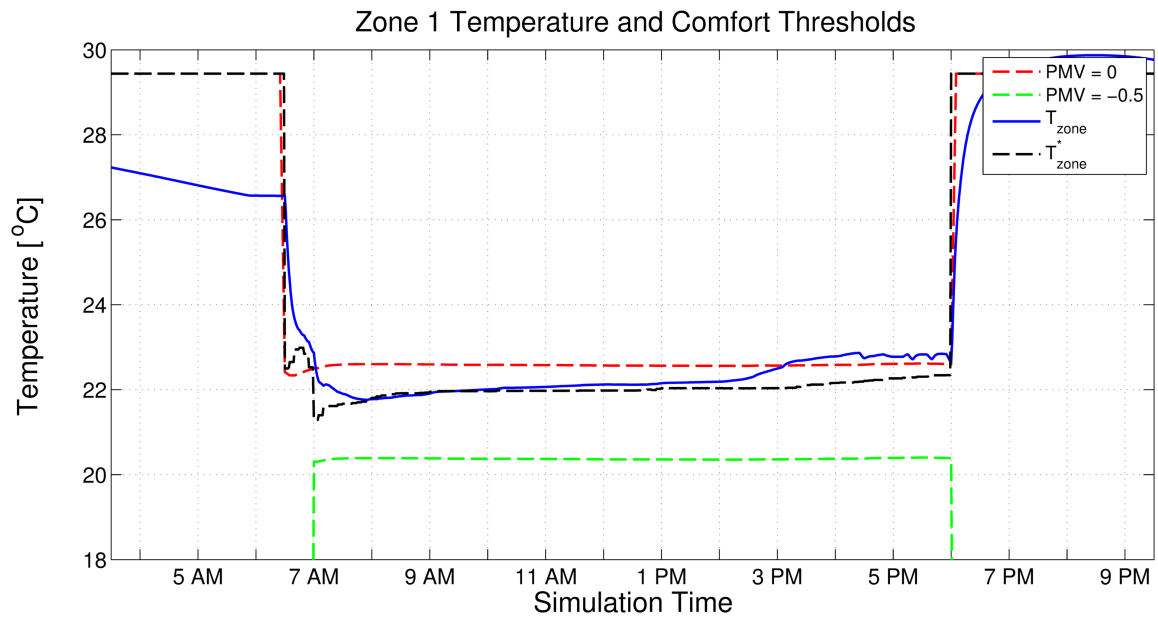


Figure 2.28: Zone 1 temperature for the MPC controller.

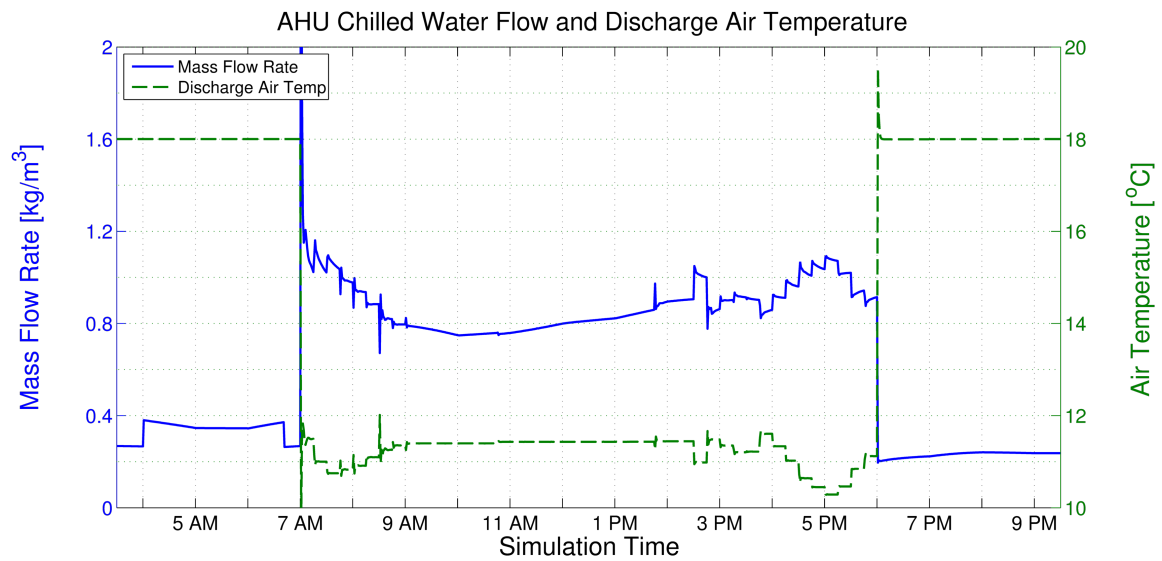


Figure 2.29: Mass flow of chilled water and the discharge air temperature for the steady-state method.

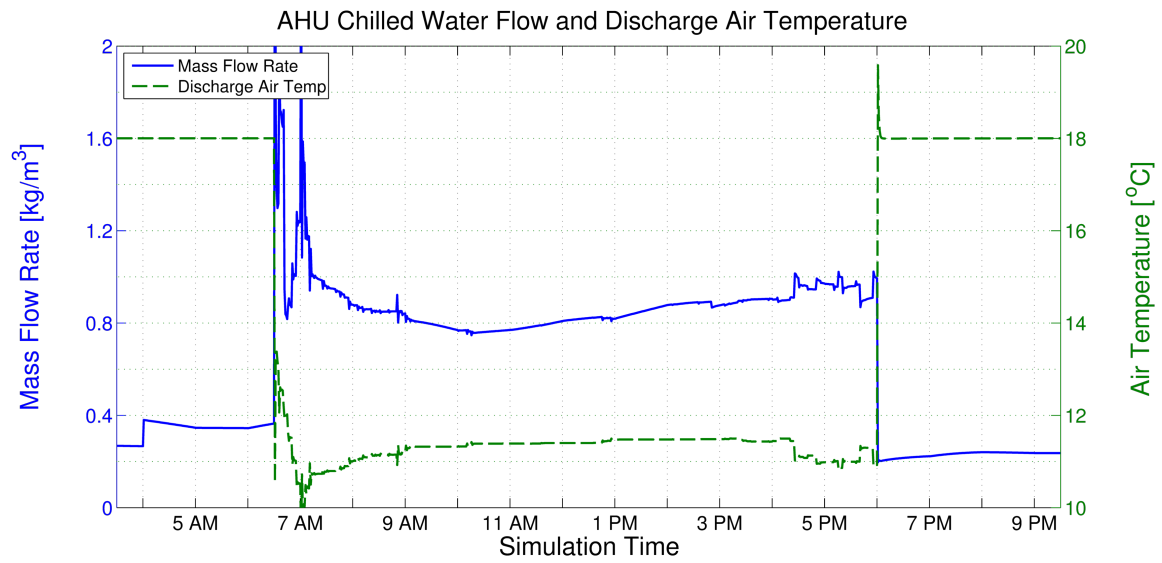


Figure 2.30: Mass flow of chilled water and the discharge air temperature for the MPC controller.

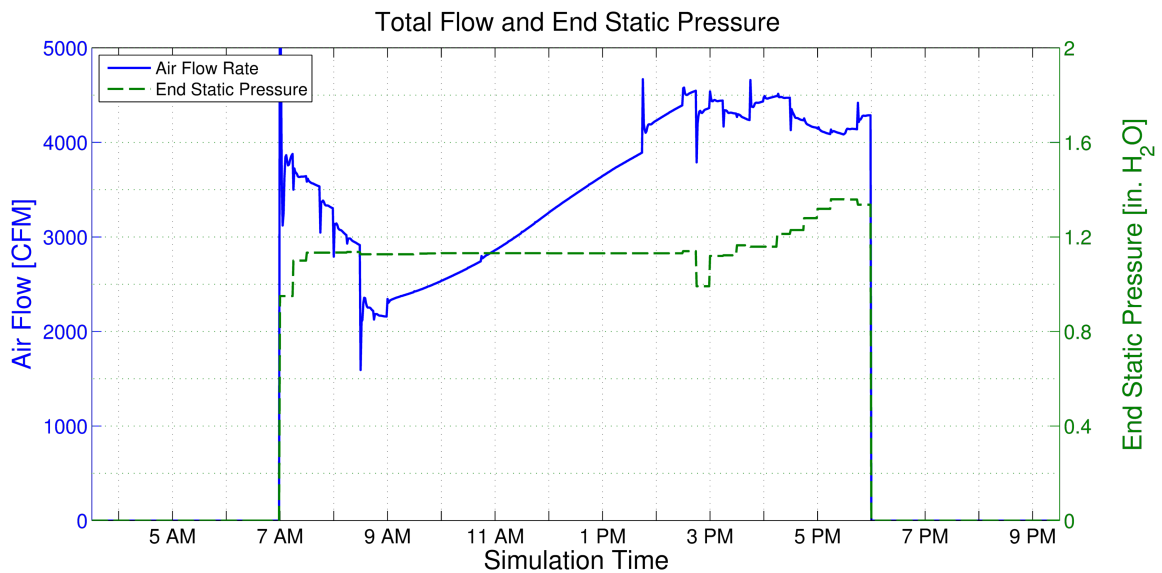


Figure 2.31: AHU air flow and end static pressure for the steady-state method.

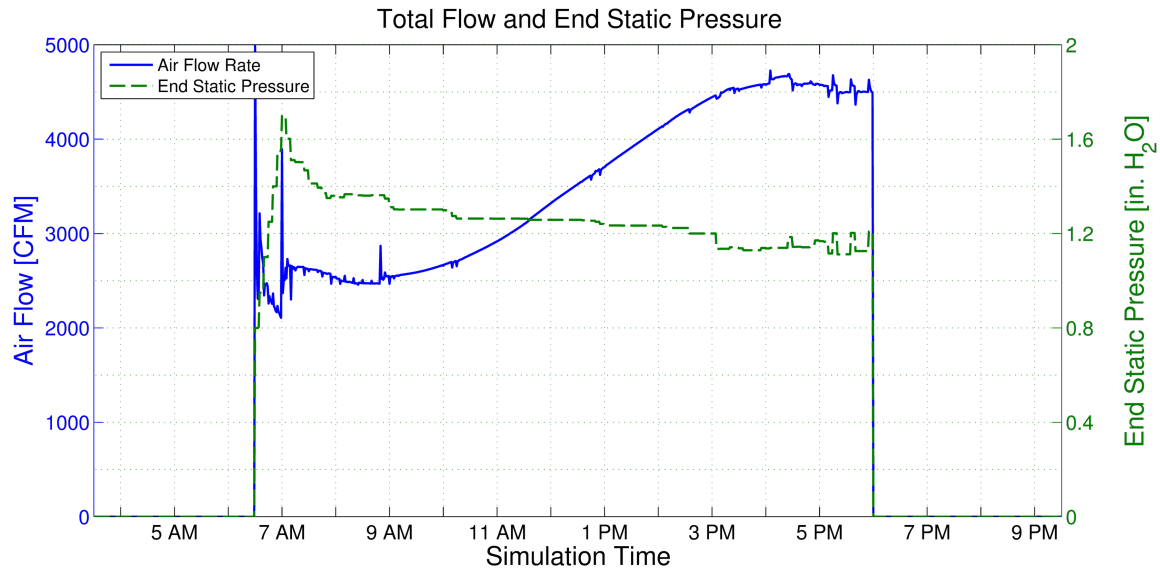


Figure 2.32: AHU air flow and end static pressure for the MPC controller.

The economic costs associated with loss of productivity due to discomfort as well as utilities from each of the simulations were annualized using cooling degree day data for College Station, TX. The results are shown in Table 2.4. The first row of Table 2.4 shows the costs associated with the current control methodology and technician-defined zone temperature setpoints. The second row indicates that for an increase in utility cost of \$1,085.16 per year, the combined economic cost of discomfort and utilities in the UBO could be reduced by \$5,307.85, or 38.1%. Essentially by just altering the zone temperature setpoints from 23 °C to approximately 22 °C, significant economic savings in productivity can be had from decreases in discomfort. For comparing the steady-state and MPC control methods, the current control method with the PMV optimal setpoints will be used as the baseline. The steady-state control method gives an 8.7% reduction in utility costs and a 10.24% reduction in overall costs from the baseline. The MPC control method gives an 8.4% reduction in utility costs and a 13.3% reduction in overall costs. Although the MPC method has a slightly higher utility cost than the steady-state, it is able to provide better

comfort control. This is partly due to the pre-cooling as a result of the prediction horizon, as well as determining optimal trajectories of setpoints versus the steady-state predictions.

Table 2.4: Annualized economic costs for constant occupation profiles.

	Zone			Total	
	Comfort Cost	CHW Cost	Fan Cost	Utility Cost	Total Cost
Current Control w/ User Setpoints	\$6,869.38	\$4,093.61	\$1,255.67	\$7,046.03	\$13,915.41
Current Ctrl. w/ PMV Setpoints	\$476.37	\$4,562.48	\$1,677.62	\$8,131.19	\$8,607.56
Steady-State w/ PMV Setpoints	\$299.21	\$4,383.28	\$1,226.63	\$7,426.72	\$7,725.93
MPC w/ PMV Setpoints	\$15.05	\$4,310.81	\$1,351.90	\$7,449.47	\$7,464.52

2.3.11 Random Occupancy Simulation

Newer building energy management systems will include occupancy sensors in rooms that allow for setbacks to occur when the room is unoccupied. The next simulation that was completed included a random occupancy profile of the zones in the UBO to simulate having occupancy sensors. The setback temperatures used for the zones were the same as those currently used in the UBO (29.4 °C). The zone temperatures from the steady-state and MPC simulations are shown in Figures 2.33 and 2.34. Again, the two control methods show similar results, with the MPC controller implementing pre-cooling. For the MPC controller, perfect knowledge of the future outdoor air temperature, outdoor air relative humidity, and occupancy of the zones are assumed. In application, the outdoor air temperature and relative humidity could be estimated from weather forecasts and the future zone occupancy could be estimated from employee scheduling.

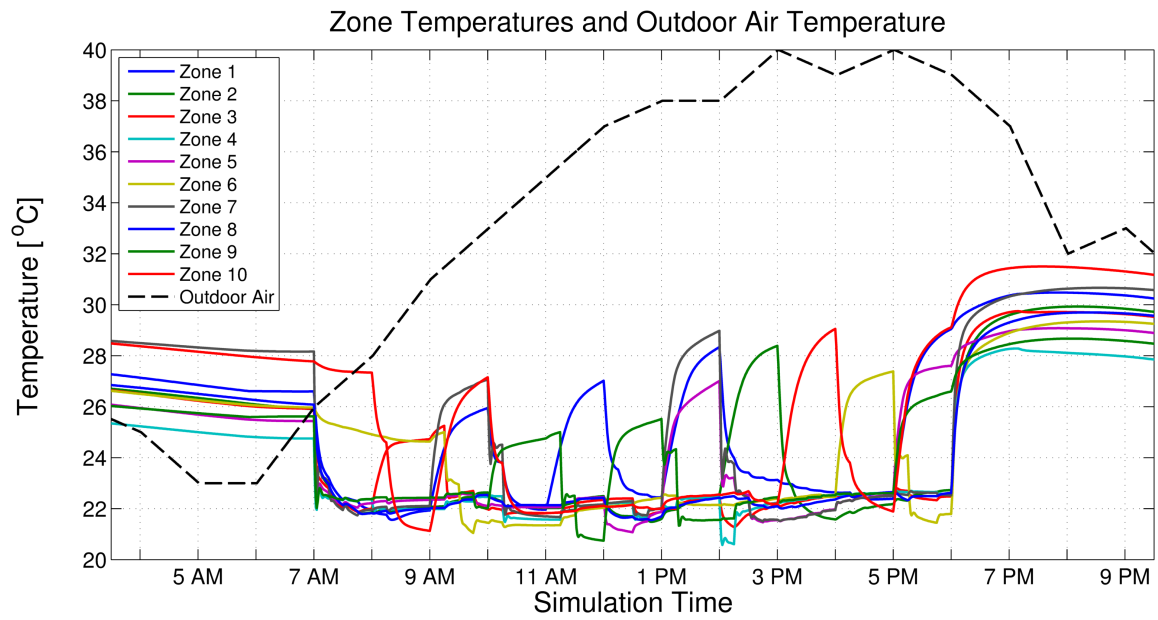


Figure 2.33: Zone temperatures for the steady-state method with random occupancy.

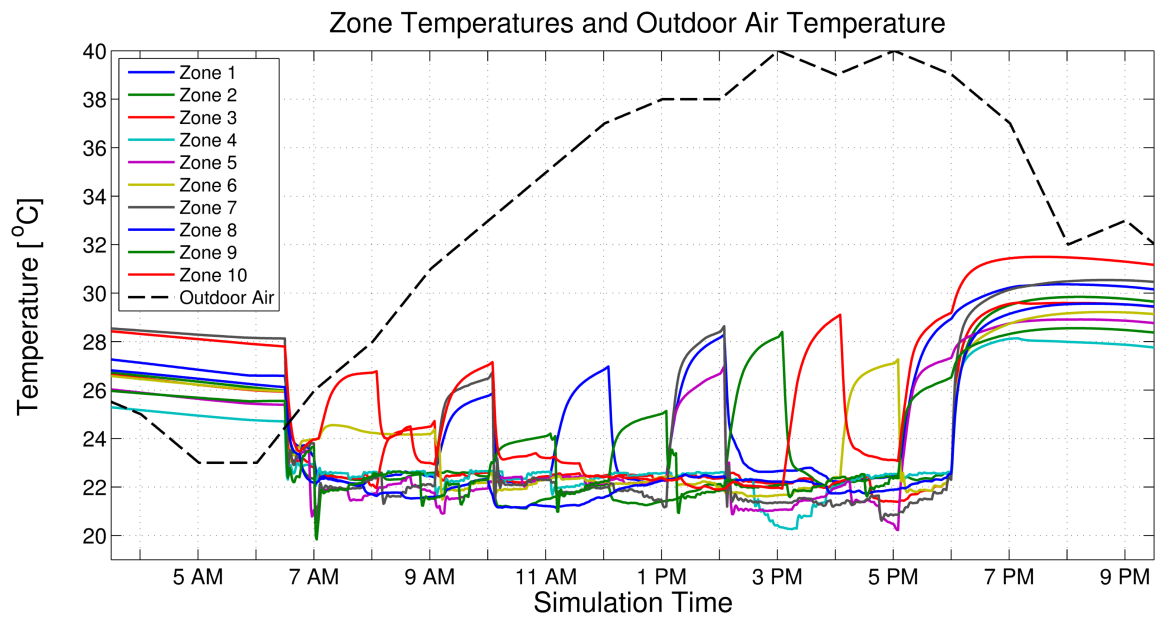


Figure 2.34: Zone temperatures for the MPC controller with random occupancy.

The economic cost results for the random occupancy simulations with the currently used setback temperatures are listed in Table 2.5. The steady-state method did slightly worse than the current control in overall zone comfort cost but managed to reduce both the total utility cost and the combined comfort and utility costs by 7.2% and 4.4%, respectively. These savings were mainly due to decreased electricity use by the fan. The MPC controller was able to improve on the overall cost with a reduction of 7.1% while giving 5.3% savings in utilities used from improved comfort and decreased fan use.

Table 2.5: Annualized economic costs for random occupation profiles and standard setback temperatures.

	Zone			Total	
	Comfort Cost	CHW Cost	Fan Cost	Utility Cost	Total Cost
Current Ctrl. w/ PMV Setpoints	\$1,598.30	\$4,051.93	\$1,699.75	\$7,431.14	\$9,029.44
Steady-State w/ PMV Setpoints	\$1,729.78	\$4,045.36	\$1,177.67	\$6,899.78	\$8,629.56
MPC w/ PMV Setpoints	\$1,354.39	\$4,056.91	\$1,296.88	\$7,035.31	\$8,389.71

2.3.12 Random Occupancy with Optimal Setbacks Simulation

Simulations were completed in which the steady-state and MPC methods were allowed to determine the optimal setback temperatures as opposed to using the setback temperatures currently in place in the UBO. The zone temperatures from both control methods are shown in Figures 2.35 and 2.36. Compared to the zone temperature profiles with the standard setbacks, the use of optimal setbacks resulted in zone temperature deviating less from the optimal comfort zone between 0 and -0.5 PMV for zero loss of productivity.

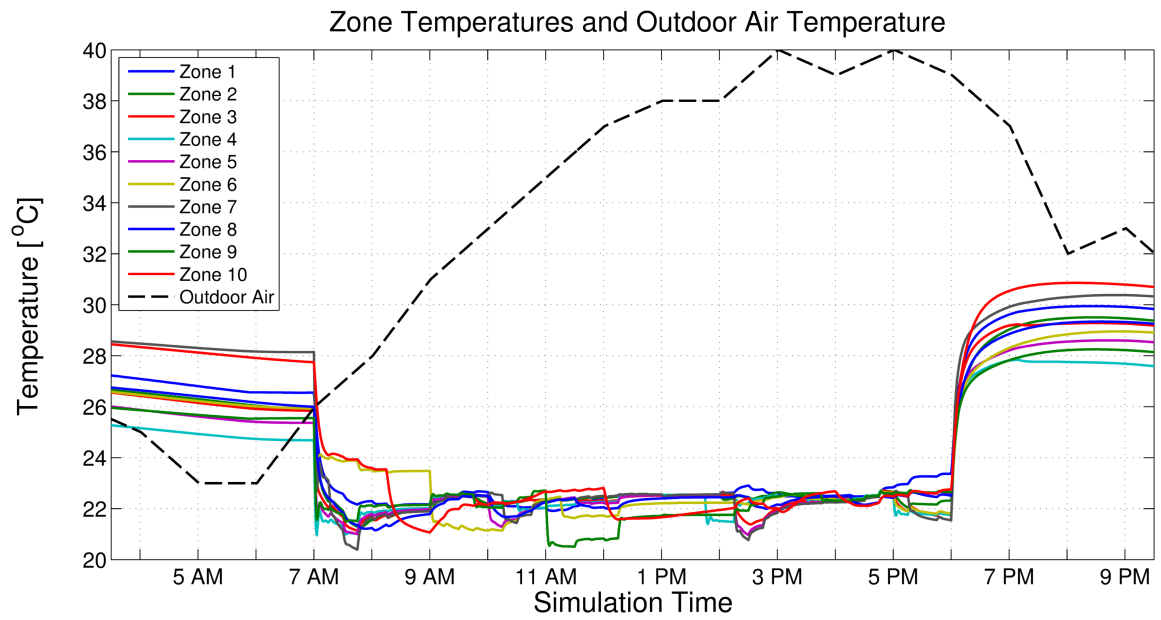


Figure 2.35: Zone temperatures for the steady-state method with random occupancy and optimal zone setbacks.

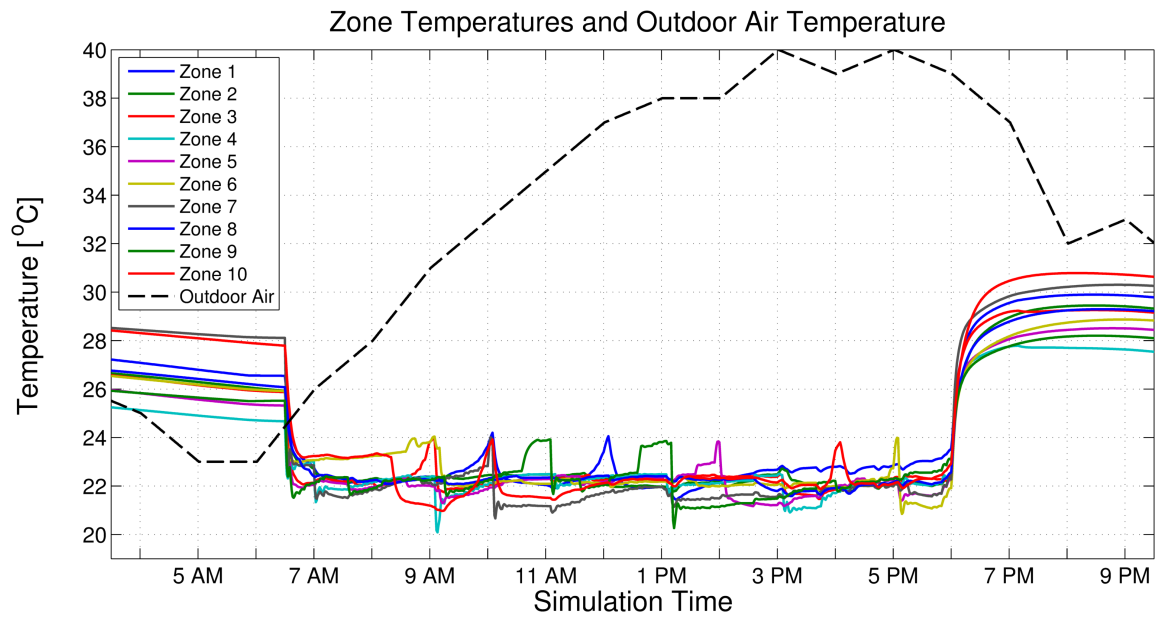


Figure 2.36: Zone temperatures for the MPC controller with random occupancy and optimal zone setbacks.

The economic results of the simulations with random occupancy and optimal setback temperatures are detailed in Table 2.6. Compared to the baseline of the current control using standard setbacks, both the steady-state and MPC methods provide performance improvements. While the decrease in annual utility costs are only \$187.49 (2.5%) and \$135.23 (1.8%) for the steady-state and MPC methods, respectively, the methods are much more efficient with the resources that they use. The steady-state was able to reduce the cost of loss productivity by \$1,297.50 for a combined decrease in discomfort and utility costs of 16.5%. The MPC method reduced the cost of loss productivity by \$1,416.32 to give a combined decrease in total costs of 17.2%. Both algorithms are able to provide significant productivity savings while decreasing the overall utility cost by using slightly more chilled water and less fan power than the current control method. Additionally, the added capability of determining optimal setback temperatures eliminates the need for building technicians to determining and adjusting setbacks, which can be a significant time investment as the number of zones and equipment grows.

Table 2.6: Annualized economic costs for random occupation profiles and optimal setback temperatures.

	Zone			Total	
	Comfort Cost	CHW Cost	Fan Cost	Utility Cost	Total Cost
Steady-State w/ PMV Setpoints	\$300.79	\$4,320.13	\$1,132.90	\$7,243.65	\$7,544.45
MPC w/ PMV Setpoints	\$181.97	\$4,276.29	\$1,247.17	\$7,295.91	\$7,477.89

2.3.13 Priority Zone Case

Sometimes it may be desirable to prioritize some zones over others with regard to maintaining comfort. It may be that certain zones are not occupied for extended periods of time, so slightly more uncomfortable conditions may be permissible to reduce utility costs. It could be that one zone is the office or work space of a very important person (VIP) that the building operators wish to ensure stay comfortable; conversely, the value of some occupants productivity and comfort may be less than others, such that the savings in utilities are of more importance to the business or building operator. A zone may require specific comfort conditions to be maintained for a certain process or piece of equipment. In any case, the ability to prioritize certain zones over others, either permanently or temporarily, is a desirable feature in building controls. Both the steady-state and MPC methods are capable of doing this through adjusting the annual salary assigned to each zone. This value can essentially act as weight to allow the building operator to prioritize certain zones over others. Simulations were performed in the extreme case where the other zones annual salaries are a factor of 1,000 less than Zone 5. The large magnitude was chosen to produce an easily identifiable visual difference to demonstrate the proposed control strategies' capabilities. The steady-state results can be seen in Figures 2.37 and 2.38. Zone 5 is maintained at a lower temperature than the other zones, with several zone temperatures increasing in the afternoon. Figure 2.38 shows that Zone 5's temperature is maintained between the zero LOP thresholds of -0.5 to 0 PMV. The very similar results from the MPC method are shown in Figure 2.39 and 2.40. Several zone temperatures can be seen rising away from their 0 PMV temperature threshold beginning around 12 PM. As the cooling demand of these zones increases with the outdoor air temperature and solar loads, their temperatures continue to increase; however, Zone 5 is again maintained in the zero loss of productivity range of -0.5 to 0 PMV, as shown in Figure 2.40. The discomfort costs

between the two methods were essentially equal, but the MPC method was able to reduce annualized utility costs by 8.2% from the utility costs of the steady-state method.

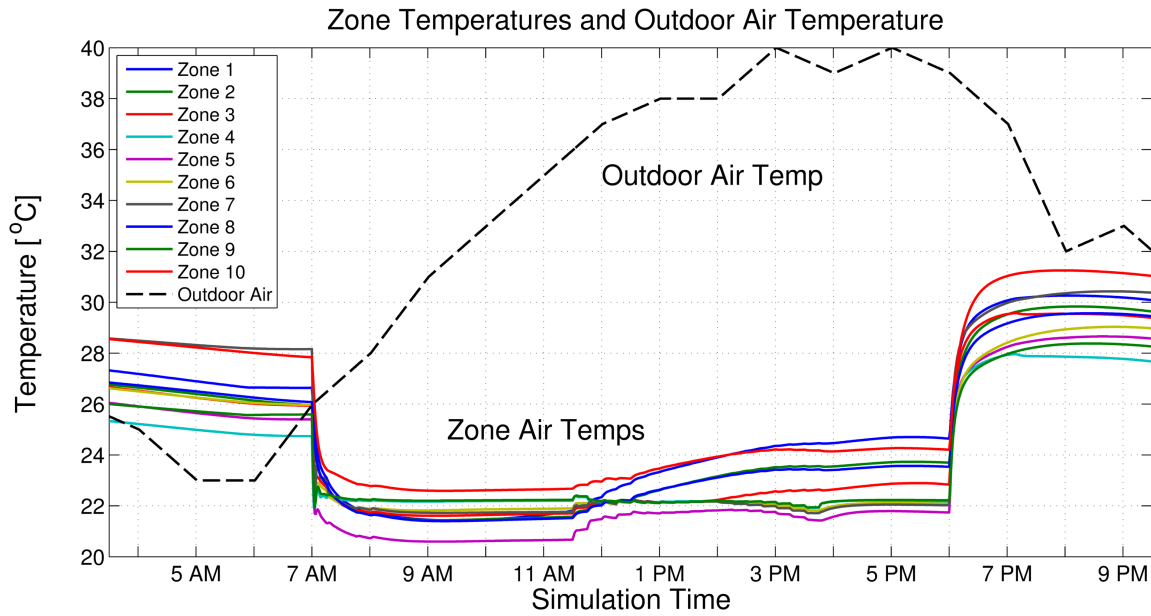


Figure 2.37: Zone temperatures for the steady-state method with Zone 5 prioritized over other zones.

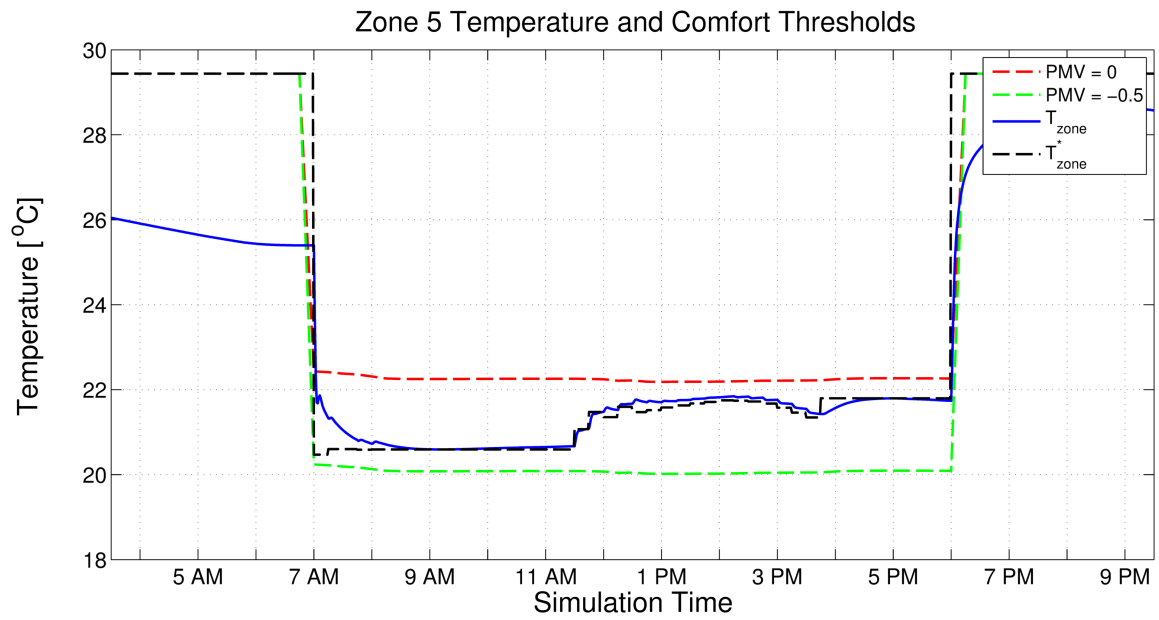


Figure 2.38: Zone 5 temperature with comfort limits as a prioritized zone using the steady-state method.

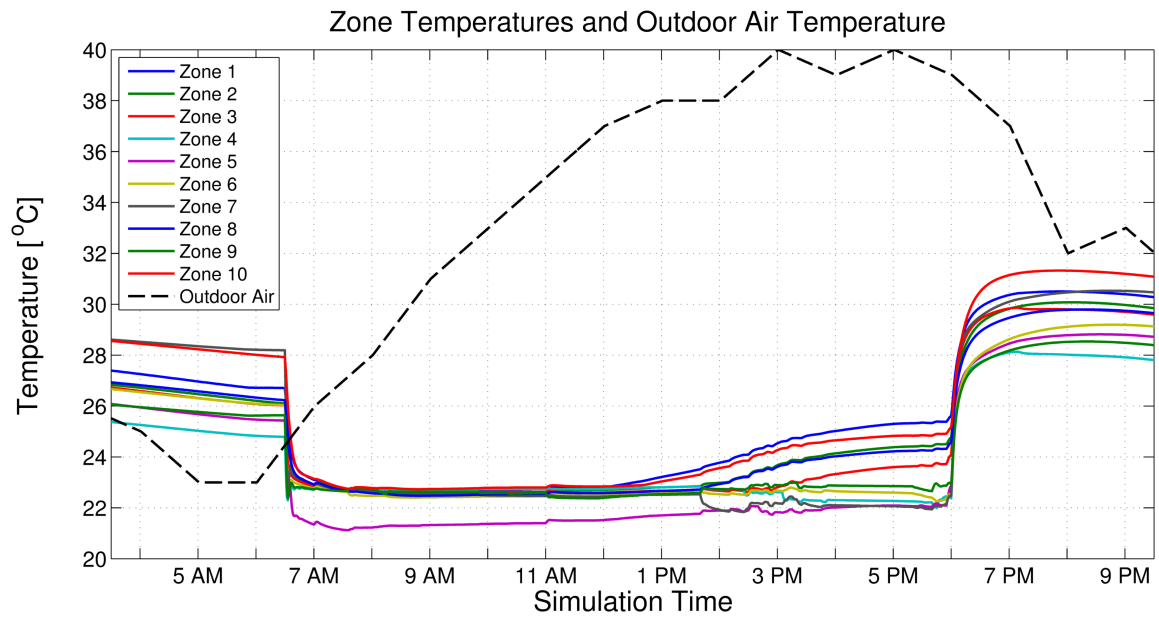


Figure 2.39: Zone temperatures for the MPC method with Zone 5 prioritized over other zones.

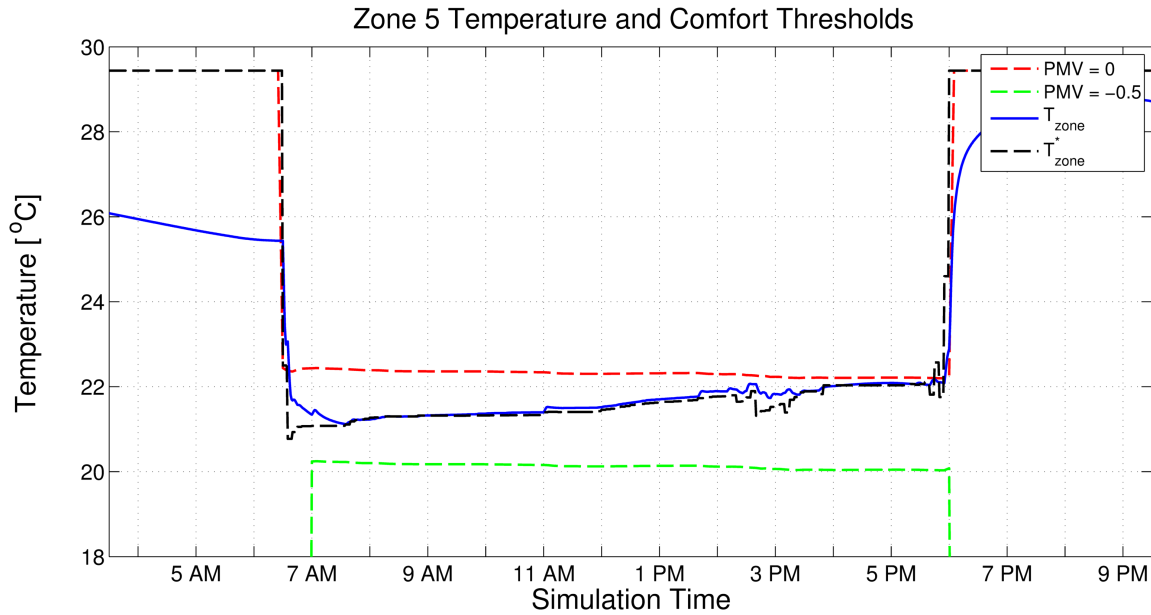


Figure 2.40: Zone 5 temperature with comfort limits as a prioritized zone using the MPC method.

2.4 Overall Comparison

Table 2.7 presents the overall comparison between the steady-state and MPC methods with the current control method using LOP optimal temperature setpoints ($PMV = -0.25$). Based on the results in the table, both the steady-state and MPC methods outperformed the current control methods with regard to total cost savings. The new methods showed the best performance increase (16.5% and 17.2%) when random occupancy profiles were present and the methods were able to determine the optimal setback temperatures. In the case of random occupancy profiles and standard setbacks, the steady-state algorithm had higher costs due to loss productivity than the current control method, but was able to use less chilled water and fan power to produce greater overall savings.

Comparing the steady-state performance to the MPC performance shows that, depending on the operational case, a full MPC implementation gives you marginal improvement.

Table 2.7: Summary of results of the simulations.

Method	Setbacks	Occupancy	Comfort Savings	Utility Savings	Total Savings
Steady-State	N/A	Constant	37.19%	8.66%	10.24%
MPC	N/A	Constant	96.84%	8.38%	13.28%
Steady-State	Current	Random	-8.23%	7.15%	4.43%
MPC	Current	Random	15.26%	5.33%	7.08%
Steady-State	Optimal	Random	81.18%	2.52%	16.45%
MPC	Optimal	Random	88.61%	1.82%	17.18%

Specifically, in the case of constant occupancy the steady-state method was able to achieve approximately 77.1% of the performance of MPC method. With random occupancy and optimal setback temperatures, the difference between the two methods was even smaller with the steady-state achieving 95.8% of the performance of the MPC. This comparison is insightful given that a fully dynamic MPC solution becomes increasingly more complex to implement as systems scale, along with an increase in computational burden and time. Essentially, in control of building energy systems, a steady-state prediction method can provide nearly as well as a full MPC solution with less complexity and computational requirements.

3. PAPER A: SIMULATION AND VALIDATION OF INTERIOR AND EXTERIOR NAVIGATIONAL STRATEGIES FOR AUTONOMOUS ROBOTIC ASSESSMENTS OF ENERGY¹

3.1 Synopsis

Approximately 40% of all US energy usage and carbon emissions is attributed to buildings. Energy audits of buildings are an effective way to identify significant energy savings, but the extensive training of auditors and cost of the audits result in only a very small fraction of buildings receiving an audit. Automation of the audit process using robots can offer more detailed information for better recommendations and greatly reduce the cost of audits. This paper discusses navigational strategies that would be used by ground and aerial robots as they conduct automated energy audits. The strategies are split into two scenarios: interior and exterior navigation. Simulations for both the interior and exterior navigational algorithms are presented. Lastly, a ground robot platform is developed to validate the navigational strategies.

3.2 Introduction

Significant cost savings and lower carbon emissions can potentially be realized through the reduction of commercial and industrial building energy usage. Buildings consume a large amount of all energy produced in the U.S. at approximately 40% [15]. As for U.S. carbon emissions, buildings are also responsible for approximately 40% [16]. Of those

¹© 2015 IEEE. Reprinted, with permission, from: Bay, CJ, Terrill, TJ, and Rasmussen, BP, "Simulation and validation of interior and exterior navigational strategies for autonomous robotic assessments of energy," in *American Control Conference (ACC)*, July, 2015. In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Texas A&M University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

percentages, commercial buildings makeup roughly half (20%) of the building energy usage and carbon emissions [2, 17].

While seemingly unchanging, buildings are very dynamic structures that evolve over varying time-scales. Construction materials degrade over time, resulting in reduced insulation and costly drafts. Building equipment degrades over time, becoming less efficient and costing more to operate. The weather changes with seasons and climates can change over years, giving different operating conditions than originally planned for. Energy audits are a mechanism to help cope with these changes. During audits, trained individuals travel to locations and perform assessments of the buildings, identifying cost saving measures that can reduce waste, save energy, and increase productivity. The Department of Energy (DOE) has provided funding for several Industrial Assessment Centers (IACs) throughout the country which, on average, find \$55,000 in annual energy savings for small- and medium-sized manufacturers [94]. The IACs provide these audits free of cost, as many businesses don't deem it financially feasible otherwise. This is due to the fact that audits are not cheap as they are labor intensive and require extensive training of the auditors over several years. Additionally, from auditor to auditor, the quality of recommendations can vary greatly based on differences in experience and expertise.

Autonomous robotic assessments of energy (AuRAE) have been previously proposed [5] to reduce costs and improve effectiveness of energy audits. These AuRAEs will be performed by an autonomous fleet of ground and aerial vehicles, collecting data and measurements as they navigate through and around buildings. The AuRAEs have many benefits over traditional energy audits including higher data resolution, more consistent recommendations, and lower costs through reduced man-hours and training required for audits. This paper serves to present and validate navigational strategies for these vehicles. During these audits, the robots will need to autonomously navigate both the interiors and exteriors of buildings. For the interior of a building, robots will be required to fully explore an

unknown environment to ensure adequate measurements and successful detection of all target features (lights, air ducts, etc.). On the exterior of a building, robots will need to follow the perimeter, maintaining a certain distance from the building in order to correctly identify and assess all required features (building envelope, fenestration, etc.). Due to the differing nature of the problems associated with interior and exterior building navigation, two separate methods were developed for each situation. SRT-Target was previously developed for interior navigation in [5]. An instant goals approach for exterior navigation was developed and is presented for the first time in this paper. Both algorithms were simulated in MATLAB on an arbitrary building shape. Lastly, a ground robot platform was developed to validate each navigational strategy.

The rest of the paper is organized as follows. Section II discusses other navigational strategies and details the selected algorithms for interior and exterior navigation. In Section III, simulation of the algorithms is presented with several cases demonstrating the exploration capabilities of the robot in both interior and exterior scenarios. Section IV then describes the ground robot platform that was developed for validating the navigational algorithms and the preliminary work completed. Section V discusses future efforts for the project, and Section VI provides conclusions for the work.

3.3 Navigational Strategies

Navigational strategies and path-planning for autonomous robots have been researched for quite some time. Many algorithms are available with a recent focus on using rapidly-exploring random trees (RRTs), originally proposed by LaValle in [73]. In this first paper, the RRT algorithm is described as starting at an initial point and growing “branches” in random directions whose selection is biased by their Voronoi diagrams. Essentially, branches are biased to grow towards regions that do not currently contain nodes. With enough iterations, the branches fill the space and there will exist a path from the starting point to any

space (within the resolution of the tree). Later, LaValle built upon this work to propose RRT-Connect [74] which incrementally builds two RRTs. One originates from the starting point and the other from the goal. They expand until they connect with one another. This of course works in a known environment with a known goal, but doesn't provide a solution for unknown environments.

A data structure based on RRT was proposed by Oriolo in [75], called Sensor-based Random Tree (SRT). As the name states, this method adapts the RRT strategy to the characteristics of typical hardware used in robotic sensing. The sensor perceives the surrounding and an associated safe region is determined. The tree is then incrementally built based on the generation of random points within this safe region. The resulting data structure represents a global map of safe regions which expands as the robot navigates. In [75], two exploration techniques are detailed: SRT-Ball and SRT-Star. Both methods assume 360° of perception while SRT-Ball's radius of perception is limited by the nearest sensed obstacle and SRT-Star's area of perception is divided up into conical areas with each cone extending as far as the nearest obstacle in the cones region. SRT-Star is more confident as it uses more of the information available while SRT-Ball provides more conservative navigation. With SRT-Star, additional computation is required in determining each cone from one another as perception and navigation takes place. A method that builds on SRT-Ball and SRT-Star, SRT-Radial, was proposed by Espinoza in [76]. In the improved SRT-Radial method, the perceived safe region contours to obstacles thus eliminating the need to determine various regions as in the SRT-Star method. In [76], 360° of perception is assumed, and with all the SRT strategies, it is assumed that perfect localization is provided by a SLAM method. The algorithm previously proposed in [5], SRT-Target, built on SRT-Radial by adding the ability to check for targets in real-time and navigate to those targets, if necessary. In autonomous energy audits, these targets could be lights, air ducts, or other items of interest. Furthermore, a calibration factor was introduced to facilitate the physical

visitation of the robot to all areas. This factor accounts for the different perception ranges of the sensors used in the target identification, encouraging the identification of all targets possible. SRT-Target was chosen as the method to be used on the AuRAE for interior navigation and exploration because of its ability to explore unknown environments and facilitation of discovering targets in different situations.

The previous methods work for navigating unknown environments that have a closed outer boundary; however, for performing various audit tasks on the exterior of a building, a different approach is required. Numerous boundary following procedures and techniques have been proposed over the years. Some have been based on fuzzy logic controllers [77], Lyapunov functions [78], or even the sensing mechanism of cockroaches [79]. Many use single or multiple ultrasonic sensors [81, 82, 80] and are meant to operate in known environments. One boundary following method [6] uses instant goals to provide path planning. Instant goals allows for a robot to follow arbitrarily shaped shapes of simple or complex geometries. Additionally, the robot can encounter disturbing obstacles and continue moving along the previous boundary, subject to the robots geometry being able to pass between the two obstacles. For these reasons, and the possibility of being able to circumnavigate buildings without a closed boundary (open bay doors, passageways, etc.) with some modification, an algorithm based on the instant goals method presented in [6] was developed for external navigation.

3.3.1 Interior Navigation

For interior navigation, the ground robot uses the SRT-Target algorithm, discussed in detail in [5]. The algorithm can be seen in Fig. 3.1. Briefly, the algorithm uses a data-tree structure as it incrementally builds a map of the environment as it explores. During each iteration the robot perceives its surroundings with the available depth sensor and generates the region S that gives the free and occupied space about the current location, q_{curr} . The

current location and S are then added to the global map T . The space $S \cdot \beta$ is added to map U for the purpose of identifying targets during an energy audit, for example, ceiling lights. The calibration factor β is to restrict the robots movement to ensure that it visits every possible space that a target could be. Without β , the robot could add sensed areas to

```

SRT_TARGET( $q_{init}, K_{max}, I_{max}, \alpha, d_{min}$ )
1   $q_{curr} = q_{init};$                                      # Initialize current position.
2  for  $k = 1$  to  $K_{max}$ 
3       $S \leftarrow \text{PERCEPTION}(q_{curr});$                  # The perceived environment is stored in S.
4       $\text{ADD}(T, U, (q_{curr}, S, \beta));$                    # Add the current location and S to the global
5                                                         # tree T and add  $S \cdot \beta$  to the tree U.
6       $i \leftarrow 0;$                                      # Initialize the count.
7      loop # Loop until a valid candidate is found or max iterations is reached.
8           $(target, q_{target}) \leftarrow \text{TARGET\_CHECK};$ 
9                                                         # Check for target at current location.
10         if ( $target = true$ )
11              $q_{cand} = q_{target};$                      # Make the candidate location the target location.
12             break;                                     # Stop searching for candidate.
13         else
14              $\theta_{rand} \leftarrow \text{RANDOM\_DIR};$          # Pick random direction in sense range.
15              $r \leftarrow \text{RAY}(S, \theta_{rand});$            # Find greatest distance  $r$  along
16                                                         #  $\theta_{rand}$  in the local safe region.
17              $q_{cand} \leftarrow \text{DISPLACE}(q_{curr}, \theta_{rand}, \alpha \cdot r);$ 
18                                                         # Calculate candidate location from
19                                                         # current location,  $r$ , and  $\theta_{rand}$ .
20              $i \leftarrow i + 1;$ 
21         end if
22     until ( $\text{VALID}(q_{cand}, d_{min}, U)$  or  $i = I_{max}$ )
23     if ( $\text{VALID}(q_{cand}, d_{min}, U)$ )
24          $\text{MOVE\_TO}(q_{cand});$                              # Move to candidate.
25          $q_{curr} \leftarrow q_{cand};$                      # Update current location.
26     else # If valid candidate can't be found move to parent location.
27          $\text{MOVE\_TO}(q_{curr}.\text{parent});$ 
28          $q_{curr} \leftarrow q_{curr}.\text{parent};$            # Update current location.
29     end if
30 return  $T;$  # The result is given in the global map T.

```

Figure 3.1: The SRT-Target algorithm, previously presented in [5].

the global map large enough that targets could be outside of the target-sensing hardware's range. The robot then begins searching for a candidate location to move to. If the robot is under a target, it will move towards the target, otherwise the robot searches for a random direction in which to proceed.

A valid candidate position is found if along the random direction the robot can travel a minimum distance and if the candidate position hasn't been previously visited. After a certain number of iterations, if the robot cannot find a valid candidate position, it will move to the previous location and continue the search. As the robot fully explores an area, it will retrace it's steps, eventually returning to it's original starting location.

3.3.2 Exterior Navigation

To move and map around the outside of a building, the method of using instant goals from [6] was selected. This method allows for the following of arbitrarily shaped obstacles in unknown environments. The algorithm used can be seen in Fig. 3.2. The robot begins by perceiving the initial surroundings, S , and determines the coordinates of the nearest point on the obstacle it senses, q_{wall} , relative to the robot. The robot then checks to see if it is the user-defined distance away from the obstacle by calculating the distance between itself and the wall. If the robot needs to move closer or further away from the obstacle, it performs this motion and the algorithm proceeds as normal. A portion of the perceived environment S is then split into sections, or slices, and the nearest obstacle measurement point from each slice is found and stored in D . This allows for the data to be stored in a vector format, taking up less memory and providing quick access/processing. Fig. 3.3 shows the perceived environment with obstacles split into sections. The number of slices can be defined with n_{slices} , depending on computation power and desired resolution. The portion of S to be stored is determined by the $srch_{from}$, $srch_{to}$, and $srch_{dir}$ parameters. The $srch_{from}$ parameter defines what heading to begin the slices at while the $srch_{to}$ parameter

defines where to stop. The $srch_{dir}$ parameter tells the algorithm which direction to move through the slices, left or right, which also determines if the robot will navigate the obstacle clockwise or counter-clockwise. This value can be predefined by the user or determined

```

EXT_NAV( $q_{init}, srch_{from}, srch_{to}, srch_{dir}, n_{slices}$ )
1   $q_{curr} = q_{init};$                                      # Initialize current position.
2   $S \leftarrow \text{PERCEPTION}(q_{curr});$                      # The perceived environment is stored in S.
3   $q_{wall} = \text{NEAR\_OBS}(q_{curr}, S);$                      # Find nearest point on obstacle.
4  if ( $\text{DIST}(q_{curr}, q_{wall}) \neq \perp_{dist}$ )
5       $q_{\perp} = \text{PERP}(q_{curr}, q_{wall}, \perp_{dist});$          # If the robot is not the user-defined
6       $\text{MOVE\_TO}(q_{\perp}); q_{curr} \leftarrow q_{\perp};$          # distance away from the wall, move to the
7       $S \leftarrow \text{PERCEPTION}(q_{curr});$                  # correct position and update  $S$  and  $q_{wall}$ .
8       $q_{wall} = \text{NEAR\_OBS}(q_{curr}, S);$ 
9  end if
10  $D \leftarrow \text{SLICES}(S, n_{slices}, srch_{from}, srch_{to}, srch_{dir});$ 
11                                     # Divide  $S$  into sections with obstacle info.
12  $q_{dist} = \text{MAX}(\text{DIST}(D, q_{wall}));$                  # Find the max. distance between the
13                                     # nearest wall point and the obstacles sensed in  $D$ .
14  $[targ_{dist}, targ_{head}] = \text{LINE}(q_{dist}, q_{wall});$ 
15  $q_{targ} = \text{TARGET}(q_{curr}, targ_{dist}, targ_{head});$      # Draw a line between the points
16                                     # and find the target coordinate.
17 if ( $targ_{dist} = 0$ )                                     # If only one obstacle is sensed use prev. motion.
18      $\text{MOVE}(q_{curr}, targ_{dist-prev}, targ_{head-prev});$ 
19 else
20     if ( $\text{DIST}(q_{targ}, q_{wall}) < \perp_{dist}$ )             # If  $q_{targ}$  is too close to the obstacle, revise
21          $[targ_{dist}, targ_{head}] = \text{LINE}(q_{targ}, q_{wall});$  #  $q_{targ}$  to be farther from obstacle.
22          $q_{targ} = \text{TARGET}(q_{wall}, \perp_{dist}, targ_{head});$ 
23          $\text{MOVE\_TO}(q_{targ});$ 
24     else
25          $\text{MOVE}(q_{curr}, targ_{dist}, targ_{head});$          # Move to target location.
26     end if
27 end if
28  $[targ_{dist-prev}, targ_{head-prev}] = [targ_{dist}, targ_{head}];$ 
29 while  $q_{curr} \neq \text{THRESH}(q_{init})$  do
30     Repeat lines 2-27;

```

Figure 3.2: The external navigation algorithm, based off of the use of instant goals described in [6] .

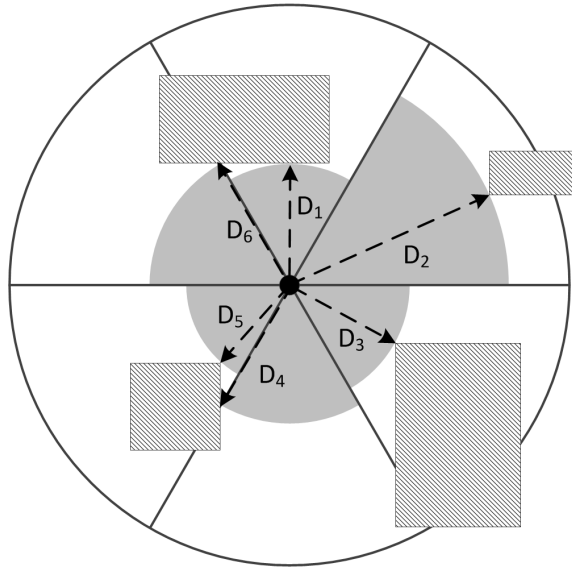


Figure 3.3: The perceived environment divided up into sections for the exterior navigation algorithm.

based on the perceived surroundings.

Distances between the nearest obstacle point and the other points within the slices are then determined and the maximum is found. The line between the nearest point and the point farthest away is calculated and the length of the line is used as the target distance to travel, $targ_{dist}$, and the slope of the line is used to determine the heading of the target location, $targ_{head}$. If only one obstacle distance measurement is available, such as near a corner, the robot will repeat the previous motion in order to reveal more of the obstacle and give more slices with obstacle measurements. Otherwise, the robot will move the target distance along the target heading along as the target coordinate does not move the robot within the use defined distance to remain from the obstacle. If the target coordinate is within this distance, than a new target coordinate is generated at the point that is on the line between the obstacle point and the old target coordinate with a distance equal to the required distance from the obstacle. An example of how the robot would navigate a corner

is shown in Fig. 3.4. This process is repeated until the robot moves within a threshold distance of the starting location. Currently this method assumes the robot is placed in a starting position within the sensor's range of an obstacle and that the building is a closed loop, i.e. no open doors.

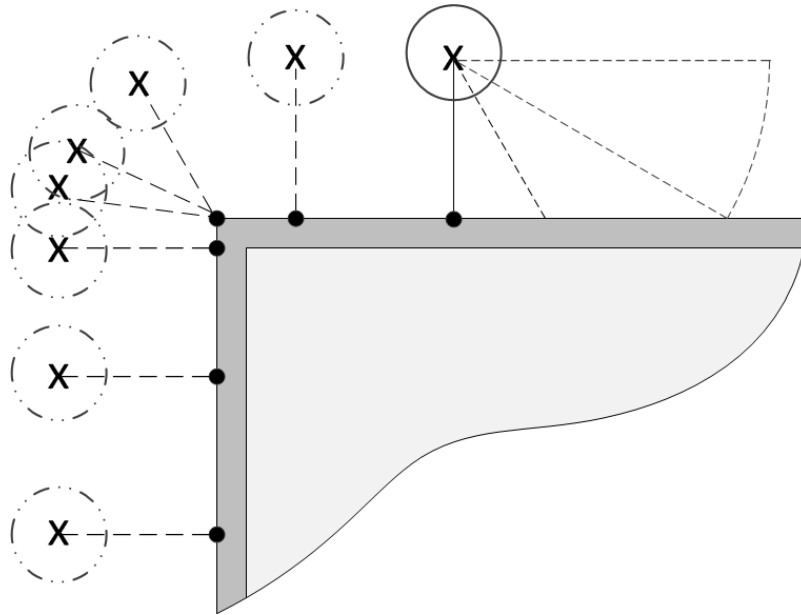


Figure 3.4: Exterior navigation around a corner.

3.4 Simulation Studies

Simulations of the interior and exterior navigation strategies were performed in MATLAB. An arbitrarily shaped building was created with different types of obstacles to demonstrate the capabilities of the algorithms. The simulations were started from several locations to show that the strategies are successful from various initial conditions.

3.4.1 Interior Navigation Simulation

Fig. 3.5 shows the evolution of the global map as the robot explores the interior of the simulation building. The robot, shown in blue, starts exploring the upper left room and then moves out into the more open area of the building. The tree-like nature of the path, shown in green, can be seen as the robot chooses random directions to explore in. On the right side of Fig. 3.5 the UAV has fully explored the area and returned to its starting location, per the nature of the algorithm.

Four of the completed simulations can be seen in Fig. 3.6. In each of the maps, the robot was started at different locations. Variance in the paths created by the algorithm can be observed. The simulations show full or nearly full exploration of the interior space, with only a little space left unexplored in different corners in the top-right and bottom-right

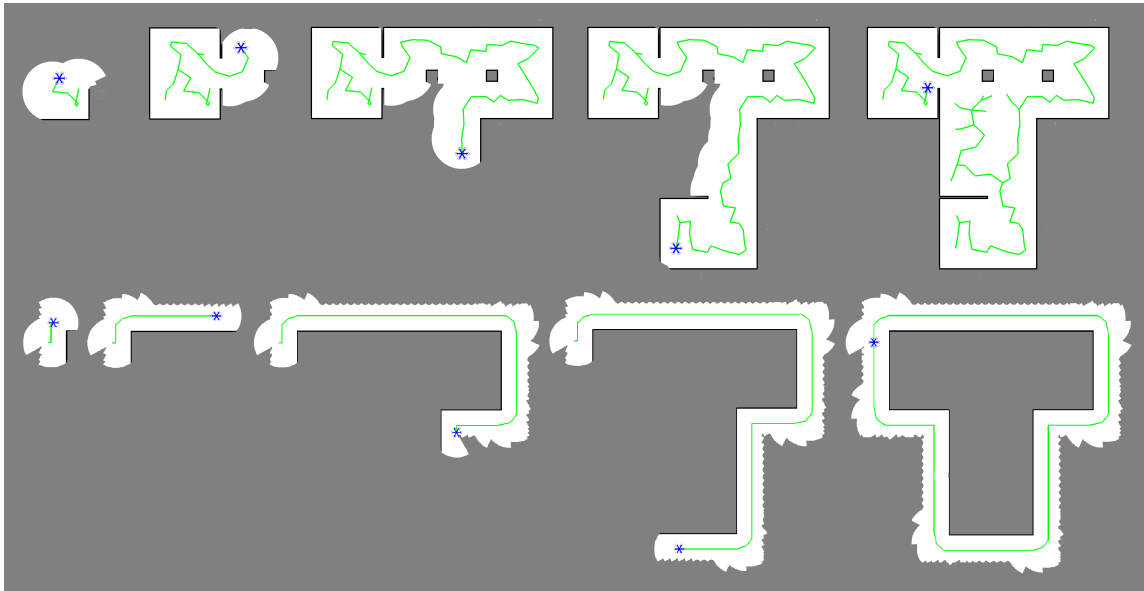


Figure 3.5: This figure shows the robot as it explores the interior and exterior of the building. The robot is shown in blue, the path of the robot in green, unexplored space in gray, obstacles in black, and free space in white. The tree-like nature of the interior navigational strategy can be seen in the top part of the figure.

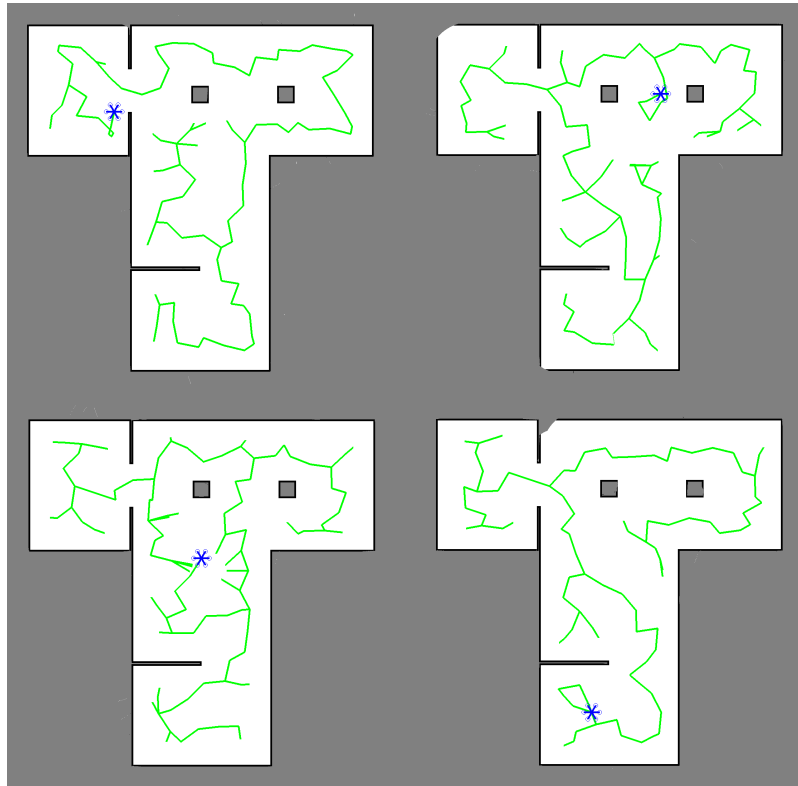


Figure 3.6: Interior navigation simulations showing full exploration from different starting locations.

map. Overall, the simulations showcase the success of the algorithm.

3.4.2 Exterior Navigation Simulation

The evolution of the external exploration path of the robot can be seen in Fig. 3.5. As in the interior simulations, the robot is shown in blue, the robot's path in green, unexplored space in gray, obstacles in black, and free space in white. It can be seen that the robot tracks the walls of the building well, navigating both convex and concave corners with the simulation ending once the robot has returned to its starting location.

Again, four simulations are shown in Fig. 3.7 for the exterior navigation algorithm with the robot starting at four random locations. Uniform paths can be observed, with

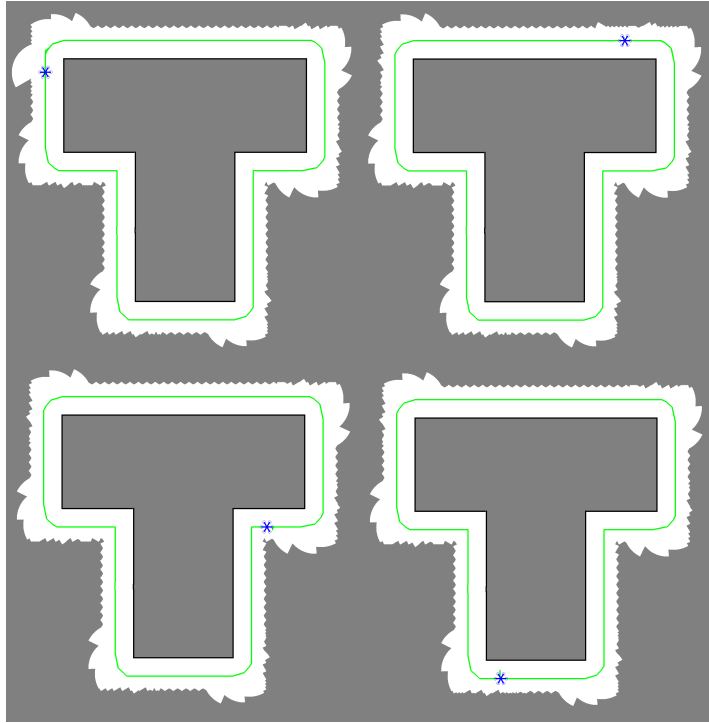


Figure 3.7: Exterior navigation simulations showing complete building navigation from different starting locations.

small variations at the corners due to the differing target distances as the robot approached the corners.

3.5 Algorithm Validation with Ground Robot Platform

AuRAE will use both aerial and ground vehicles to perform audits. In this preliminary work a ground robot platform was developed to demonstrate the navigational strategies. A ground robot was used to allow for ease of testing before implementing the algorithms on an UAV. The hardware and software used are described in the following section.

3.5.1 Ground Robot Platform

The ground robot platform (see Fig. 3.8) consists of a differential drive chassis with two castor wheels, allowing for motion in all directions. Each wheel is powered by a

geared DC motor with attached encoder, which gives the possibility of calculating odometry from wheel rotations. The primary computing source of the ground robot is a laptop with a quad-core, 2.1 GHz processor. A microprocessor is used to interface with the motor controllers and encoders. The core software of the system is ROS (Robot Operating System [95]). Several ROS packages are being used to facilitate mapping and navigation. ROS-arduino-bridge provides communication between ROS and the microcontroller, motors, and encoders. Hector mapping is being used along with a scanning LASER range finder (Hokuyo URG-04LX) to accomplish SLAM (Simultaneous Location and Mapping) while the ROS Navigation stack gives access to the maps generated by hector mapping for path planning. The interior and exterior navigational algorithms were coded as plugins to the ROS Navigation stack to supply the local and global navigational goals. Lastly, a 9 degree-of-freedom IMU is employed to provide additional odometry information. An



Figure 3.8: The ground robot platform used in experiment.

extended Kalman filter is used by the ROS package robot-pose-ekf to fuse the three odometry information sources: pose estimation from hector mapping, odometry calculations from the encoders, and odometry calculations from the IMU.

3.5.2 Ongoing Efforts

Work is ongoing to extend the algorithm to accommodate the eccentricities that are common with exterior building envelopes. Fig. 3.9 shows a building on the campus of Texas A&M where preliminary validation of the exterior navigational algorithm is being done. Pillars and trees near the building offer navigational challenges. Eventually the robot will have additional sensors, such as a thermal camera, for identifying building fenestration and insulation faults with windows and doors.

Interior navigation validation is being pursued as well, with promising initial results. Fig. 3.10 shows a successful mapping of several rooms in a small building. All doors were open prior to the experiment. The final platform will be able to locate and identify lights, take several point measurements (temperature, humidity, CO₂), and enumerate building occupants.



Figure 3.9: Exterior navigation around a building on Texas A&M's campus.

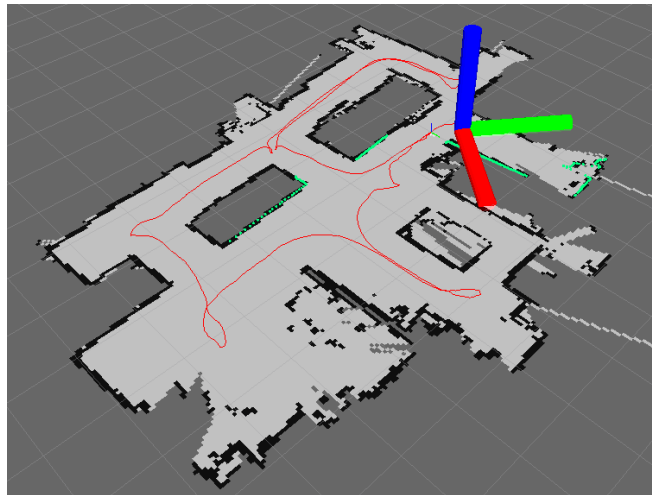


Figure 3.10: Map developed by ground robot platform for interior navigation.

3.6 Future Work

Modification of the exterior navigation algorithm to allow for open areas and doors is left to be completed. Experimental validation in a real-world environment is next, followed by the integration of a lighting audit package described in [96] to begin testing efficacy of automated lighting audit recommendations. Also, implementation of the navigational strategies on an UAV will be completed in the near future.

3.7 Conclusion

In this paper navigational algorithms for use by ground and aerial vehicles during autonomous robotic assessments of energy (AuRAE) were discussed. These navigational algorithms are needed by robots if the cost of energy audits are to be reduced. SRT-Target was selected for interior navigation and an instant goal approach was developed for exterior navigation. The algorithms were then simulated on an arbitrary building shape with the presence of random obstacles. The simulations showed successful performance of the algorithms for both interior and exterior navigation. Finally, a ground robot platform was developed to demonstrate the navigational strategies in a laboratory environment. The algorithms presented in this paper are just one step of many in the development of AuRAE. While developed for performing autonomous energy audits, these algorithms are also applicable in other cases where target identification in unknown environments or boundary following is required.

4. PAPER B: AUTONOMOUS LIGHTING ASSESSMENTS IN BUILDINGS: PART 1 - ROBOTIC NAVIGATION AND MAPPING¹

4.1 Synopsis

Approximately 40% of all US energy usage and carbon emissions are attributed to buildings. Energy audits of buildings are an effective way to identify significant energy savings, but the extensive training required by auditors and cost of the labor intensive audits result in only a small fraction of buildings receiving an audit. Automation of the audit process using robots can offer more detailed information for better recommendations, greater consistency in analysis and recommendations, and greatly reduce the cost of audits. This paper introduces such a system and proposes navigational strategies that would be used by ground and aerial robots as they conduct automated energy audits. The strategies are divided into the interior and exterior environments. Simulations for both the interior and exterior navigational algorithms are presented, showing success in completely exploring previously unknown areas, identifying and maneuvering to objects of specific interest to energy audits, and circumnavigating open exterior perimeters of buildings.

4.2 Introduction

Buildings, as they currently are, and the way in which buildings consume energy have immense potential for significant monetary savings and reduction of carbon emissions. According to the Department of Energy (DOE), buildings accounts for approximately 40% of all U.S. energy usage and carbon emissions [15, 16]. In 2010 buildings accounted for 82.2%, or \$301.6 billion, of the total U.S. electricity expenditures [2]. Commercial buildings represent roughly half of the building energy usage and carbon emissions [17,

¹This is the authors accepted manuscript of an article published as the version of record in *Advances in Building Energy Research* © 06 Oct 2016, pp 1-22, DOI:10.1080/17512549.2016.1237377. <http://www.tandfonline.com/doi/full/10.1080/17512549.2016.1237377>

2]; additionally, with an average energy expenditure greater than \$2 per square foot (ft²) in commercial and government buildings, the argument for savings becomes even more apparent [2].

One method to attain energy savings in buildings is to implement and follow the recommendations given by an energy audit. Buildings, although seemingly static, are dynamic environments that change over several time-scales. HVAC and lighting equipment degrade through use, resulting in efficiency losses. Building insulation and window seals deteriorate with age and shifts in the building structure. Environmental conditions change with the weather and the seasons, altering the optimal operating set-points of equipment which are often set upon installation and never updated. During an energy audit, a trained individual or team identifies opportunities within a building to increase productivity, reduce waste, and save energy. The different levels of an energy audit can be seen in Figure 4.1, as defined by the American Society of Heating, Refrigeration and Air-conditioning Engineers (ASHRAE) [7]. First, preliminary energy-use analysis is performed by examining utility bills and other energy-use documents to allow for comparisons to similar entities. Next, a level 1 audit involves sending auditors to a site to perform a high-level walk-through of the facilities, visually identifying energy saving measures and taking few measurements. A level 2 audit is more involved, requiring several auditors to gather numerous measurements and complete in-depth analysis of the energy used, taking several days depending on the size of the audit. Finally, a level 3 audit requires extensive measurements (environmental and structural) so that energy simulations of the building can be conducted, producing the most tailored and all-encompassing set of possible recommendations. A level 3 audit can last up to several weeks.

While a level 3 audit provides the most accurate and in-depth recommendations, energy savings resulting from even basic Level 1 and 2 audits can be significant. For example, a DOE funded program that offers one day audits for small to medium sized manufacturers

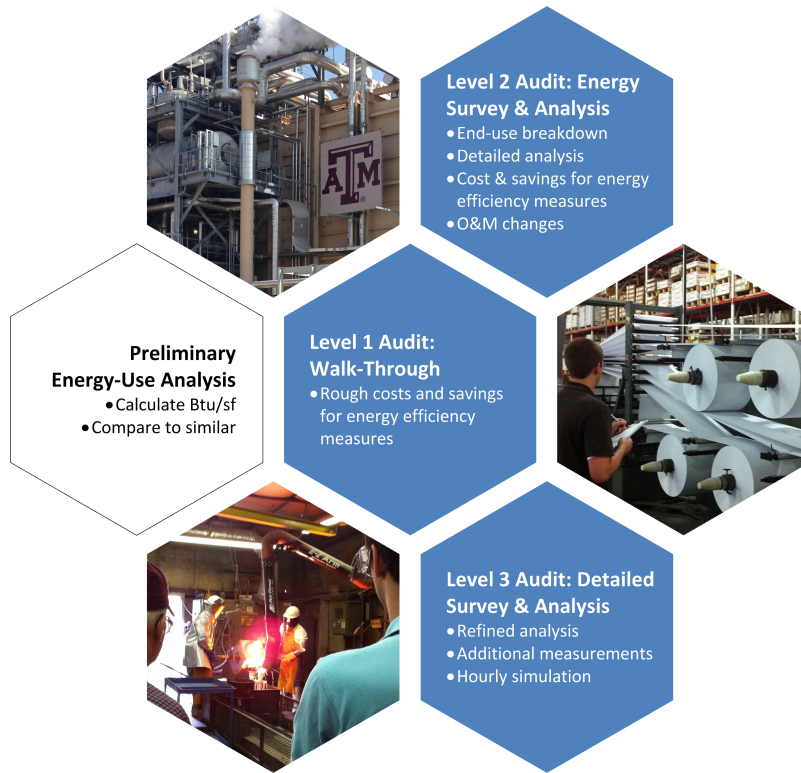


Figure 4.1: Levels of energy audits, as defined by the American Society of Heating, Refrigeration and Air-Conditioning Engineers adapted from [7].

has resulted in an average of approximately 17% annual savings per client [18]. These audits are freely provided, as frequently the cost deters businesses from having an audit performed. This cost is derived from the labor intensive nature of energy audits as well as the extensive and lengthy training required for the auditors. As the manpower and time required for each level of audit increases, so does the cost. Additionally, the quality of the audit and recommendations given can vary greatly based on the experience of the auditor(s).

Thus, to reduce the cost and improve on the effectiveness of energy audits, the authors have proposed autonomous robotic assessments of energy (AuRAE) [5]. These AuRAEs will be performed by an semi-autonomous fleet of ground and aerial vehicles, collecting

data and measurements as they navigate through and around buildings. The data and measurements include temperature, relative humidity, human occupancy, CO₂ levels, lighting levels, light locations & types, missing/faulty insulation (both in walls and around windows), thermal reflectivity of the exterior, and overall dimensions of the building (interior & exterior). From this data, many of the typical recommendations for energy and cost savings can be made. Additionally, 3-dimensional models can be automatically generated in such a format that they can be imported into energy simulation software so that buildings with varying conditions and equipment can be simulated. The combination of these will result in an in-depth report given to the building owner/operator with recommendations for energy savings.

Table 4.1 shows a breakdown of the capital investment and per audit cost between traditional auditors and the proposed AuRAE system. The main cost difference comes from the capital costs required for each. An auditor has to undergo numerous training courses, after which they must sit for a certification class, both of which incur significant cost. On top of the training and class, in order to be certified an auditor must have a minimum of 3 years of experience. An average salary of \$70,000 was used in these calculations after discussions with several auditors and auditing sources. For AuRAE, the capital investment cost is significantly cheaper. For a set of 5 robotic platforms along with technician training estimates, the total is \$22,000, vs. the \$438,000 for traditional auditors. Furthermore, the per audit cost is lower as fewer trained personnel are needed. The per audit cost shown in Table 4.1 is for a typical level 1 audit. With more advanced audits, the traditional method cost would significantly increase, while the AuRAE method would only see a slight increase, mainly due to the labor time saved by automating the measurement procedures.

Through automating many of these tasks, several unique challenges arise. When navigating in a building environment, robots are not able to utilize current positioning technologies, such as the global positioning system (GPS). In addition to navigating in a GPS-

Table 4.1: Breakdown of costs between traditional auditors and proposed AuRAE system.

Auditors		AuRAE	
Training courses	\$15,000	UAV (x5)	\$6,000
Certification class	\$3,000	Technician training	\$1,500
Minimum 3 years experience (pay x2)	\$420,000	Sensors (x5)	\$15,000
Capital investments	\$438,000		\$22,000
Travel (2 auditors)	\$1,500	Travel (1 technician)	\$750
Cost of audit (20 hrs at \$60/hr)	\$2,400	Cost of audit (20 hrs at \$60/hr)	\$1,200
Cost per audit	\$3,900		\$1,950

denied environment, supplying the robots with an accurate map before the audit begins is infeasible; therefore, the robot must fully explore an unknown building environment autonomously, tracking its location as it travels. To complicate matters further, some recommendations require knowledge about specific objects. Thus, the robots must be able to identify these objects of interest, alter its exploration path, navigate to the targets, and make the required measurements and observations. These difficulties experienced while navigating the inside of a building require an integrated mapping, localization, exploration, and target identification solution.

While technical in nature, the benefits of an AuRAE over a traditional audit are numerous. Data gathered continuously as the robots travel through the building provide greater data resolution and detailed maps of the environment versus single point measurements taken by an auditor. These measurements can be taken much more quickly, recording different types of readings at the same time resulting in reduced audit times. Additionally, the robots will give more consistent results without the variation of experience that exists from person to person. Operators of these teams of robots will require less training than that

of current auditors. These AuRAEs could be performed once, or be reoccurring with improvements in data and sensing from previous assessments. The fleets could be marketed as purchasable sets, allowing for continuous use by building managers and technicians. If a fleet was purchased for a building, it could interface with the building's energy management system. With this level of synthesis, audits could result in day-to-day energy efficient operation and comfort levels for the occupants.

Initially, these teams of robots will augment the manual audit process, eliminating simple, time-consuming tasks and providing additional depth of analysis not possible with a human auditor. The robots will operate semi-autonomously, capable of following an auditor in difficult navigational situations (closed doors), or exploring autonomously, freeing up the technician to monitor other processes or duties. Aerial robots will work together to produce maps of the environment and take measurements, while ground robots will serve as base stations, relaying data and providing additional power as needed. Upon completion of this research, entirely automated audits performed by these robots are envisioned. Ultimately, the successful implementation of AuRAE will increase the number of buildings receiving an energy audit that, in today's operating climate, would not normally receive an audit. This translates into significant energy savings reducing the demand on current power grids and energy sources as well as decreasing businesses operating costs and raising profitability.

The purpose of this paper, the first in a series, is two-fold: 1) to explain the automation of the energy audit process through the use of AuRAE, beginning with unmanned vehicles (UVs) and lighting, and 2) present building- and audit-centric navigational strategies to explore and map unknown environments, both on the interior and exterior of a building. For the interior of a building, robots will be required to fully explore an unknown, GPS-denied environment to ensure adequate measurements and successful detection of all targets of interest (lights, air ducts, etc.). The robots must also alter their exploration plans to navi-

gate to targets and integrate the data collection required to complete the audit tasks. On the exterior of a building, robots will need to follow the perimeter, maintaining a proscribed distance from the building in order to correctly identify and assess all required features (building envelope, fenestration, etc.). Furthermore, there often exist large doors that are open majority of the time. As such, the exterior navigational strategy needs to be able to detect these gaps in the perimeter, and traverse them accordingly. Due to the differing nature of the problems associated with interior and exterior building navigation, two separate methods were developed for each situation. Preliminary versions of an interior navigation algorithm (SRT-Target) and an instant goals approach for exterior navigation were proposed by the authors [5, 8] and are expanded upon here with greater capability. These strategies have been developed with the specific challenges due to operating in buildings and performing energy audits in mind. An overview of the proposed interior navigational and target identification process is shown in Figure 4.2. As stated, this paper primarily

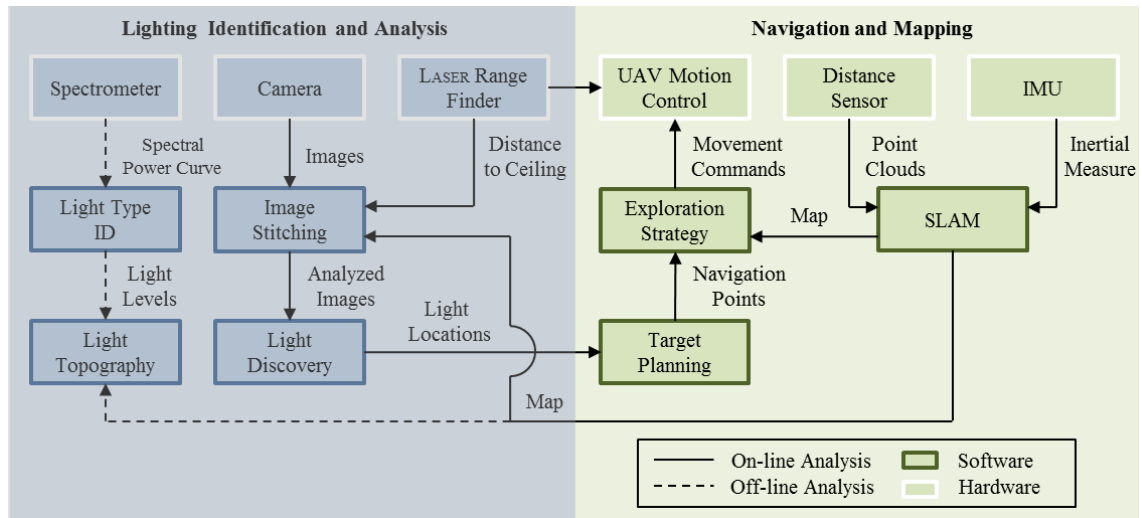


Figure 4.2: A diagram of the overall lighting audit information and process flow. The right portion of the figure is covered in this paper while the left greyed-out portion is covered in part 2.

focuses on the navigation and mapping strategy, shown on the right half of Figure 4.2. The identification and analysis of lighting is detailed in part 2 of this paper [96].

Briefly, part 2 of this paper describes the data collection and analysis of building lighting systems. As the robots navigate and explore the unknown space, a camera is used to take images of the ceiling. These images, along with distance measurements from a laser-range finder, undergo processing and analysis to determine the presence and location of lights. The images are stitched together using position data available from the navigational algorithm to prevent double light counting. Once a relative light location is determined, it is transformed to the global reference frame of the robot and passed to the navigational algorithm as a target location. The light locations are recorded and used later in simulations and generating recommendations. Upon reaching the light location, a spectrometer is used to gather spectral data from the light to assist in classifying the type of bulb in use. Using both spectral data from the spectrometer and size information from the camera images and distance measurements, the bulb type and size can be determined. After a space is fully explored, the lighting information gathered is used to perform energy efficiency analysis (amount of energy used by different lighting types, replacement and maintenance costs, electrical demand generated by lighting, etc.) and simulation of lighting levels to ensure the proper amount of lighting (underlit areas can create hazards and cause stress for occupants while overlit areas are opportunities for cost savings). Details of the processing, analysis, report generation, and type of recommendations given can be found in part 2 of this paper.

The rest of the paper is organized as follows. First there is a brief discussion of mapping and navigational strategies, followed by details of the proposed algorithms for interior and exterior navigation. Next, simulations of the interior and exterior algorithms are presented demonstrating the exploration capabilities of the robot in both interior and exterior scenarios. Future work is then described, including expanding the algorithms' capabilities

and developing sensing packages to perform various parts of an energy audit. Final conclusions cover the performance of the algorithms in simulation and the possibility of the algorithms use in other applications.

4.3 Navigational Strategies

Mobile robotic mapping has been an area of active research for many years. To successfully create a map of an area, a robot must be able to sense its environment and determine its location with respect to its surroundings. This problem is known as the Simultaneous Location and Mapping (SLAM) problem, which has been a large area of research in autonomous robotics. An example of what SLAM provides is shown in Figure 4.3. Obstacles and boundaries are shown in black, observed space is shown in light grey, and the robot's path is shown in red. This map was generated on a preliminary ground robot platform. Much research has been done on autonomous UVs in an outdoor environment which include GPS reliance to estimate the robots position and odometry [57, 58]; however, an increase in focus on SLAM systems that can perform indoors in GPS-denied environments has occurred over the last decade [59, 60, 61]. Different sensors have been utilized in these studies, such as 2D LASER range finders [62, 63, 64], monocular vision cameras [65, 66, 67, 68], stereo camera systems [69], and more recently, 3D depth cameras [70, 71, 72]. While SLAM solutions exist for operating indoors, robotic systems still require navigational strategies to determine where to move to and how to get there. For ease of implementation, work has started with a 2D LASER range finder for simplified navigational and mapping purposes. In parallel, a 3D depth camera (Microsoft Kinect) is being explored, with the aim of implementing full 3D mapping as demonstrated by a few [70, 97, 98]. A full 3D model will be used in generating automated energy simulation solutions.

Navigational strategies and path-planning for autonomous robots have been researched

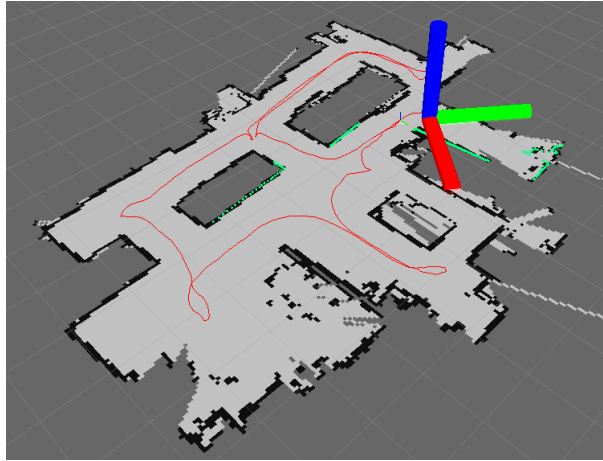


Figure 4.3: Map created by ground robot platform during interior navigation.

for quite some time. Many algorithms are available with a recent focus on using rapidly-exploring random trees (RRTs), originally proposed by LaValle in [73]. In this first paper, the RRT algorithm is described as starting at an initial point and growing “branches” in random directions whose selection is biased by their Voronoi diagrams. Essentially, branches are biased to grow towards regions that do not currently contain nodes. With enough iterations, the branches fill the space and there will exist a path from the starting point to any space (within the resolution of the tree). Later, LaValle built upon this work to propose RRT-Connect [74] which incrementally builds two RRTs. One originates from the starting point and the other from the goal. They expand until they connect with one another. Both of these strategies work in a known environment with a known goal, but do not provide a solution for a robot continually exploring an unknown environment. However, the random nature of the RRT algorithm provides for probabilistic complete exploration of an enclosed space, which is useful for interior exploration.

A data structure based on RRT was proposed by Oriolo in [75], called Sensor-based Random Tree (SRT). As the name states, this method adapts the RRT strategy to the char-

acteristics of typical hardware used in robotic sensing. The sensor perceives the surrounding and an associated safe region is determined. The tree is then incrementally built based on the generation of random points within this safe region. The resulting data structure represents a global map of safe regions which expands as the robot navigates. In [75], two exploration techniques are detailed: SRT-Ball and SRT-Star. Both methods assume 360° of perception, requiring a complete rotation sensor or integrating multiple devices that overlap to provide full coverage. SRT-Ball's radius of perception is limited by the nearest sensed obstacle. SRT-Star's area of perception is divided up into conical areas with each cone extending as far as the nearest obstacle in the cones region. SRT-Star is more confident as it uses more of the information available while SRT-Ball provides more conservative navigation. With SRT-Star, additional computation compared to SRT-Ball is required in determining each cone from one another as perception and navigation takes place. A method that builds on SRT-Ball and SRT-Star, SRT-Radial, was proposed by Espinoza in [76]. In the improved SRT-Radial method, the perceived safe region contours to obstacles, making complete use of the information available. In [76], 360° of perception is assumed, and with all the SRT strategies, the authors assume that perfect localization is provided by a SLAM method. Achieving 360° of perception can be expensive and difficult. Also, these strategies do not allow for identifying and navigating to targets as they are discovered in real-time.

The algorithm proposed by the authors for interior building navigation, termed SRT-Target, is built on SRT-Radial by adding the ability to check for targets in real-time and navigate to those targets, if necessary. In autonomous energy audits, these targets could be lights, air ducts, or other items of interest. Furthermore, a calibration factor was introduced to ensure the physical visitation of the robot to all areas. This factor accounts for the different perception ranges of the sensors used in the target identification, which is key in identifying all of the possible targets. SRT-Target was chosen as the method to be used on

the AuRAE for interior navigation and exploration due to its ability to explore unknown environments and facilitation of discovering targets in different situations, abilities that are required when performing audits in buildings. Currently, there are no other available algorithms that offer these capabilities together.

The SRT-Target method works for interior navigation of unknown environments that have a closed perimeter; however, for performing various audit tasks on the exterior of a building with potential openings, a different approach is required. Numerous boundary following procedures and techniques have been proposed over the years. Some have been based on fuzzy logic controllers [77], Lyapunov functions [78], or even the sensing mechanism of cockroaches [79]. Many use single or multiple ultrasonic sensors [80, 81, 82] and are meant to operate in known environments. These methods were designed to perform wall-following, but lack the capability to circumnavigate perimeters that are not fully closed. One boundary following method [6] uses instant goals to provide path planning. Instant goals allows for a robot to follow arbitrarily shaped shapes of simple or complex geometries. Additionally, the robot can encounter disturbing obstacles and continue moving along the previous boundary, subject to the robots geometry being able to pass between the two obstacles. For these reasons an algorithm based on the instant goals method presented in [6] was developed for external navigation. This proposed algorithm can follow convex and concave perimeters while being able to traverse potential openings in the boundary (open doors, passageways, etc.), which again, is not currently achievable with known algorithms. This is ideal for navigating the exterior profile of various industrial and commercial buildings; however, the proposed strategy can also be used where non-closed boundary following is required. The exterior perimeter can then be used a boundary for the interior navigation algorithm, allowing for interior exploration to occur without requiring the closure of building doors and openings which may interrupt business productivity.

4.3.1 Interior Navigation

For interior navigation, we propose an approach termed SRT-Target [5]. The algorithm can be seen in Figure 4.4. Briefly, the algorithm uses a data-tree structure as it incrementally builds a map of the environment as it explores. During each iteration the robot perceives its surroundings with the available depth sensor and generates the region S that gives the free and occupied space about the current location, q_{curr} . The current location and S are then added to the global map T . The space $S \cdot \beta$ is added to map U for the purpose of identifying targets during an energy audit, for example, ceiling lights. The calibration factor β is to restrict the robots movement to ensure that it visits every possible space that a target could be. Without β , the robot could add sensed areas to the global map large enough that targets could be outside of the target-sensing hardware’s range. The robot then begins searching for a candidate location to move to. If the robot is under a target, it will move towards the target, otherwise the robot searches for a random direction in which to proceed.

A valid candidate position is found if along the random direction the robot can travel a minimum distance and if the candidate position hasn’t been previously visited. After a certain number of iterations, if the robot cannot find a valid candidate position, it will move to the previous location and continue the search. As the robot fully explores an area, it will retrace its steps, eventually returning to its original starting location. This process is visually represented by a flowchart in Figure 4.5. The building- and audit-centric nature of the algorithm specifically addresses the challenges that arise from performing automated energy audits within buildings.

```

SRT_TARGET( $q_{init}, K_{max}, I_{max}, \alpha, d_{min}$ )
1   $q_{curr} \leftarrow q_{init};$                                 # Initialize current position.
2  for  $k = 1$  to  $K_{max}$ 
3       $S \leftarrow \text{PERCEPTION}(q_{curr});$                 # The perceived environment is stored in S.
4       $\text{ADD}(T, U, (q_{curr}, S, \beta));$                 # Add the current location and S to the global
5                                                    # tree T and add  $S \cdot \beta$  to the tree U.
6       $i \leftarrow 0;$                                 # Initialize the count.
7      loop                                            # Loop until a valid candidate is found
8                                                    # or max iterations are reached.
9           $(target, q_{target}) \leftarrow \text{TARGET\_CHECK};$  # Check for target at current location.

10     if ( $target = true$ )
11          $q_{cand} = q_{target};$                 # Make the candidate location the target location.
12         break;                                # Stop searching for candidate.
13     else
14          $\theta_{rand} \leftarrow \text{RANDOM\_DIR};$  # Pick random direction within sense range.
15          $r \leftarrow \text{RAY}(S, \theta_{rand});$                 # Find greatest distance  $r$  along
16                                                    #  $\theta_{rand}$  in the local safe region.
17          $q_{cand} \leftarrow \text{DISPLACE}(q_{curr}, \theta_{rand}, \alpha \cdot r);$  # Calculate candidate location
from
18                                                    # current location,  $r$ , and  $\theta_{rand}$ .
19          $i \leftarrow i + 1;$ 
20     end if
21     until ( $\text{VALID}(q_{cand}, d_{min}, U)$  or  $i = I_{max}$ )
22     if ( $\text{VALID}(q_{cand}, d_{min}, U)$ )
23          $\text{MOVE\_TO}(q_{cand});$                 # Move to candidate.
24          $q_{curr} \leftarrow q_{cand};$                 # Update current location.
25     else # If valid candidate can't be found move to parent location.
26          $\text{MOVE\_TO}(q_{curr}.parent);$ 
27          $q_{curr} \leftarrow q_{curr}.parent;$                 # Update current location.
28     end if
29 return  $T;$                                 # The result is given in the global map T.

```

Figure 4.4: The SRT-Target algorithm, previously presented in [5].

nearest point on the obstacle it senses, q_{wall} , relative to the robot. The robot then checks to see if it is within the user-defined distance away from the obstacle by calculating the distance between itself and the wall. If the robot needs to move closer or further away from the obstacle, it performs this motion and the algorithm proceeds as normal. A portion of the perceived environment S is then split into sections, or slices, and the nearest obstacle measurement point from each slice is found and stored in D . This allows for the data to be stored in a vector format, taking up less memory and providing quick access/processing. Figure 4.7 shows the perceived environment with obstacles split into sections. The number of slices can be defined with n_{slices} , depending on computation power and desired resolution. The portion of S to be stored is determined by the $srch_{from}$, $srch_{to}$, and $srch_{dir}$ parameters. The $srch_{from}$ parameter defines what heading to begin the slices at while the $srch_{to}$ parameter defines where to stop.

The $srch_{dir}$ parameter tells the algorithm which direction to move through the slices, left or right, which also determines if the robot will navigate the obstacle clockwise or counter-clockwise. This value can be predefined by the user or determined based on the perceived surroundings.

If the robot is at a corner, an extended perception will be performed to determine if a gap in the perimeter exists. The depth of this extended perception is determined by a user-defined factor, which should be equal to the largest expected gap to be encountered. An example case is shown in Figure 4.8, where a gap is found. Distances between the nearest obstacle point and the other points within the slices are then determined. If at a corner, the nearest point along the line extended from the last known wall reference point is used to determine $targ_{dist}$ as well as $targ_{head}$. If not at a corner, the maximum distance is found. The line between the nearest point and the point farthest away is calculated and the length of the line is used as the target distance to travel, $targ_{dist}$, and the slope of the

```

EXT_NAV( $q_{init}$ ,  $srch_{from}$ ,  $srch_{to}$ ,  $srch_{dir}$ ,  $n_{slices}$ ,  $d_{gap}$ )
1   $q_{curr} = q_{init}$ ;                                     # Initialize current position.
2   $S \leftarrow \text{PERCEPTION}(q_{curr})$ ;                     # The perceived environment is stored in S.
3   $q_{wall} = \text{NEAR\_OBS}(q_{curr}, S)$ ;                     # Find nearest point on obstacle.
4  if ( $\text{DIST}(q_{curr}, q_{wall}) \neq \perp_{dist}$ )
5       $q_{\perp} = \text{PERP}(q_{curr}, q_{wall}, \perp_{dist})$ ;        # If the robot is not the user-defined distance
6       $\text{MOVE\_TO}(q_{\perp})$ ;  $q_{curr} \leftarrow q_{\perp}$ ;          # away from the wall, move to the correct
7       $S \leftarrow \text{PERCEPTION}(q_{curr})$ ;                 # position and update  $S$  and  $q_{wall}$ .
8       $q_{wall} = \text{NEAR\_OBS}(q_{curr}, S)$ ;
9  end if
10 if ( $\text{CORNER}(S, q_{wall}) = \text{true}$ )                       # If at a "corner", extend the perception
11      $S_{extended} \leftarrow \text{PERCEPTION}(q_{curr}, srch_{dir}, d_{gap})$  # to determine if there is a gap.
12     if ( $S_{extended} \neq \text{empty}$ )                         # Divide  $S_{extended}$  into sections.
13          $D \leftarrow \text{SLICES}(S_{extended}, n_{slices}, srch_{from}, srch_{to}, srch_{dir})$ ;
14          $q_{dist} = \text{DIST}_{last}(D, q_{wall})$              # Find distance to last obstacle sensed.
15     else
16          $D \leftarrow \text{SLICES}(S, n_{slices}, srch_{from}, srch_{to}, srch_{dir})$ ; # Divide  $S$  into sections.
17          $q_{dist} = \text{MAX}(\text{DIST}(D, q_{wall}))$ ; # Find the max. distance between the nearest
18     end if                                               # wall point and the obstacles sensed in  $D$ .
19 else
20      $D \leftarrow \text{SLICES}(S, n_{slices}, srch_{from}, srch_{to}, srch_{dir})$ ; # Divide  $S$  into sections.
21      $q_{dist} = \text{MAX}(\text{DIST}(D, q_{wall}))$ ; # Find the max. distance between the nearest
22 end if                                                 # wall point and the obstacles sensed in  $D$ .
23  $[targ_{dist}, targ_{head}] = \text{LINE}(q_{dist}, q_{wall})$ ;    # Draw a line between the points
24  $q_{targ} = \text{TARGET}(q_{curr}, targ_{dist}, targ_{head})$ ;    # and find the target coordinate.
25 if ( $targ_{dist} = 0$ )
26      $\text{MOVE}(q_{curr}, targ_{dist-prev}, targ_{head-prev})$ ; # If only one obstacle is sensed
27 else                                                 # use previous motion.
28     if ( $\text{DIST}(q_{targ}, q_{wall}) < \perp_{dist}$ )             # If  $q_{targ}$  is too close to the obstacle, revise
29          $[targ_{dist}, targ_{head}] = \text{LINE}(q_{targ}, q_{wall})$ ; #  $q_{targ}$  to be farther from obstacle.
30          $q_{targ} = \text{TARGET}(q_{wall}, \perp_{dist}, targ_{head})$ ;
31          $\text{MOVE\_TO}(q_{targ})$ ;
32     else
33          $\text{MOVE}(q_{curr}, targ_{dist}, targ_{head})$ ;        # Move to target location.
34     end if
35 end if
36  $[targ_{dist-prev}, targ_{head-prev}] = [targ_{dist}, targ_{head}]$ ;
37 while  $q_{curr} \neq \text{THRESH}(q_{init})$  do Repeat lines 2-40;

```

Figure 4.6: The modified external navigation algorithm from [8], based off of the use of instant goals described in [6].

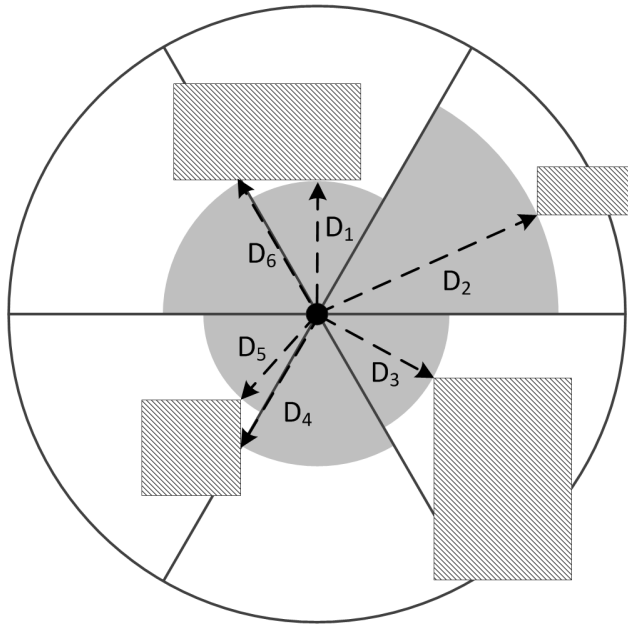


Figure 4.7: The perceived environment divided up into sections for the exterior navigation algorithm [8].

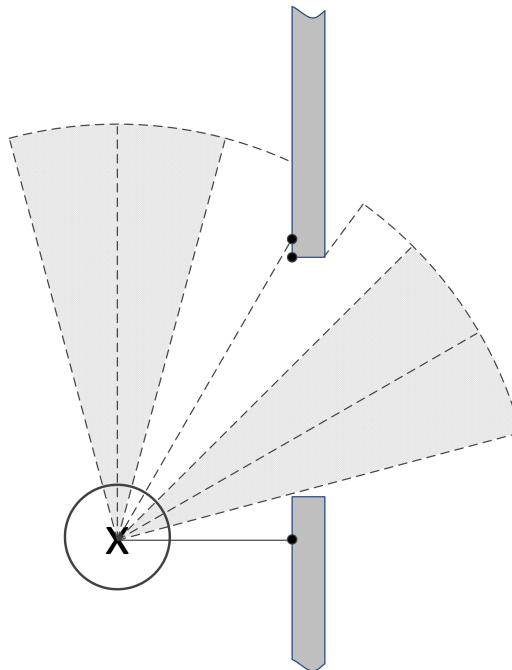


Figure 4.8: Exterior navigation across a gap.

line is used to determine the heading of the target location, $targ_{head}$. If only one obstacle distance measurement is available, such as near a corner, and nothing is found in the extended perception as shown in Figure 4.9, the robot will repeat the previous motion in order to reveal more of the obstacle and find more slices with obstacle measurements. Otherwise, the robot will move the target distance along the target heading along as the target coordinate does not move the robot within the use defined distance to remain from the obstacle. If the target coordinate is within this distance, than a new target coordinate is generated at the point that is on the line between the obstacle point and the old target coordinate with a distance equal to the required distance from the obstacle. An example of how the robot would navigate a corner is shown in Figure 4.10. This process is repeated until the robot moves within a threshold distance of the starting location. The algorithm is shown visually in a flowchart in Figure 4.11. Currently this method assumes the robot is placed in a starting position within the sensor's range of an obstacle. Again, the building-centric nature of the exterior navigational algorithm specifically addresses the challenges that arise from performing automated energy audits on buildings.

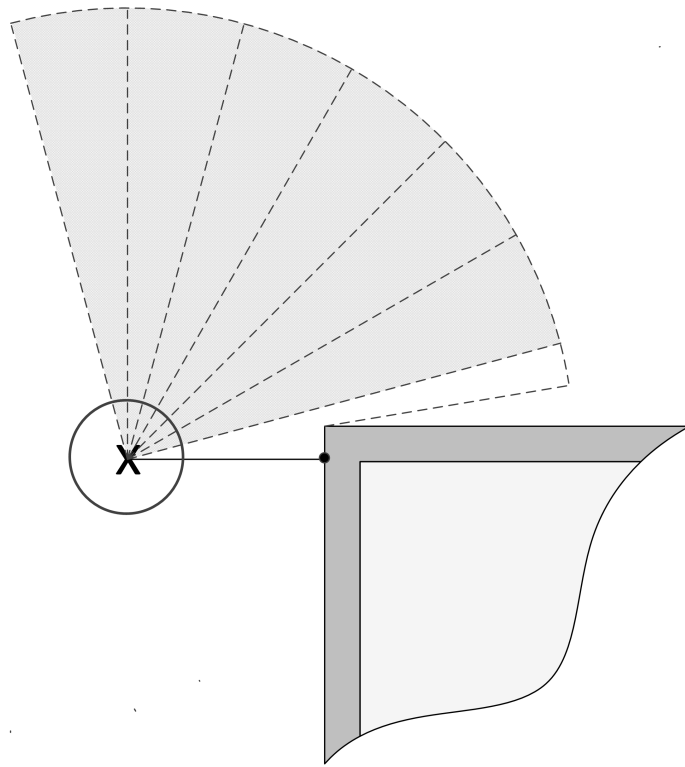


Figure 4.9: Performing an extended perception at a corner, not finding a gap. The grey slices indicate no obstacles detected.

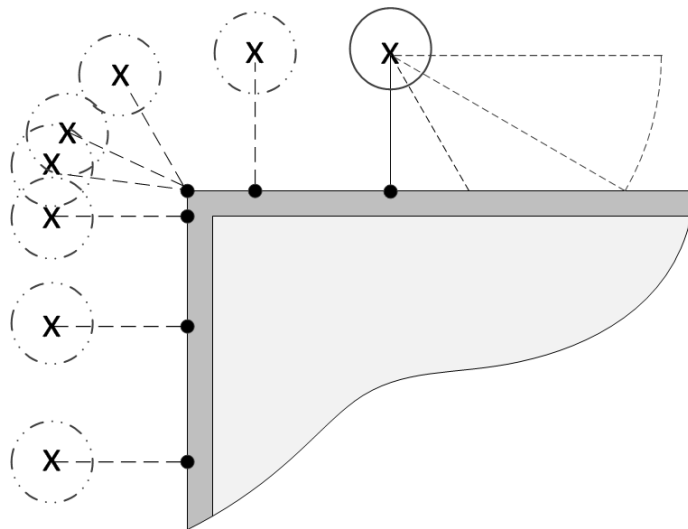


Figure 4.10: Exterior navigation around a corner [8].

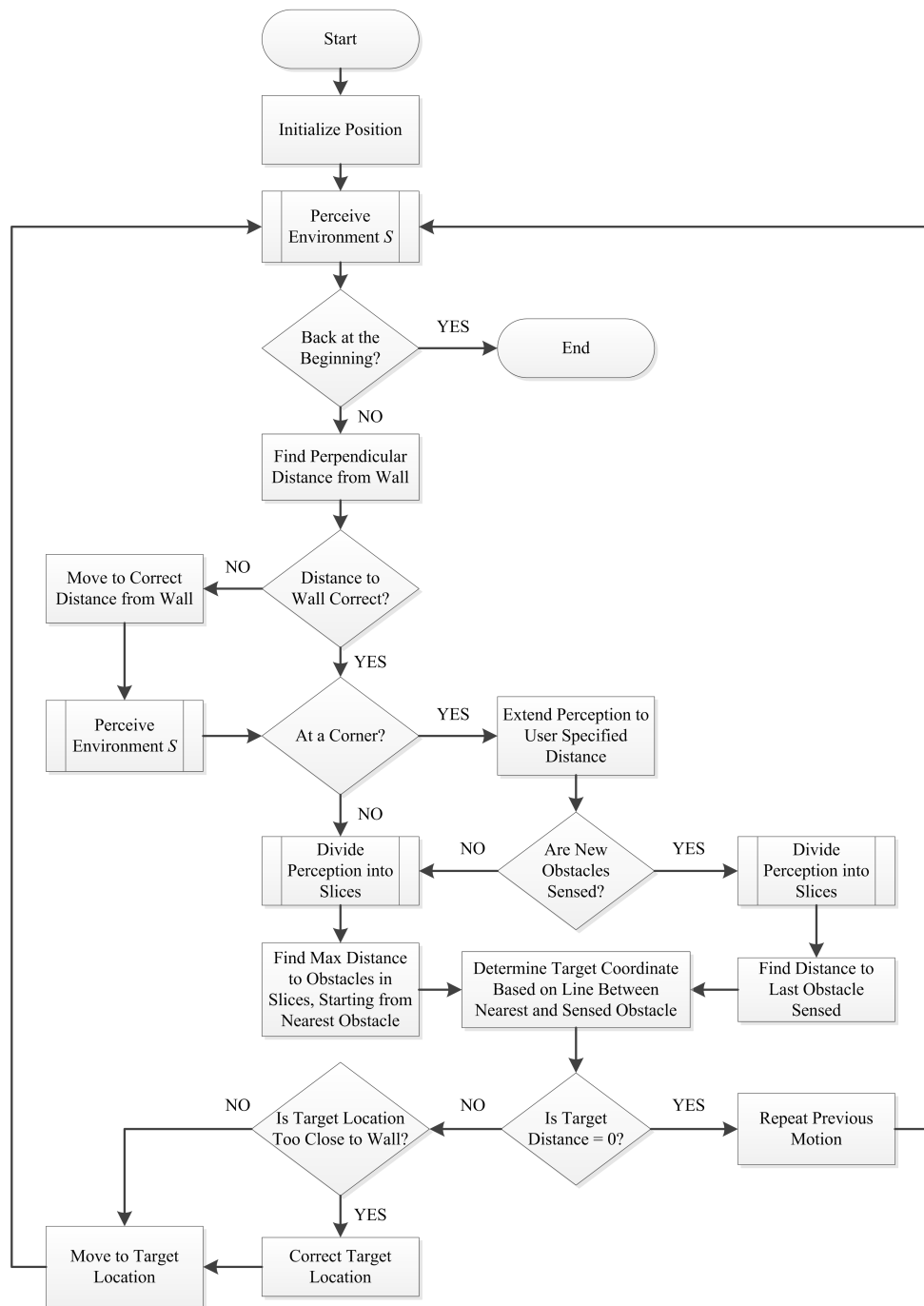


Figure 4.11: Flowchart showing the process of exterior navigational algorithm.

4.4 Simulation Verification

Simulations of the interior and exterior navigation strategies were performed to verify correct operation and adequate performance. For the interior simulation, the layout of a laboratory with a closet area, connected to an office by a hallway was chosen. This provided both open and smaller spaces to demonstrate the algorithm. For the exterior simulation, an arbitrary building shape was created, showing some common features that may be encountered during audits.

4.4.1 Interior Navigation Simulation

Figure 4.12 shows the evolution of the global map as the robot explores the interior of the laboratory environment. The robot, shown as an asterisk in blue, starts at the left end of the hallway. Initially, the robot makes its way into the office. Once fully explored, the robot then re-enters the hallway and discovers the entrance to the lab, fully exploring it before returning to the starting location. The tree-like nature of the path, shown in green, can be seen as the robot chooses random directions to explore in. At some points, the robot crosses its previous path. This is due to the fact that the algorithm does not consider a space fully explored until it has been visited twice and determined to not have any new avenues to pursue. During its exploration, the robot navigates to the rectangular fluorescent lights, shown in red. The center of the lights are represented by red squares, while the area of discovery of the lights is outlined by the red rectangles.

A series of interior simulations were conducted, consisting of five similarly sized layouts of different physical configurations and light locations. For each room, five simulations were performed with the robot starting from a different point each time. The results of these simulations are summarized in Table 4.2. In each case of the 25 different simulations the robot fully explored the layout and found 100% of the lights.

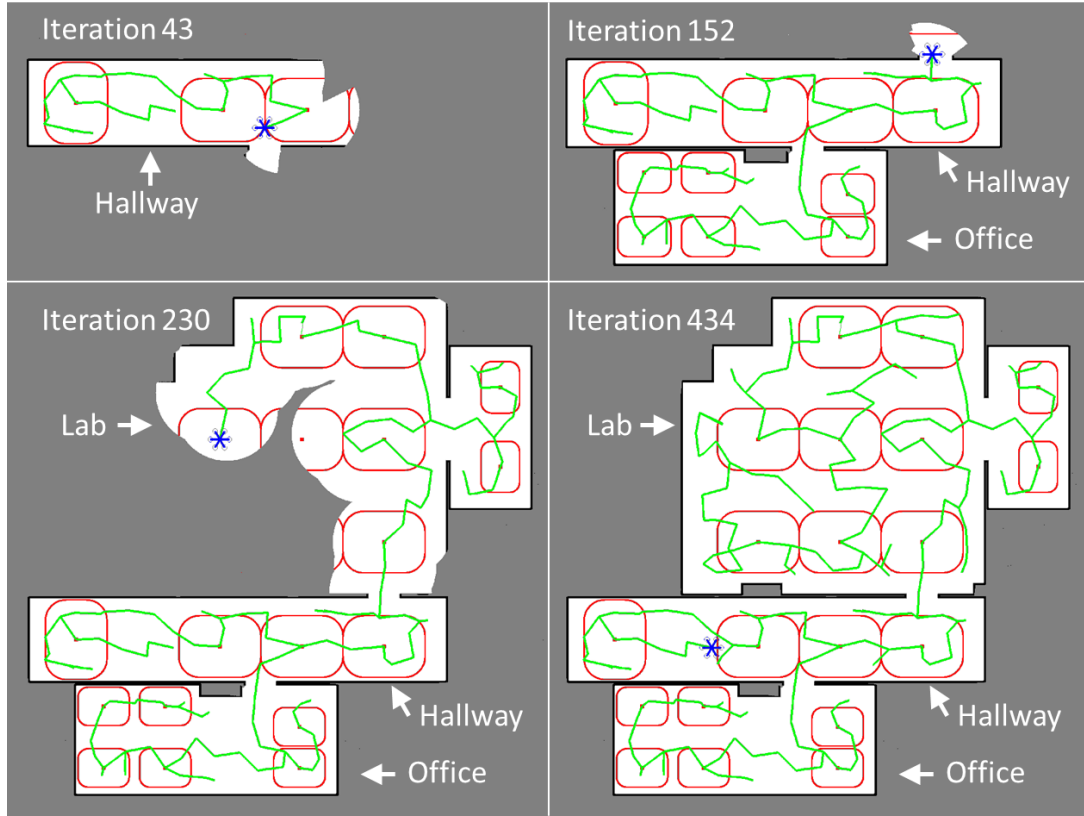


Figure 4.12: This figure shows the robot as it explores the interior of the building beginning in the top-left, then top-right, followed by bottom-left, and then bottom-right. The robot is shown in blue, the path of the robot in green, unexplored space in gray, obstacles in black, and free space in white. The tree-like nature of the interior navigational strategy can be seen as the exploration develops.

Table 4.2: Summary of simulation results for five different layouts. For each layout, five simulations were completed starting from five separate locations.

	Layout 1	Layout 2	Layout 3	Layout 4	Layout 5
Average Iterations	112.6	105.8	127.8	97.8	149.4
Avg. Distance Traveled (ft.)	130.3	125.0	152.5	111.6	175.2
Avg. Exploration Completion (%)	100.0	100.0	100.0	100.0	100.0
Correct Light Detection (%)	100.0	100.0	100.0	100.0	100.0

4.4.2 Exterior Navigation Simulation

The evolution of the external exploration path of the robot can be seen in Figure 4.13. As in the interior simulations, the robot is shown in blue, the robot's path in green, unexplored space in gray, obstacles in black, and free space in white. The robot tracks the walls of the building well, navigating both convex and concave corners with the simulation ending once the robot has returned to its starting location. It also successfully traverses the openings in the perimeter. At the corners and gaps the extended perception can be seen by the increase in explored space (shown as white). When the robot detects an obstacle with the extended perception (i.e. a large door opening), it traverses the gap. At the corners where no obstacle is detected with the extended perception, the robot navigates around the corner and continues around the perimeter.

4.4.3 Transitioning Between Exterior and Interior Navigation

Since the targeted environment is likely to have large open doors, establishing boundaries is important for the robot when operating in interior navigation mode. This can be done with the information gained from the exterior data collected. Open gaps in the perimeter can be closed and an invisible boundary can be imposed during interior navigation. For this to occur practically, on completion of the exterior navigation the robot could move to the nearest open entrance and pass inside the building, thus knowing its starting position in relation to the exterior map and boundaries. The interior starting location could also be provided by the technician upon startup through a graphical interface. A method involving pattern recognition and scan matching could be developed to allow for the robot to determine its location relative to the exterior map as it gathers more information about the interior environment. These are just a few possible solutions for this issue.

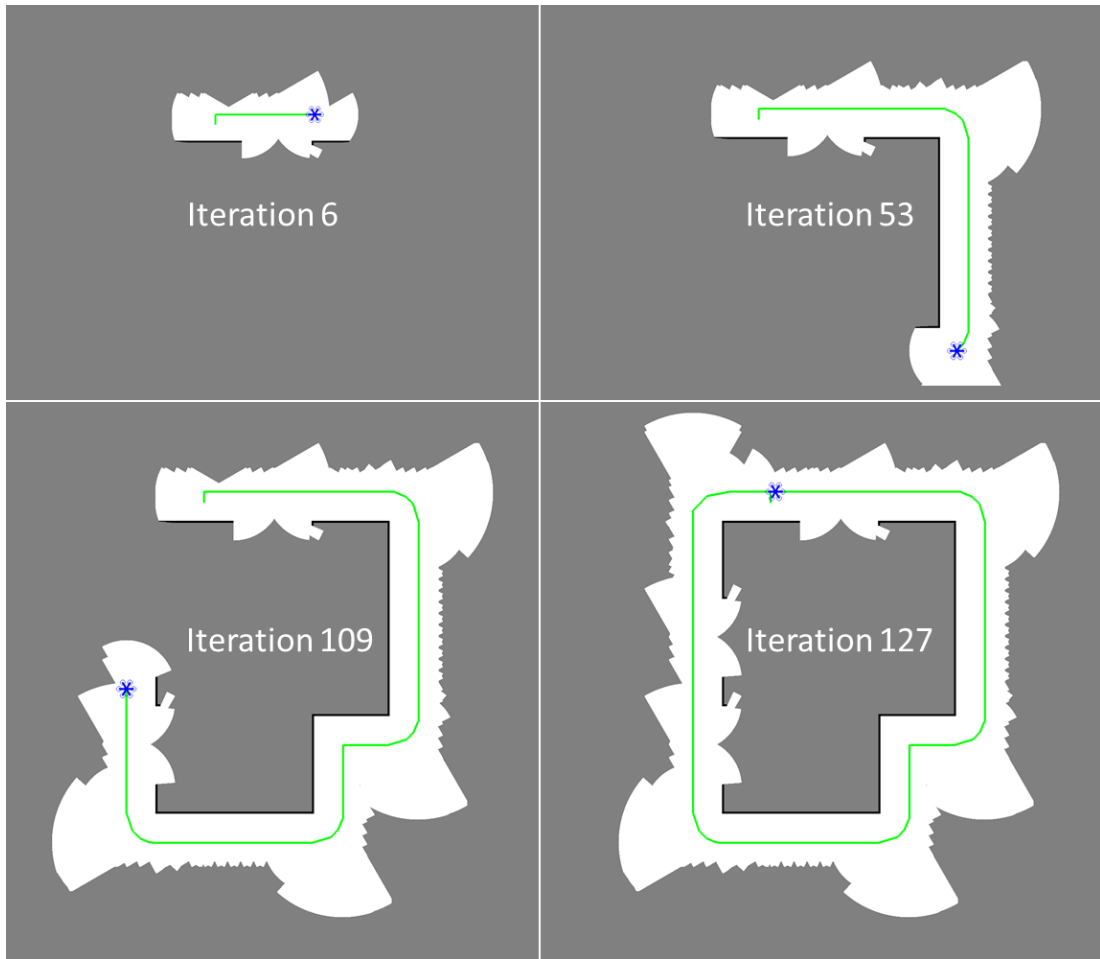


Figure 4.13: Exterior navigation simulation showing the progression as the robot travels around the building that has several large doors (gaps).

4.5 Future Work

Having established building- and audit-centric methods for interior and exterior building navigation, future efforts are focused on in-situ evaluation of the proposed algorithms using ground and aerial vehicles. The platforms currently under development can be seen in Figures 4.14 and 4.15. Work is ongoing to extend the algorithm to accommodate the eccentricities that are common with exterior building envelopes. Pillars and trees near

a building offer navigational challenges in obstacle avoidance. A ground robot platform is under development to experimentally validate both the exterior and interior navigation strategies. A lighting identification package has been developed and is planned to be integrated with the ground robot, a UAV, or used as a stand-alone hand-held device.



Figure 4.14: Ground robot currently under development.

Beyond experimental validation of the navigation strategies and the lighting identification package, future work includes the design of other sensing packages. The final platform will be able to locate and identify lights, take several environmental measurements (temperature, humidity, CO_2), enumerate building occupants, and provide a comprehensive 3D map of a building for energy modeling purposes. Creating a mode in which the UVs would follow the technician is of interest, addressing navigational difficulties such as doors and elevators (enabling multi-floor access). Also, a technician interface will be required, as well as automation of the report generation based on the measurements and

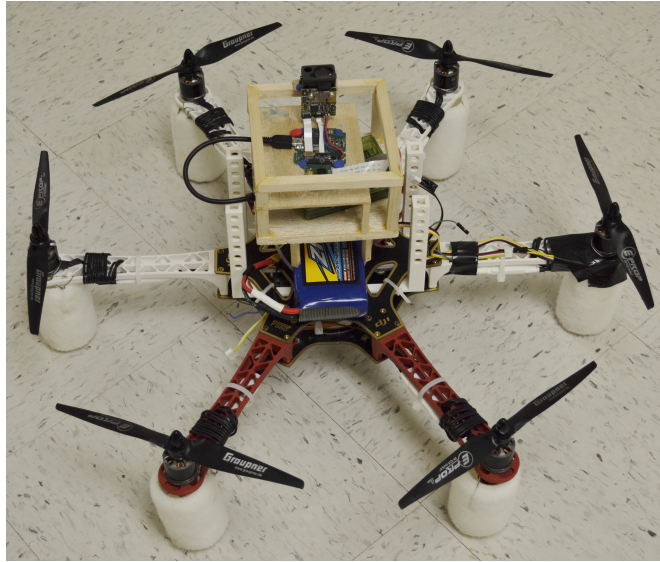


Figure 4.15: UAV platform with lighting package currently under development.

observations made.

4.6 Conclusion

In this paper, AuRAE were discussed and navigational algorithms for use by ground and aerial vehicles during AuRAE were presented. These navigational algorithms are needed by robots if the cost of energy audits are to be reduced. SRT-Target was selected for interior navigation and an instant goal approach with the ability to navigate gaps in the perimeter was developed for exterior navigation. The algorithms were then simulated on different building layouts. The simulations showed successful performance of the algorithms for both interior and exterior navigation. The algorithms presented in this paper are just one step of many in the development of AuRAE. While developed for performing autonomous energy audits, these algorithms are also applicable in other cases where target identification in unknown environments or non-closed boundary following is required.

5. PAPER C: AUTONOMOUS ROBOTIC BUILDING ENERGY AUDITS: DEMONSTRATED CAPABILITIES AND OPEN CHALLENGES¹

5.1 Synopsis

Building operations are a significant consumer of energy and contributor to carbon emissions in the U.S. and around the globe. Energy audits offer significant potential in reducing building energy use by providing tailored recommendations involving equipment upgrades, operational adjustments, and building recommissioning. However, energy audits are a time intensive process that requires significant experience and training. This causes high costs related to performing an audit and prevents many businesses from having an audit completed. Automating the audit process will not only reduce the cost of audits, but clients will be provided with more repeatable and accurate recommendations based on improved data collection and analysis. Previous work and the current state of the art of robotic auditing tools are discussed in this paper. Preliminary results from a lighting assessment prototype and an environmental baseline measurement prototype are presented. The paper closes with open challenges and future possibilities of autonomous vehicles for conducting audits.

5.2 Introduction

Designed to operate efficiently when constructed, building performance can degrade over time. This degradation can be due to normal deterioration of systems, altered use or misuse from the original intended operation, or even more gradual changes in climate and the surrounding environment. Moreover, new building technologies are being developed and made available at an increasing rate, but frequently these advances only appear in new

¹This is the authors accepted manuscript (LB-17-001) of an article published as the version of record in ASHRAE Transactions © 2017 by ASHRAE. <https://ashraem.confex.com/ashraem/s17/webprogram/Paper22462.html>

construction and not existing buildings. As such, productivity losses and energy inefficiencies add up, resulting in increased cost to businesses and building owners. To give some monetary perspective, buildings accounted for 82.2%, or \$301.6 billion of the total U.S. electricity expenditures in 2010 [2]. With such a large financial factor, even small inefficiencies become significant. Beyond the economic cost resulting from these deficiencies, the energy used in buildings has a significant impact on the environment. The U.S. Energy Information Administration has reported that building operations account for approximately 40% of all energy usage in the United States [15]. Furthermore, buildings and their associated energy consumption account for 40% of U.S. carbon emissions [16]. Commercial buildings alone account for half of the building energy usage [2] and building carbon emissions [17]. Between the financial and environmental effects of buildings, any improvement in their energy consumption is welcome and worth pursuing.

Energy audits of buildings are a proven mechanism for achieving significant energy savings and reductions. Gaining initial popularity after the energy crises of the 1970s, energy audits have seen a recent resurgence in popularity as society has gained a better understanding of human impact on the environment as well as monetary incentives such as savings and recently created tax incentives. The savings from audits can be substantial; for example, small to medium sized manufacturers receiving one day energy audits have resulted in an average of approximately 17% annual savings per client [94]. These facts notwithstanding, energy audits are still widely underused by commercial and industrial entities. While there is no national census detailing how many buildings could benefit from an energy audit, a U.S. industry survey estimated that below 5% of existing buildings have received audits [19]. Another study in the state of California estimated only 0.03% of existing buildings and 5% of newly constructed buildings have received audits [20].

One potential major barrier to the more widespread use of energy audits is cost [99]. Assessments can be complex in nature and require significant personnel hours, extensive



Figure 5.1: Levels of energy audits, as defined by the American Society of Heating, Refrigeration and Air-Conditioning Engineers adapted from [7].

training over several years, and professional certification. Figure 5.1 shows a delineation of the varying levels of an energy audit, as defined by the American Society of Heating, Refrigeration and Air-Conditioning Engineers (ASHRAE). A preliminary energy-use analysis is the simplest in nature as the auditor examines utility bills and other energy use documents, performs some data analysis, and provides recommendations based on similar businesses or facilities. A Level 1 audit requires auditors to visit a site and perform a high-level walk-through of the buildings, visually identifying energy saving measures and collecting limited data. This level of audit is usually limited to a day of on-site work, including travel. The next level audit, Level 2, is more involved, necessitating several auditors to gather many measurements and perform a complete, in-depth analysis of the energy usage. A Level 2 audit can last several days, depending on the size of the facility. The most comprehensive of all audits, a Level 3 audit, involves extensive measurement collection (both environmental and structural) to support software energy simulations of the buildings. These simulations, along with a more refined analysis and study of the data, produces the most tailored and inclusive set of possible energy saving recommendations. A Level 3 audit can last up to several weeks. Time estimates of audits are taken from guidelines given by [100]. With a significant portion of an auditor's time devoted

to taking measurements, processing data, and choosing recommendations, any automation of these steps would lower the cost of an audit. Another variable in the cost of energy audits is the experience training required of auditors. To become fully certified, auditors must attend specific training and courses, followed by a certification test, all which carry financial burdens. In addition to this official training, auditors are required to have a minimum of 3 years of experience before becoming certified. The capital costs of auditor salary combined with training and certification constitute a substantial component of the cost of audits. Furthermore, experience can vary from auditor to auditor. Although auditors are subject to the same certification courses, in practice they can implement the audit procedures differently, resulting in variation of recommendations and potential savings. By automating the audit process, costs are greatly reduced and audits can be made more repeatable while providing increased richness of data. An estimation of the difference in costs is given in Table 5.1. An average annual salary of \$70,000 was based on numbers reported in [101]. The biggest savings can be seen in the capital investments made between auditors and automated audits. Additionally, the cost per audit is lower due to personnel and time savings.

The rest of the paper is organized as follows. In discussing the state of the art, the authors will first summarize previously published methods for conducting automated building light audits, and then include the results of recent field tests of these methods using campus buildings using Unmanned Ground Vehicles (UGV). This will be followed by a demonstration using a Unmanned Aerial Vehicle (UAV) and multi-sensor system to collect building environmental data. Open challenges will then be discussed, specifying areas where investigation is still needed. The last section will offer some conclusions about the current state and future direction of automated robotic energy audits.

Table 5.1: Comparison of costs between auditors and automated audits [3].

Auditors		AuRAE	
Training courses	\$15,000	UAV (x5)	\$6,000
Certification class	\$3,000	Technician training	\$1,500
Minimum 3 years experience (pay x2)	\$420,000	Sensors (x5)	\$15,000
Capital investments	\$438,000		\$22,000
Travel (2 auditors)	\$1,500	Travel (1 technician)	\$750
Cost of audit (20 hrs at \$60/hr)	\$2,400	Cost of audit (20 hrs at \$60/hr)	\$1,200
Cost per audit	\$3,900		\$1,950

5.3 Background

Robots are increasingly being used as tools for building inspection and auditing. Over recent years, the availability of economical robots and sensors has allowed for development of several new devices. Industrial inspection and maintenance has seen the development of serpentine robots [22] and climbing machines [23] that eliminate the need for workers to enter dangerous and difficult-to-reach areas. Much research has been completed concerning thermography of buildings. One group is developing a thermal, indoor ground-robot mapper [24]. Their aim is to create full 3D models of thermal maps of indoor environments. Preliminary results show the capability to overlay surface temperature measurements on 3D building models and is promising for analyzing heat distribution and flows on objects, but lacks the identification of thermal distribution in the full 3D space (the surrounding air), which would aid in advanced building modeling and fault detection of equipment. Other researchers have attached thermal cameras to unmanned aerial vehicle (UAVs) to examine both the interior [25] and exterior [26, 27, 28] of buildings. Oreifej et al. have produced a backpack device that produces similar output to Borrmann’s work,

which is excellent for identifying hot or cold spots in insulation, potential thermal leaks, and other thermally significant objects; however, the backpack device is rather large (approximately 2 feet by 4 feet) and requires the auditor to physically visit every space to be mapped. Wang et al. developed a 3D light detection and ranging (Lidar) system capable of rapidly collecting and fusing 3D point cloud and temperature data from the exterior of existing buildings. They also developed the ability to measure the temperature of transparent windows. Currently the system is only a hand-held device, limiting its ability to thermally map large and/or tall buildings. Mauriello et al. performed a user study regarding the use of thermography specific devices in energy audits. Some of the outcomes of the study were a desire for automation in data collection as well as report generation. Several groups have even worked to create 3D models suitable for energy simulation software by using a UAV equipped with a Lidar sensor [29] or just hand-held devices [30, 31]. Roca et al. used GPS sensors along with aerial photography to produce thermal maps of the exterior of buildings; unfortunately, their solution, due to its reliance on GPS and large size of UAV platform, is only able to perform exterior analysis of a building. Son et al. developed a very accurate method of mapping thermal image data onto 3D point clouds, however the portability and size of the solution limits its ability to be packaged for use on a UAV, thus requiring human intervention. Researchers in Europe have even proposed a model that simulates the behavior of a lighting assessment robot [32]. This model is aimed to reduce the time-to-market for companies developing automated auditing devices, but only simulates the device behavior and does not test the method on a physical prototype. While much research has been completed in automating building inspection tasks, much of it has centered on building thermography as that area has been the most accessible. To the author's knowledge, a complete solution to the problem of automating building energy inspections and audits does not exist, and that is the gap that this paper aims to address. For these reasons, the authors have previously proposed Autonomous Robotic Assessments of

Energy (AuRAE) [5, 96].

5.4 Overview of Autonomous Robotic Assessments of Energy

Considered as state of the art, these AuRAE will be performed by a cooperative fleet of ground and aerial vehicles, which gather data and measurements as they navigate through and around buildings. For energy modeling and making recommendations, the measurements of interest are temperature, relative humidity, human occupancy, CO₂ levels, lighting levels, light locations & types, missing/faulty insulation (both in walls and around windows), ducting leaks, thermal reflectivity of the exterior, and overall dimensions of the building (interior & exterior), etc. From these data, many of the typical recommendations for energy and cost savings can be made. Additionally, 3-dimensional models can be automatically generated in such a format that they can be imported into energy simulation software so that buildings with varying conditions and equipment can be simulated. The combination of these capabilities will result in an in-depth report given to the building owner/operator with recommendations for energy savings.

Initially, these teams of robots will augment the manual audit process, eliminating simple, time-consuming tasks and providing additional depth of analysis not possible with a human auditor. The robots will operate either semi-autonomously, capable of following an auditor in difficult navigational situations (closed doors), or autonomously, freeing up the technician to monitor other processes or duties. Unmanned vehicles (UVs) will work to create maps of the environment and take measurements, while ground robots serve as base stations, relaying data and providing recharge locations as needed. Different sensing packages can be made for the UVs, providing different capabilities. These packages could be interchanged as needed, depending on the requirements for the audit. Additionally, special dedicated UVs could be developed, such as mini or micro drones that perform measurements requiring small payloads, such as temperature, humidity, and light levels,

while large UVs could support more specialized equipment, such as IR thermal cameras, 3D scanning platforms, and human occupant identification devices. Upon completion of this research, entirely automated audits performed by these robotic fleets are envisioned. Eventually, the successful implementation of AuRAE will increase the number of buildings receiving an energy audit that, in today's business climate, would not normally receive an audit. This will produce significant energy savings, reducing the demand on current power grids and energy sources as well as lowering business operating costs and raising profitability.

While technical in nature, the benefits of AuRAE over a traditional audit are several. Gathering data continuously as the robots travel through the building result in greater data resolution and detailed maps of the environment versus single point measurements taken by an auditor. The speed of these measurements is significantly faster, recording several different types of readings simultaneously resulting in reduced audit times. In addition to speed enhancements, the robots will give more repeatable results, without the variation of experience that auditors have. To operate the teams of robots, less training will be required than that of current auditors. These AuRAE could be performed once, or become reoccurring audits, providing improvements in data and sensing from previous assessments. The fleets could be sold as off-the-shelf sets, allowing for continuous use by building managers and technicians. If purchased for a building, a fleet could interact with the building's energy management system. Week-to-week or day-to-day energy efficient operation and comfort levels for the occupant could be achieved.

Several unique challenges arise due to the automation of these tasks. When navigating in an indoor building environment, robots are not able to utilize current positioning technologies, such as the global positioning system (GPS). While some facilities may have blueprints available, translating these plans into a sufficiently accurate map for the robots navigational use is not trivial; therefore, the robots must fully explore an unknown building

environment autonomously, tracking their location as they travel. Another layer of complexity is added in that specific objects and related data must be identified and measured. Thus, the robots must be able to identify these objects of interest, alter their exploration path, navigate to the targets, and make the required measurements and observations. To solve the challenges of navigating indoors, an integrated mapping, localization, exploration, and target identification solution is needed. The next sections discuss a proposed solution to the aforementioned challenges.

5.4.1 Navigational Strategies

Solutions are needed for both the indoor and outdoor navigation problems that come with operating in and around buildings. But before implementing navigational strategies, a robot must be able to sense the environment around it and locate itself within that environment. This problem is known as the Simultaneous Location and Mapping, or SLAM, problem [102]. The SLAM problem is several decades old and has seen many techniques proposed; however, an increase in focus on performing SLAM indoors in GPS-denied environments has occurred over the last decade [59, 60, 61]. Varying sensors have been utilized in these studies, such as 2D laser range finders [63, 64, 62], monocular vision cameras [65, 66, 67, 68], stereo camera systems [69], and more recently, 3D depth cameras [70, 71, 72, 97, 98]. In the development of AuRAE, the authors have used two different SLAM solutions: 1) the Hector SLAM algorithm [63], which uses a robust laser scan matching method, and 2) Real-Time Appearance-Based Mapping (RTAB-Map) [86], which is a RGB-D graph SLAM approach based on a global Bayesian loop closure detector. Both solutions update their maps in real-time, addressing the issue of temporary obstacles (e.g. an occupant walking down a corridor). Hector Mapping has been implemented with a 2D laser scanner on a ground robot platform while RTAB-Map has been implemented with an IR camera on a UAV platform. More on this can be found in the

following sections.

Once a robot can sense the surrounding environment and locate itself, a strategy must be developed for navigating through said environment. While there exist algorithms for exploring unknown areas in GPS-denied environments, robots performing automated energy audits require the ability to integrate targets that are identified in real-time into the navigational plan. To address this requirement, the authors have previously proposed an algorithm termed SRT-Target [5]. SRT-Target is an algorithm capable of completely exploring a previously unknown, enclosed space while navigating to objects that are identified in real-time through a sensor package. Pseudo-code for the algorithm is shown in Figure 5.2. Complete details of the algorithm and how it works can be found in [5]. A brief summary of the algorithm follows. The robot first initializes its current position. Then an initial perception of the surrounding environment is taken and stored in variable S . Next, S is added to the global map T and $S \cdot \beta$ is added to the map U , a reduced global map. $S \cdot \beta$ is a reduced version of S that facilitates the robot physically visiting each space to ensure all objects of interest are identified. This is necessary because of the hardware limitations of the sensing package that require the robot to be within a certain distance of any target. A loop is then entered to find a valid candidate position to move to. The robot first checks if there is an object of interest within range. If so, the object's location becomes the candidate location, and the robot proceeds to those coordinates and the loop is exited. If there is no target, a random direction is chosen. The distance along this direction to the nearest obstacle is determined, within the limits of the range sensor. It should be noted that the obstacle boundaries are expanded in such a way to ensure the robot will never come close enough to contact them. A candidate position that is a variable distance (dependent on α) along that direction is generated and then checked to be valid, ensuring it exceeds a certain distance threshold and is within a region that hasn't previously been sensed twice. The robot moves to the location if it is valid; otherwise, the robot

```

SRT_TARGET( $q_{init}, K_{max}, I_{max}, \alpha, d_{min}$ )
1   $q_{curr} = q_{init};$                                      # Initialize current position.
2  for  $k = 1$  to  $K_{max}$ 
3       $S \leftarrow \text{PERCEPTION}(q_{curr});$                  # The perceived environment is stored in S.
4       $\text{ADD}(T, U, (q_{curr}, S, \beta));$                  # Add the current location and S to the global
5                                                         # tree T and add  $S \cdot \beta$  to the tree U.
6       $i \leftarrow 0;$                                      # Initialize the count.
7      loop                                               # Loop until a valid candidate is found
8                                                         # or max iterations are reached.
9           $(target, q_{target}) \leftarrow \text{TARGET\_CHECK};$  # Check for target at location.
10         if ( $target = true$ )
11              $q_{cand} = q_{target};$                      # Make the candidate location the target location.
12             break;                                     # Stop searching for candidate.
13         else
14              $\theta_{rand} \leftarrow \text{RANDOM\_DIR};$  # Pick random direction within sense range.
15              $r \leftarrow \text{RAY}(S, \theta_{rand});$            # Find greatest distance  $r$  along
16                                                         #  $\theta_{rand}$  in the local safe region.
17              $q_{cand} \leftarrow \text{DISPLACE}(q_{curr}, \theta_{rand}, \alpha \cdot r);$  # Calculate candidate location
18              $i \leftarrow i + 1;$                          # from current location,  $r$ , and  $\theta_{rand}$ .
19         end if
20         until ( $\text{VALID}(q_{cand}, d_{min}, U)$  or  $i = I_{max}$ )
21         if ( $\text{VALID}(q_{cand}, d_{min}, U)$ )
22              $\text{MOVE\_TO}(q_{cand});$                        # Move to candidate.
23              $q_{curr} \leftarrow q_{cand};$                  # Update current location.
24         else # If valid candidate can't be found move to parent location.
25              $\text{MOVE\_TO}(q_{curr}.\text{parent});$ 
26              $q_{curr} \leftarrow q_{curr}.\text{parent};$        # Update current location.
27         end if
28 return  $T;$                                            # The result is given in the global map T.

```

Figure 5.2: Pseudo-code for the SRT-Target indoor navigational algorithm [5].

continues the search for a valid location candidate, up to a fixed number of iterations. If no valid candidate location can be found, then the robot returns to the last, or parent location and starts the process over. In this fashion, the robot completely explores an enclosed space, ensuring discovery of all objects of interest. Upon finding no new valid candidate locations, the robot returns to its starting point.

Simulations of the interior navigational algorithm SRT-Target have been performed [3]. Results from some of the simulations are shown in Figure 5.3. The UAV can be seen as a star shape in blue, its trajectory by the thin green line, objects of interest (in this case, lights) and their associated radius of discovery are represented by the rounded rectangles in red, obstacles are shown in black, open space is shown in white, and unknown territory is shown in grey. The layout shown is representative of an actual office space, hallway, and laboratory. As the iterations increase, beginning in the top left and finishing in the bottom right, the random navigational direction of the UAV can be seen. Upon entering

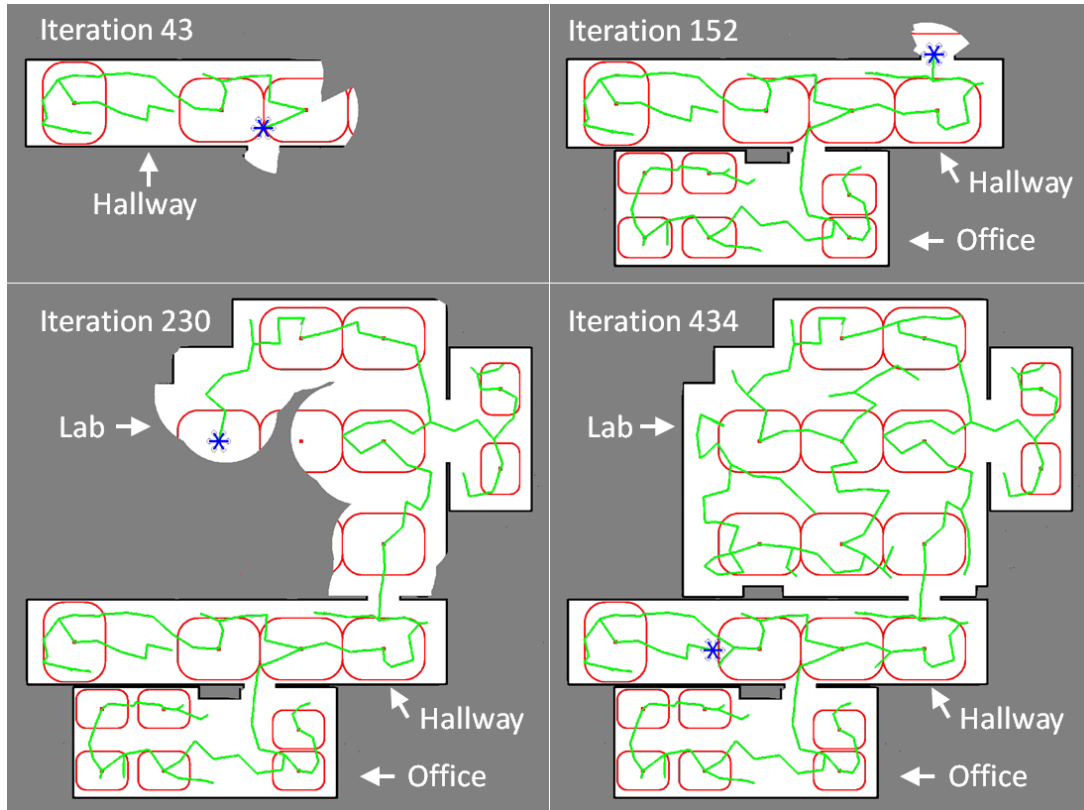


Figure 5.3: Results from a simulation of the SRT-Target algorithm [3]. The UAV is shown as the star shape in blue, its path as the thin green line, objects of interest as red dots with their associated radius of discovery as rounded red rectangles, obstacles as black, open space as white, and unknown territory as grey.

the radius of discovery for a light, the UAV navigates to the center of the light to facilitate data measurement (spectrometer and light level). This behavior can be repeated for any object of interest, provided there is a sensor package to determine the presence of the object.

In addition to SRT-Target, a navigational strategy for the exterior of buildings is needed. An ideal strategy would be able to follow arbitrarily shaped buildings, with both convex and concave corners, and be able to traverse open bay doors, common with commercial and industrial buildings. This would simplify the audit process by not requiring the customer to close all exterior openings that may be open during normal use. The authors have previously proposed a solution to the exterior navigation problem in [8], utilizing the method of instant goals [6]. Pseudo-code of the exterior algorithm is shown in Figure 5.4. Details of the algorithm can be found in [8], but a brief summary is given here. The robot again begins with an initialization of its current position and initial perception of the surrounding environment, dividing up the area of perception into slices, as shown in Figure 5.5(a). Finding the nearest obstacle, the robot moves to a user-defined, perpendicular distance away. The robot then perceives the space in front of and alongside it, determining if it has reached a corner. If the robot has reached what it determines to be a corner, it will extend its perception range out to a user-defined value dependent on the largest exterior gap that should be traversed, shown in Figure 5.5(b). If no additional obstacle is found, the robot will move around the corner in an arc as shown in Figure 5.5(c). If an additional obstacle is found, the robot determines that there is a traversable gap and will navigate to the other side, as shown in Figure 5.5(d). This process of sensing the wall and determining the distance to travel along the perimeter is repeated until the robot reaches its starting location.

```

EXT_NAV( $q_{init}$ ,  $srch_{from}$ ,  $srch_{to}$ ,  $srch_{dir}$ ,  $n_{slices}$ ,  $d_{gap}$ )
1   $q_{curr} = q_{init}$ ;                                     # Initialize current position.
2   $S \leftarrow \text{PERCEPTION}(q_{curr})$ ;                   # The perceived environment is stored in S.
3   $q_{wall} = \text{NEAR\_OBS}(q_{curr}, S)$ ;                 # Find nearest point on obstacle.
4  if ( $\text{DIST}(q_{curr}, q_{wall}) \neq \perp_{dist}$ )
5       $q_{\perp} = \text{PERP}(q_{curr}, q_{wall}, \perp_{dist})$ ;    # If the robot is not the user-defined distance
6       $\text{MOVE\_TO}(q_{\perp})$ ;  $q_{curr} \leftarrow q_{\perp}$ ;        # away from the wall, move to the correct
7       $S \leftarrow \text{PERCEPTION}(q_{curr})$ ;                 # position and update  $S$  and  $q_{wall}$ .
8       $q_{wall} = \text{NEAR\_OBS}(q_{curr}, S)$ ;
9  end if
10 if ( $\text{CORNER}(S, q_{wall}) = \text{true}$ )                    # If at a "corner", extend the perception
11      $S_{extended} \leftarrow \text{PERCEPTION}(q_{curr}, srch_{dir}, d_{gap})$  # to determine if there is a gap.
12     if ( $S_{extended} \neq \text{empty}$ )                      # Divide  $S_{extended}$  into sections.
13          $D \leftarrow \text{SLICES}(S_{extended}, n_{slices}, srch_{from}, srch_{to}, srch_{dir})$ ;
14          $q_{dist} = \text{DIST}_{last}(D, q_{wall})$            # Find distance to last obstacle sensed.
15     else
16          $D \leftarrow \text{SLICES}(S, n_{slices}, srch_{from}, srch_{to}, srch_{dir})$ ; # Divide  $S$  into sections.
17          $q_{dist} = \text{MAX}(\text{DIST}(D, q_{wall}))$ ; # Find the max. distance between the nearest
18     end if                                           # wall point and the obstacles sensed in  $D$ .
19 else
20      $D \leftarrow \text{SLICES}(S, n_{slices}, srch_{from}, srch_{to}, srch_{dir})$ ; # Divide  $S$  into sections.
21      $q_{dist} = \text{MAX}(\text{DIST}(D, q_{wall}))$ ; # Find the max. distance between the nearest
22 end if                                           # wall point and the obstacles sensed in  $D$ .
23  $[targ_{dist}, targ_{head}] = \text{LINE}(q_{dist}, q_{wall})$ ; # Draw a line between the points
24  $q_{targ} = \text{TARGET}(q_{curr}, targ_{dist}, targ_{head})$ ; # and find the target coordinate.
25 if ( $targ_{dist} = 0$ )
26      $\text{MOVE}(q_{curr}, targ_{dist-prev}, targ_{head-prev})$ ; # If only one obstacle is sensed
27 else                                           # use previous motion.
28     if ( $\text{DIST}(q_{targ}, q_{wall}) < \perp_{dist}$ )          # If  $q_{targ}$  is too close to the obstacle, revise
29          $[targ_{dist}, targ_{head}] = \text{LINE}(q_{targ}, q_{wall})$ ; #  $q_{targ}$  to be farther from obstacle.
30          $q_{targ} = \text{TARGET}(q_{wall}, \perp_{dist}, targ_{head})$ ;
31          $\text{MOVE\_TO}(q_{targ})$ ;
32     else
33          $\text{MOVE}(q_{curr}, targ_{dist}, targ_{head})$ ;      # Move to target location.
34     end if
35 end if
36  $[targ_{dist-prev}, targ_{head-prev}] = [targ_{dist}, targ_{head}]$ ;
37 while  $q_{curr} \neq \text{THRESH}(q_{init})$  do Repeat lines 2-40;

```

Figure 5.4: Pseudo-code for the exterior navigational algorithm [3].

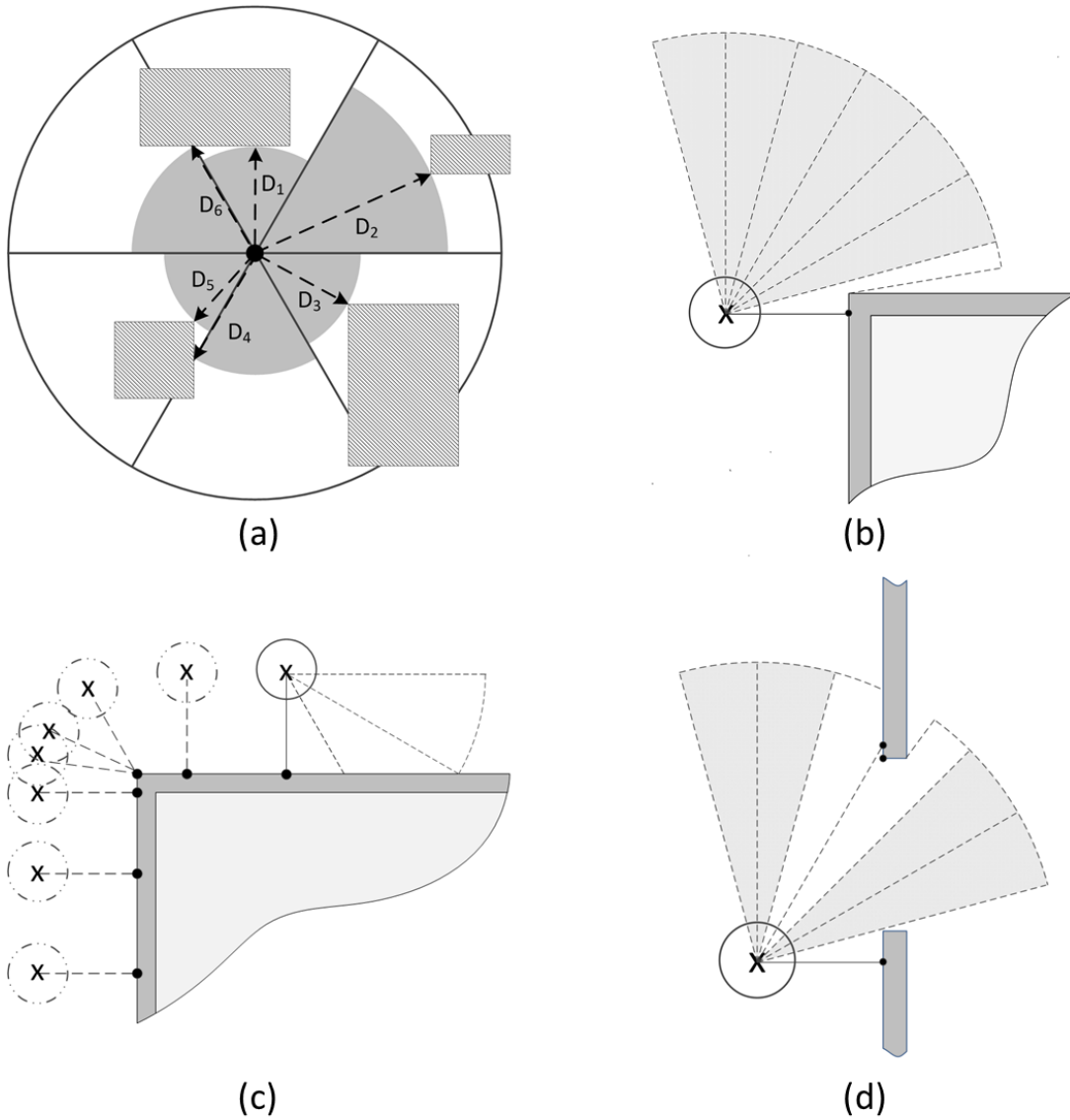


Figure 5.5: (a) How the robot divides up the perceived environment. (b) An example of the robot reaching an exterior corner. (c) An example of the robot navigating an exterior corner. (d) An example of the robot determining there is a traversable gap [8].

Figure 5.6 shows results from the simulation of the exterior navigational algorithm. Again, the UAV is shown as a blue star shape, its path as a thin green line, obstacles as black, open space as white, and unknown territory as grey. The simulations show the UAV

successfully navigate convex and concave features as well as open gaps in the perimeter. The work to implement the algorithms on the experimental robotic platforms is currently being performed.

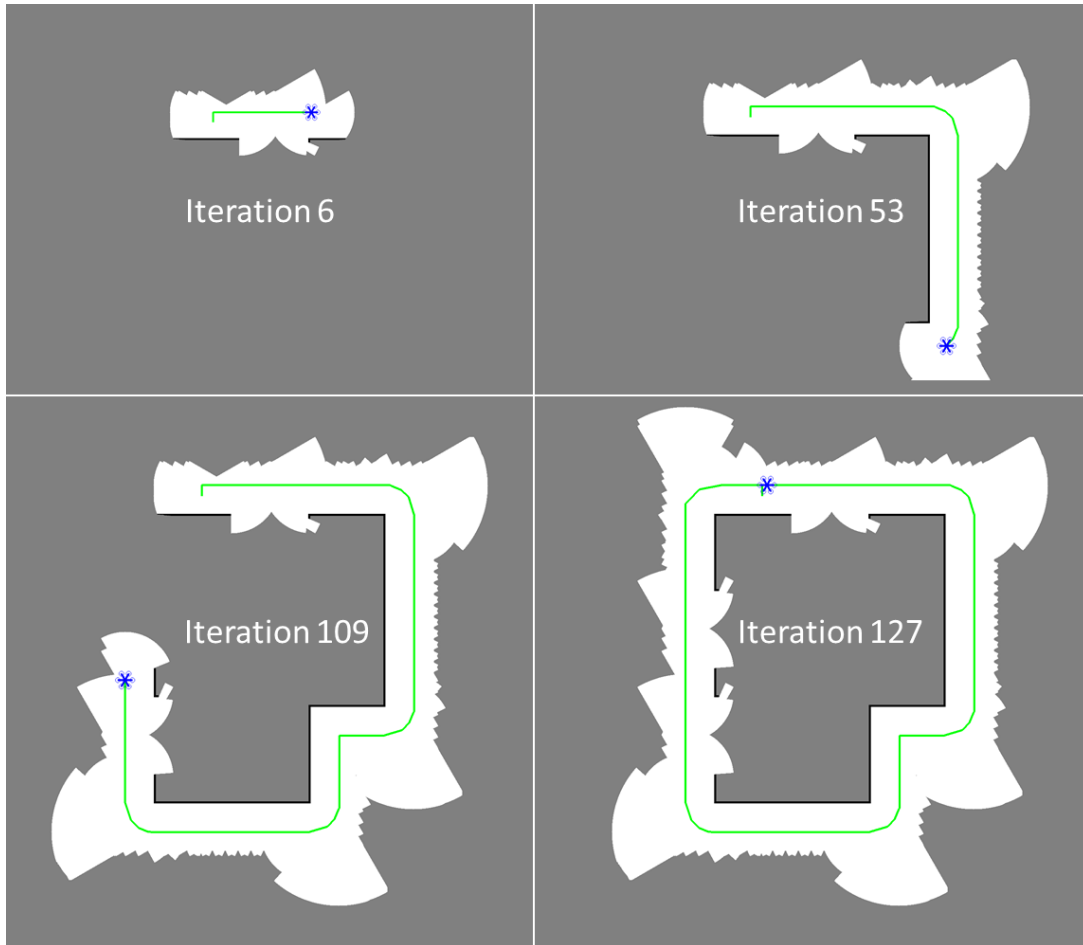


Figure 5.6: Results from a simulation of the exterior navigational algorithm [3]. The UAV is shown as the star shape in blue, its path as the thin green line, obstacles as black, open space as white, and unknown territory as grey.

5.4.2 Lighting Identification and Analysis

Lighting is a key consumer of electricity in buildings. In the U.S., approximately 12%, or 765 TWh, of the total electricity consumption is due to lighting [83]. Other estimates of the global electricity usage put lighting at 20% [84]. Having such widespread use, there is constant development of new and more efficient lighting technologies. Upgrades and changes to building lighting are also easier to implement than other building upgrades. Consequently, lighting is one of the first assessment tasks that auditors consider. Examining the Industrial Assessment Center database [85], which contains over 16,000 assessments across 40 years, shows that of the over 120,000 recommendations made, more than 20% pertain to lighting. As a key consumer in building electricity and a frequent recommendation given by auditors, lighting assessments were a logical technology to develop first for AuRAE.

To correctly identify and analyze lights, several sensors are needed. The new lighting sensing package for AuRAE contains a USB camera, a light level sensor, a USB laser rangefinder, and a spectrometer. The USB camera captures images of the ceiling. These images are used to identify the presence of lights, their shape, and their location. The light level sensor records lighting information to be used later in analysis and simulation. The USB laser rangefinder is key in determining the locations of the lights relative to the robot. The spectrometer is used to analyze the lighting spectrum to determine the type of bulb in use. The lighting identification process starts by calibrating the USB camera to remove any distortion inherent in the camera lens. This process estimates the parameters of the lens and image sensor of the camera, including camera rectification, projection, and distortion. By saving these parameters to a file, the camera image can be transformed to negate this undesirable lens distortion. This allows the method to process the output of the camera image stream using the pinhole camera model [103]. Once calibrated, the images from the

USB camera are converted to gray scale format. The images then undergo a thresholding process using Otsu's method [104], reducing it to a binary image. This method of thresholding adaptively processes the image to minimize the variance in bimodal histograms. As bulbs are typically brighter than the remainder of the image, Otsu thresholding works well for identifying the lights. After a morphological process to remove outliers and close gaps, the lights can then be identified in the image using a contour detection feature provided by OpenCV, an open-source computer vision library. The function provides the object's mid-point, bounding rectangle, and area, among other values. Since Otsu's method assumes a bimodal distribution, the method will falsely identify objects in the photos as lights when lights are not present. As such, an additional check of the initial image's brightness values is used to determine the presence of lights. The contour function from OpenCV can also be used to detect bulbs, requiring tighter tolerances and thresholds than for the general light shape. The results of this general process can be seen in Figure 5.7.

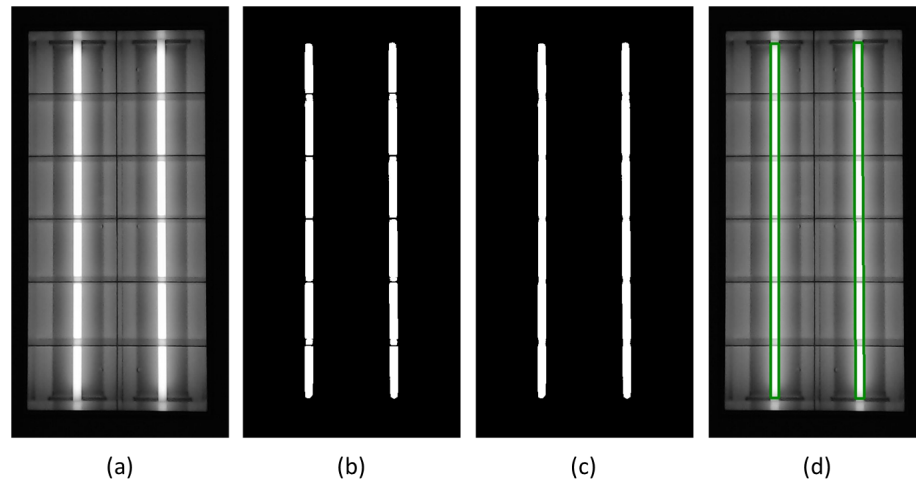


Figure 5.7: (a) The initial image of a light captured by the USB camera. (b) The thresholded image after applying Otsu's method. (c) The image after undergoing morphological operations. (d) Individual bulbs being identified using the contour detection function from OpenCV.

After receiving the relative midpoint coordinates of the light from the contour detection function, the global position of the light in three dimensional space can be determined using inverse camera projection algorithms. Because the camera captures 2D images, some of the 3D position information of the lights is lost. Using a camera coordinate system and the pinhole camera model, the 2D pixels of the image can be projected as 3D rays in space. Once the pixels have been projected as rays in 3D space, the rays can be transformed from the camera coordinate system to the global coordinate system. This transformation involves both rotation and translation. The transformation was performed using Tait-Bryan angles defined as roll (θ_R), pitch (θ_P), and yaw (θ_Y). As the rotation data provided by Hector SLAM is in the quaternion format (q_0, q_1, q_2, q_3), it must be converted to Tait-Bryan angles by using the following formula [105]:

$$\begin{bmatrix} \theta_R \\ \theta_P \\ \theta_Y \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{2(q_0q_1+q_2q_3)}{1-2(q_1^2+q_2^2)}\right) \\ \arcsin(2(q_0q_1+q_3q_1)) \\ \arctan\left(\frac{2(q_0q_3+q_1q_2)}{1-2(q_2^2+q_3^2)}\right) \end{bmatrix} \quad (5.1)$$

With the coordinate system defined for the camera above, the rotation matrices for roll $R(\theta_R)$, pitch $P(\theta_P)$, and yaw $Y(\theta_Y)$, are defined as follows: roll is rotation about the x-axis, pitch is rotation on y-axis, and yaw is rotation on the z-axis. With these definitions, the rotation matrix for each angle is defined by the following:

$$R(\theta_R) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_R) & -\sin(\theta_R) \\ 0 & \sin(\theta_R) & \cos(\theta_R) \end{bmatrix} \quad (5.2)$$

$$P(\theta_P) = \begin{bmatrix} \cos(\theta_P) & 0 & \sin(\theta_P) \\ 0 & 1 & 0 \\ -\sin(\theta_P) & 0 & \cos(\theta_P) \end{bmatrix} \quad (5.3)$$

$$Y(\theta_Y) = \begin{bmatrix} \cos(\theta_Y) & -\sin(\theta_Y) & 0 \\ \sin(\theta_Y) & \cos(\theta_Y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

The order that these rotations are performed is important. From the definition of Tait-Bryan angles, in order to rotate objects from the camera coordinate system to the global coordinate system, roll rotation must be done first, then pitch, and finally yaw [105]. After these rotations are performed, the object can then be translated to the global coordinate system by adding the robot's position (X_o) to the rotated object.

$$X_{global} = Y(\theta_Y)P(\theta_P)R(\theta_R)(X_{camera} + X_o) \quad (5.5)$$

With this formula, the equation to transform objects from the global coordinate system to the camera coordinate system is as follows:

$$X_{camera} = R(-\theta_R)P(-\theta_P)Y(-\theta_Y)(X_{global} - X_o) \quad (5.6)$$

Note that the equation above was simplified by realizing that the inverse of a rotation matrix is found by negating the rotation angle. With these equations, any geometric object can be described in the camera coordinate system to then be transformed into the global coordinate system and vice versa. In particular, the rays generated from the 2D light location can be transformed into the global coordinate system.

To find the light's 3D light position, a distance sensor measuring the robot's distance

to the ceiling was measured. This distance reading was used to create a ceiling plane at the height measured. With this, the 3D light location was calculated by finding the light ray's intersection with the ceiling plane. This was done by solving the following system of equations:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \\ z_y \end{bmatrix} t \quad (5.7)$$

$$ax + by + cz = d$$

The top part of Equation 5.7 represents the line while the bottom part represents the plane. The solution to Equation 5.7 is a point in 3D space which is the location of the light.

As lights seen in the image are blobs instead of single pixel points in the image, inaccurate light positioning may occur when these lights are seen on the edge of the camera image. This is because the light detection algorithm does not account for lights that extend out of the image's field of view. The light detection treats the midpoint of the light as the midpoint seen in the image. To mitigate this position distortion, lights on the edge of the image are marked, noting that the full light fixture has not been seen by the camera image. After the full light fixture is in the camera's field of view, the midpoint of the light is recalculated and updated in the map's display. To alleviate any errors in inaccurate pose data from Hector SLAM, lights previously found by the light location algorithm are continuously compared with lights seen in the camera's field of view. With this, lights are corrected in the map as updated information becomes available. An additional step to remove false positives due to inaccurate position data is to compare the distance of new lights to existing lights. If this distance is found to be within a specified threshold, the newly found light is disregarded and considered to be the same as a light that was previously found.

Once a light is identified and located, the light's center coordinates are communicated to the navigational algorithm, guiding the robot to the center of the light. Once directly under the light, spectrometer readings are taken to determine the light type. Spectrometer measurements are compared to reference data for different types of bulbs. After normalizing the incoming light spectrum, it is matched to different spectral curves to identify the light type with the closest match. Bulbs are then associated with their respective light and all data (position, bulb size, light type, lighting level, light ID) are written to a file.

After all the data are collected, lighting maps and simulations are generated. Results from a previous lighting simulation are shown in Figure 5.8. The simulation of illuminance levels is important in determining if the proper lighting levels exist for the tasks performed in the space. Recommendations for lighting levels are published by the Illumination Engineering Society (IES) [106]. These are only guidelines and are not always achieved during initial construction of a building or during remodeling. Under lit spaces can create hazards or a stressful environment for occupants, while over-lit areas are consume excess energy. The details of light level simulation are found in [9].

A novel lighting package prototype has been developed and installed on a ground robot platform, as shown in Figure 5.9. The ground robot uses Hector SLAM and a 2D scanning laser rangefinder to generate a map and locate the robot's position. The lighting package identifies ceiling lights in real-time as the robot navigates a space. Currently the ground robot is controlled by a user via a base station computer or mobile tablet. In place of the navigational strategy informing the lighting package to take a measurement upon arriving at the light location, the user currently sends a command to record lighting data. A ground robot is being used initially to aid in the safe testing of the lighting package prototype and the autonomous navigation algorithm. The finished lighting package is intended to be installed on a UAV which will enable illuminance level measurements at the correct height.

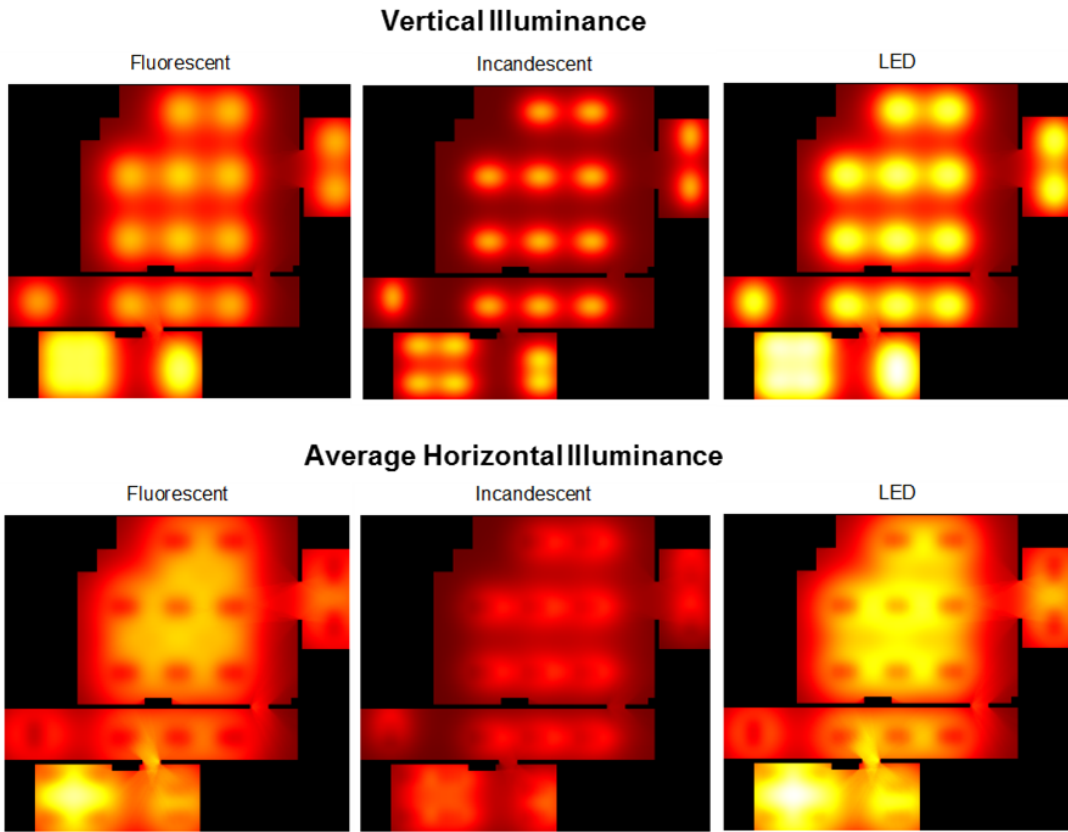


Figure 5.8: Results from lighting simulations showing illuminance levels depending on bulb type [9].

Preliminary experiments with the ground robot platform have been performed, with novel results shown in Figure 5.10. The figure shows real light detection and facility map generation using the methods described in this paper. For reference, the black shown in the images represents obstacles, the light gray represents open space, the medium grey represents unknown space, and the dark grey represents space outside of the mapping domain. The yellow spheres represent the centers of identified lights and the faint, green line shows the ground robots path. Using a 2D scanning laser rangefinder, Hector SLAM produces a 2D map of the building. The lighting package is able to determine the height of the lights with its laser rangefinder. Overall, the results are accurate and very promising.

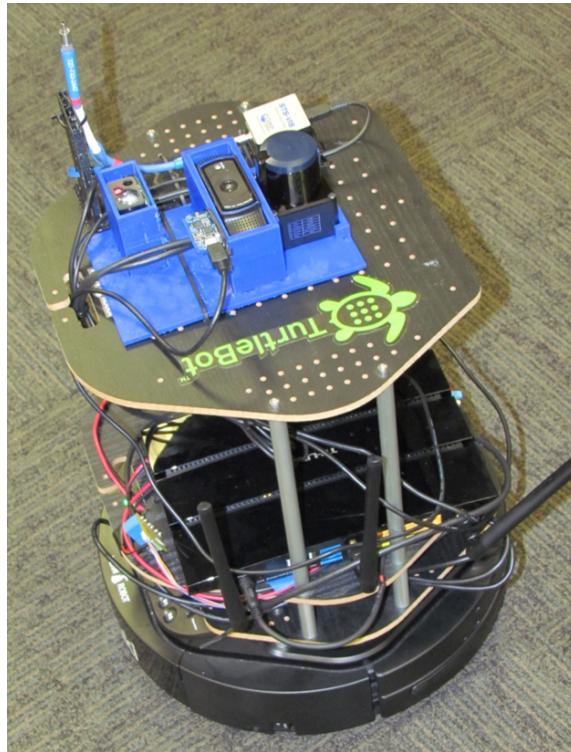


Figure 5.9: Ground robot platform prototype with lighting identification package installed.

The preliminary tests were performed in a library. The rows of obstacles observed in Figure 5.10 are the book shelves. In the lower part of the image, the lights can be seen to be in two straight rows along the wider path way between two sections of the library. Some variance in the lighting locations can be seen above the book shelves. Because the shelf height is so close to the ceiling that parts of the lights were actually obstructed from the view of the ground robot, causing the robot to only identify the center of the visible light. The size of the library and the number of lights show that the developed methods are able to handle larger commercial and industrial spaces. Additional tests are currently being performed across various locations.

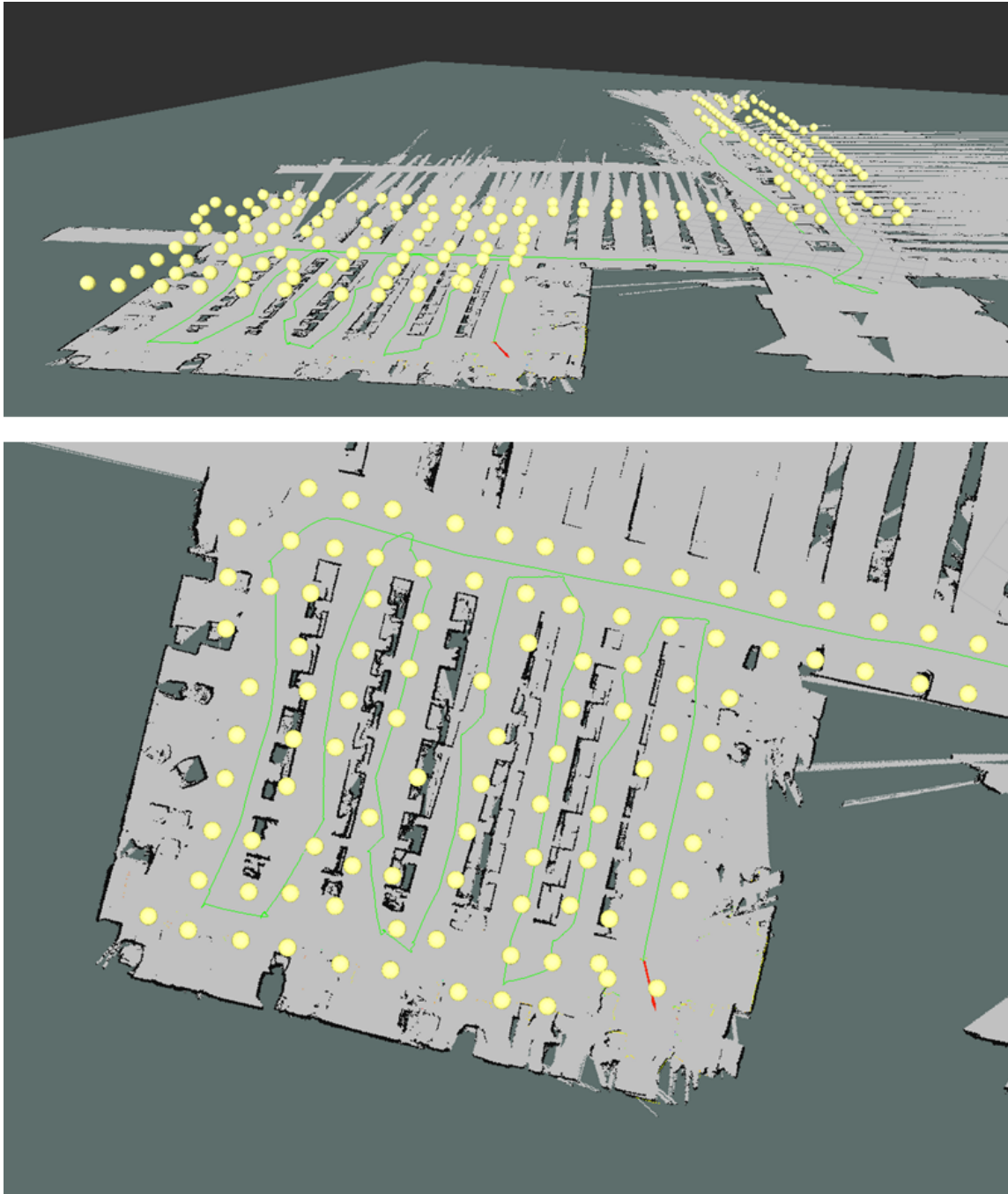


Figure 5.10: Preliminary results from the ground robot platform and lighting package prototype. The yellow spheres represent the centers of lights identified in the building (large campus library).

5.4.3 Baseline Environment Data Collection

Establishing an environment baseline of the site being assessed is useful in ensuring that comfort and health guidelines and regulations are being met. The collected data can also be used in the fault detection of existing building sensors. Duct balancing problems can be found and addressed, as well as stratification, poor ventilation, and other location dependent problems. An environment baseline consists of measurements of temperature, relative humidity, CO₂ levels, and lighting levels throughout the building. Additionally, the data gathered provide key information for performing the more advanced energy simulations and analysis that come with more comprehensive audits. By freeing the technician from having to collect these measurements, their time can be spent elsewhere interviewing staff or assessing other equipment.

To gather the required data, an environment baseline package prototype has been developed. It consists of a microcontroller, combined temperature and relative humidity sensor, a CO₂ sensor, and a light level sensor. Data is written to a SD memory card for easy access during the analysis phase. In order to gain the best representation of the building environment while working to automate the process, a UAV platform was chosen for the sensing prototype. The UAV allows the package to take measurements at various heights and locations throughout the building. One challenge of using the sensing package on the UAV platform is determining the UAV's 3-dimensional position as it moves through the building. In order to make informed decisions based on the collected data, the physical location of the data measurement is needed. For the baseline environment data collection prototype, this issue was solved by using a Google Project Tango tablet. The tablet is equipped with an advanced accelerometer and IR depth camera that enables 3D sensing of the surrounding environment. Although Hector SLAM works well for the ground robot with its 2D map, Hector SLAM is not immediately suited for generating a 3D map. As

such, Real-Time Appearance-Based Mapping, or RTAB-Map, is used [86]. RTAB-Map is a RGB-D Graph SLAM approach that uses a Bayesian loop closure detector. The loop closure detector is able to determine the probability that a newly received image comes from a previous location or a new location. When a loop closure is detected, a new constraint is added to the map's graph. An optimizer then minimizes errors in the map. The number of loop closures is limited using a memory management approach to ensure that real-time constraints on large-scale environments are met. More details can be found in [86]. In order to pair the measured data with the UAV's 3D coordinates, a timestamp with each measurement is recorded on the SD memory card at a rate of 120 Hz. This timestamp is then correlated with the closest matching coordinate timestamp. RTAB-Map saves physical coordinates at a rate of 60 Hz. The data collection and position recording can be integrated into the same platform in future iterations. The UAV platform with the sensing package prototype can be seen in Figure 5.11.

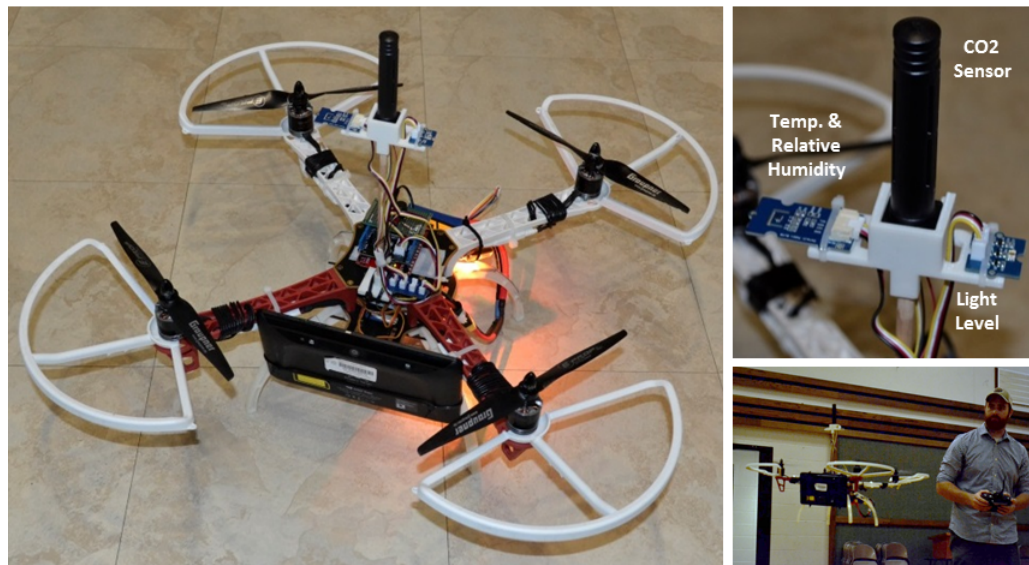


Figure 5.11: The UAV platform and sensing package prototype used to collect baseline environment measurements.

Preliminary results from taking measurements from a local church are shown in Figure 5.12. The UAV was piloted manually. Figure 5.12(a) shows an oblique view of the areas of the church that were mapped. The UAV was maintained at roughly the same height through the hallways and classrooms; thus, there was little temperature variation in the vertical direction. However, in the gym, the UAV performed multiple passes at different heights. Even with the approximately 20 foot ceiling, the temperature and humidity showed very little variation. Figure 5.12(b) shows an overlay of interpolated CO₂ concentrations throughout the building. Not all areas of the building were visited, as is visible by the trajectory of the UAV shown in red. The highest levels appear in the gym, near the divider that was partially open. This could have resulted from the researchers' extended time in this area during data collection. To verify the 3D position determination and mapping algorithms, and the multi-sensor system, the UAV was piloted manually in this example.

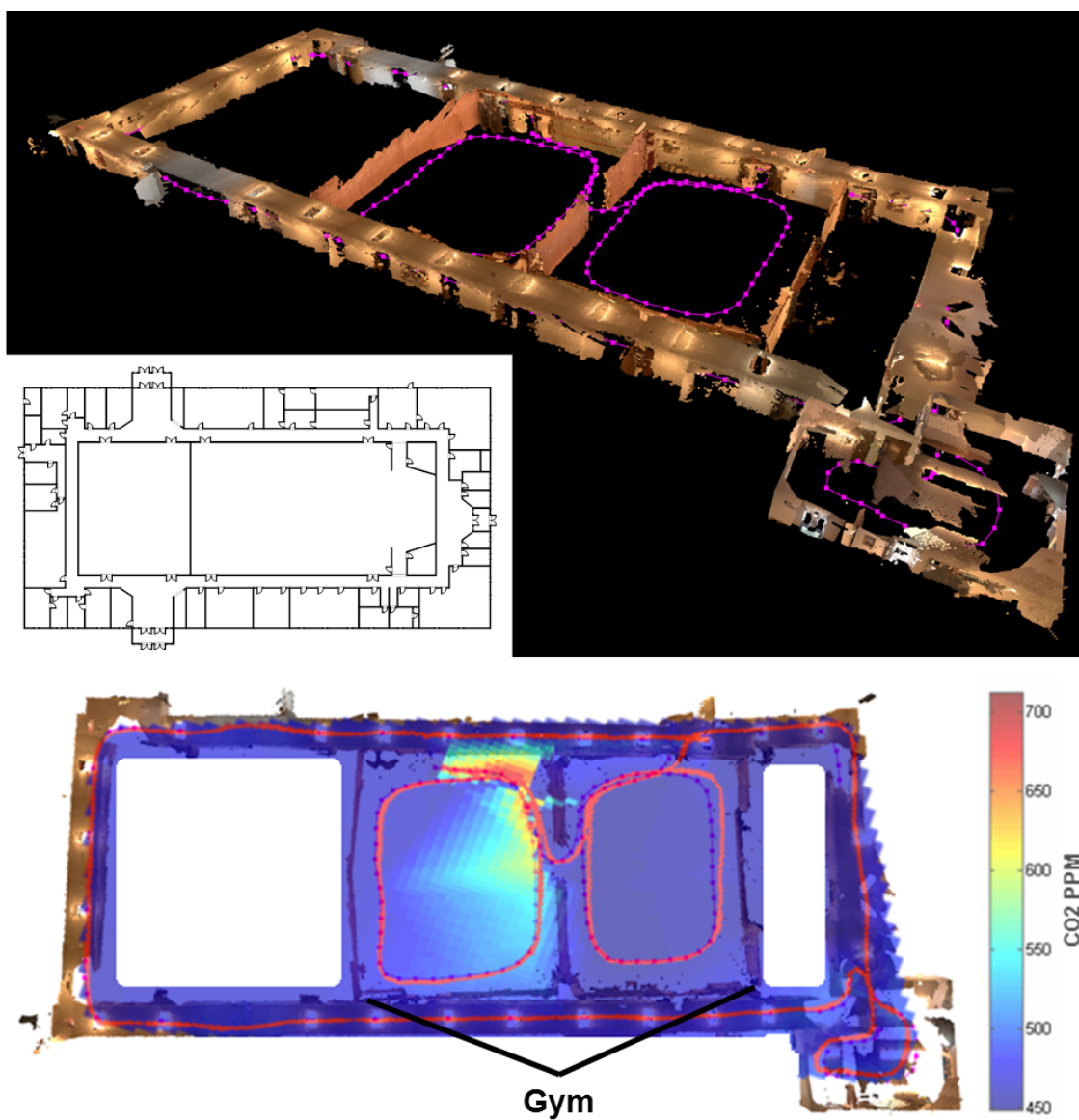


Figure 5.12: Preliminary results from the environment sensing package.

5.5 Open Challenges

While there has been much progress in the development of the AuRAE platform, there are several areas of open research within the system. Those challenges and select future possibilities are discussed below.

5.5.1 Identification of Fenestration

A task that could be automated by the AuRAE platform that is often difficult and time intensive for auditors to complete is the inspection of ducting and fenestration. Duct leaks create inefficiencies in the heating and cooling of a building. Air handling units operate more to provide the same cooling to a space, resulting in more energy consumption and greater wear on the equipment. Leaks can also prevent zones from receiving adequate heating or cooling, causing productivity losses or uncomfortable conditions for occupants. A sensing package could be developed that uses an IR thermal camera to examine the exterior of ducts, identifying large temperature gradients as potential leak points. Another possibility is for a small UAV or climbing robot to maneuver within ducts, looking for penetrating light sources to identify leaks. This same package could be used to identify insulation leaks around windows and doors, or large temperature gradients through existing walls. There is current research that has produced a Lidar-on-a-chip package [?]. These Lidar chips are orders of magnitude smaller, lighter, and cheaper than the Lidar systems that are currently available, as well as being able to scan at 1000 times the current rates with no moving parts. This new technology would further enable the development of micro-robots that have the capabilities to generate maps and navigate through ducting or other constricted environments.

5.5.2 Occupant Enumeration and Location

In addition to identifying fenestration and ventilation systems, occupant enumeration and location data are important for energy modeling purposes. Occupant data can ensure that occupancy and heat loads are accurate. Furthermore, the information gathered can be used to ensure proper emergency egress plans are in place. With the more advanced platforms that will be released over the next several years, devices could be used in actual emergency situations to scan crowds of people and the building areas, identifying occupants, or searching for people within the building and relaying their location to emergency response personnel. This could be accomplished with facial recognition technologies or embedded sensors in employee IDs.

5.5.3 External Mapping and Identification

Beyond interior measurements, the mapping of the exterior of a building as well as identification of external objects and characteristics are of great interest. As previously mentioned, infrared images of the exterior of a building are very useful in determining insulation and fenestration leaks. In addition to infrared data, the reflectivity of the roof's materials and other exterior elements can be used to improve energy simulations, influencing how much solar irradiance is absorbed by a building. Mapping the exterior dimensions of the building are also necessary to create accurate building models to be imported into energy simulation software, as discussed below. With the continuing reduction in size of sensors as well as increase in available computing power, solutions for these tasks can be developed over the coming years.

5.5.4 Cooperative Autonomous Audits

Preliminary work thus far has been completed with individual robots; however, fully cooperative fleets are the final vision of the AuRAE system. Coordinating UAVs to work

as a team to scan a building is much more efficient than a using lone platform performing the same tasks. Additionally, specialized units could be developed, thereby producing savings by only having to purchase limited quantities of specialized sensors, such as IR thermal cameras and spectrometers. A group of micro-UAVs could quickly explore and map the building, taking temperature, relative humidity, and light level measurements as they move. Upon completion, this generated map could be utilized by the specialist robot to navigate around the facility and identify target objects more efficiently. These aerial devices would be supported by ground vehicles, providing adequate communication relays, the necessary computational power, and ability to recharge as needed. Cooperative exploration and navigation strategies need to be investigated and developed to implement this part of the AuRAE platform.

5.5.5 Automated Energy Simulation Software Support

In the automated modeling area of research, some work has started creating 3D models of buildings through robotic measurements and importing those models into energy simulation software [30, 29, 31]; however, a complete solution does not yet exist. An additional potential ability of AuRAE is to be able to have a robot completely map a building, identify the relevant energy features for the simulation, process the map to create an accurate 3D model in a proper format, and then import this model into the simulation software. Challenges include generating smooth, accurate models of the buildings from the laser and IR depth scanning technologies available today. Also, a solution is needed for translating this model into the correct format for the energy simulation software, complete with locations, sizing, and other relevant information about the energy objects (air ducts, terminal boxes, windows, doors, etc.).

5.5.6 Automated Report and Recommendation Generation

Another significant time-sink for auditors is compiling the report that details recommendations for the client. An automated solution to identify recommendations and generate a comprehensive report with the relevant information is an open challenge. This would require the data to be collected and written in a standard format so that it could be processed by a scripted program to perform the required analysis and ascertain the best recommendations. The report could then be generated, requiring minimal user input about the client (name, utility information, etc.). The program could be designed in such a way that the level of the audit could be specified, providing recommendations and the required implementation difficulties commensurate with the client's priorities and available resources. Recent advances in deep learning and neural network methods (see review article [107] for more information) that have driven the success of speech recognition technologies, visual object recognition, and even drug discovery can be leveraged to identify patterns in the gathered and simulated data. Over time, these patterns can be connected with appropriate recommendations, further alleviating the human intervention required, eventually working towards a fully automated report generation process.

5.6 Conclusion

Autonomous Robotic Assessments of Energy (AuRAE) have potential to greatly impact the energy audit industry. By reducing the time and training required to perform an audit, while also providing more dense and repeatable measurements and recommendations, AuRAE can greatly reduce the cost of the audit resulting in significant increases in audit market penetration. This paper has presented significant progress in developing the AuRAE platform. Specifically, interior and exterior navigational strategies have been created. A lighting identification and analysis package has been developed, and a prototype implemented on a ground robot has generated promising preliminary results.

Baseline environment data collection has been performed with a sensing package installed on a UAV that measures temperature, relative humidity, CO₂ concentrations, and lighting levels. Several open challenges still exist towards the completion of the AuRAE platform. These include identification of fenestration and ventilation systems, the location and enumeration of occupants, cooperative strategies to coordinate multiple robots, automated energy software support, and automated recommendation and report generation. Upon completion of this research, fully autonomous one-time audit fleets as well as continuous auditing groups that interface with a building's energy management system are envisioned, producing significant energy savings and cost reductions for businesses that would otherwise not receive an audit.

5.6.1 Contributors

Christopher J. Bay developed the navigational strategies, mentored the lighting identification and mapping student team, completed the research pertaining to the environment baseline measurement platform, composed the paper, and helped develop the project over several student team iterations.

Trevor J. Terrill completed lighting simulations and analysis, provided paper revisions, and helped develop the project over several student team iterations.

Bryan P. Rasmussen engendered the original idea, advised C. Bay, T. Terrill, and the student teams in their research, and provided paper revisions.

6. PAPER D: STEADY-STATE PREDICTIVE OPTIMAL CONTROL OF BUILDING ENERGY SYSTEMS USING A MIXED ECONOMIC AND OCCUPANT COMFORT FOCUSED OBJECTIVE FUNCTION

6.1 Synopsis

Control of energy systems in buildings is an area of increasing interest as the importance of energy efficiency and occupant health and comfort grows. The objective of this study is to demonstrate the effectiveness of a novel steady-state optimal control method in minimizing the economic costs associated with operating a building. Specifically, the cost of utility consumption and the cost of loss productivity due to occupant discomfort are minimized. This optimization is achieved through the use of steady-state predictions and component level economic objective functions. Specific objective functions are developed and linear models are identified from data collected from a building on Texas A&M University's campus. The building consists of multiple zones and is serviced by a variable air volume (VAV), chilled water air handling unit (AHU). The proposed control method is then co-simulated with MATLAB and EnergyPlus. Simulation results show improved comfort performance and decreased economic cost over the currently implemented building control, minimizing productivity loss and utility consumption. The potential for more serious consideration of the economic cost of occupant discomfort in building control design is discussed.

6.2 Introduction

As the global population has grown and modern technology has become available to more areas of the world, the use and consumption of natural resources has become a pertinent concern for current and future generations. Consider the area of energy. In the U.S. alone, total energy consumption has nearly tripled over the last 65 years from 34.6

quadrillions Btus (quads) in 1950 to 97.4 quads in 2015 [12]. Of the energy consumed in the U.S., non-renewable energies still represent over 90% of energy sources [12]. In response to growing energy needs and increased understanding of environmental impacts due to traditional energy sources, many nations have put forth specific renewable energy targets. These targets aim to reduce dependence on non-renewable energies and to maintain a competitive edge in the global energy technology market. For example, the European Union’s (EU) Renewable Energy Directive has established a goal of 20% final energy consumption from renewable sources by 2020 [13]. In the U.S., the Department of Energy has set a goal for the nation to have 20% of its electricity sourced from wind energy by the year 2030 [14]. While renewable sources will likely continue to increase over the coming years, reductions in energy usage can also play a significant role in alleviating the looming scarcity of natural resources and maintaining a competitive edge in the global energy technology market.

Focusing on the U.S., Americans consume 19% of the total world energy consumption, as shown in Figure 6.1. Delving into the energy consumption practices in the U.S., approximately 40% of all energy goes to building operations in the commercial and resi-

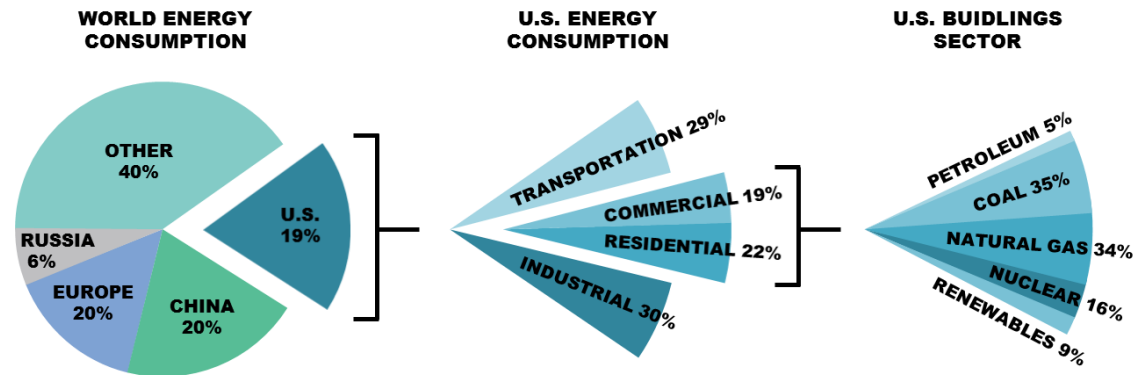


Figure 6.1: U.S. energy consumption and breakdown of energy used in different sectors [2].

dential sectors [15]. Breaking the energy used in the commercial and residential building sectors down by source (right side of Figure 6.1), the data shows that approximately 75% comes from fossil fuels. As a result, energy usage in buildings account for 40% of the total U.S. carbon emissions [16]. Figure 6.2 shows that the buildings' share of U.S. energy consumption has increased from approximately 34% in 1980 to the approximately 40% it is now (as of 2010). Not only has it increased, but it is projected to continue growing over the next 20 years. Looking closer, Figure 6.2 shows that the commercial sector specifically has shown increased consumption percentage over time while the residential share has remained relatively constant. From Figure 6.2 the data shows that the commercial buildings sector, at approximately 20%, consumes nearly 20 quads (20 quadrillion BTUs).

Investigating the commercial buildings sector further, the commercial site energy consumption can be broken down into specific end uses, as shown in Figure 6.3. Examining

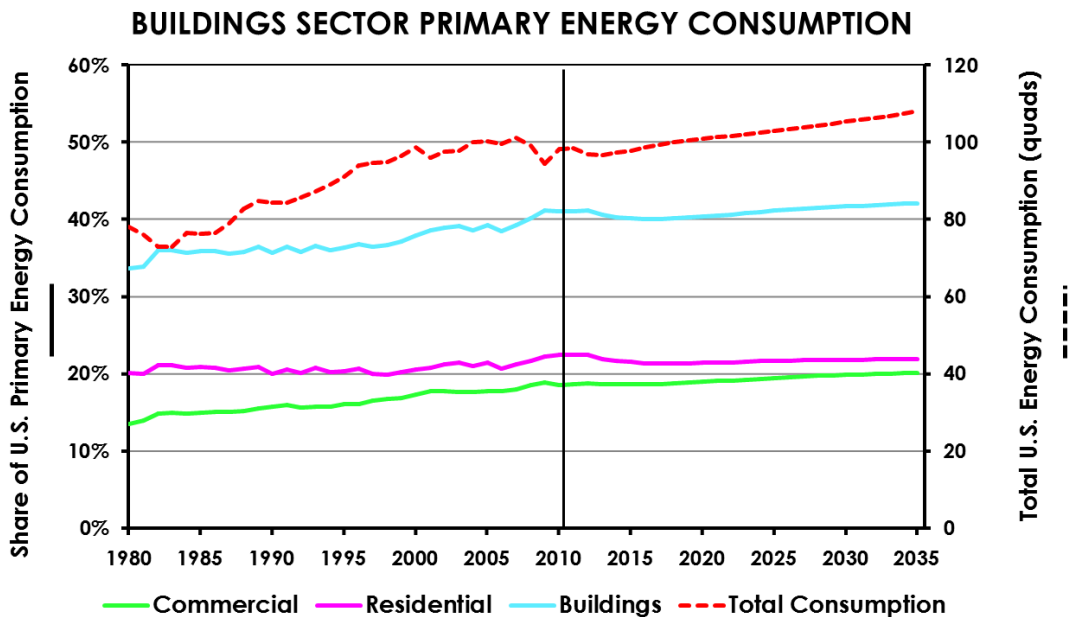


Figure 6.2: Past and projected primary energy consumption for the U.S. buildings sector [2]. The vertical line represents when the data was collected.

the top end uses reveals that space heating (27%), lighting (14%), space cooling (10%), water heating (7%), and ventilation (6%) are responsible for 64% of commercial building energy usage. These categories can be combined more generally to refer to services required mostly when a building is occupied (space conditioning and lighting). The smaller categories and all-encompassing “other” category represent 25%, while the other 11% is an adjustment that the Energy Information Administration uses to relieve discrepancies between data sources, specifically defined as “Energy attributable to the commercial buildings sector, but not directly to specific end-uses” [15]. To put this data in economic terms, in 2010 utilities cost businesses and building owners in the commercial sector \$179.4 billion. Not surprisingly, when this cost is broken down by use, the same five categories described at the beginning of this paragraph are at the top: lighting (\$35.4 billion), space heating (\$27.5 billion), space cooling (\$25.3 billion), ventilation (\$15.9 billion), and water

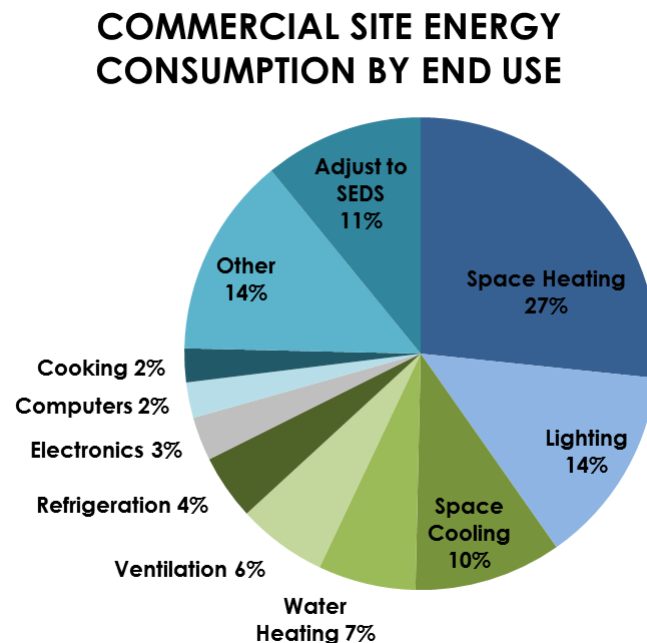


Figure 6.3: Site energy consumption by end use for U.S. commercial buildings [2].

heating (\$7.3 billion). Given the facts described above, buildings are a readily apparent and prime target for reductions in energy use and environmental impact.

Lately one of the major trends in building energy research is the area of advanced building controls. Current practice in building energy systems is to implement control with low-level controllers (proportional-integral or proportional-integral-derivative) in a decentralized fashion. In some cases, there may be some supervisory control, but mostly the systems operate independently. Because of the physically interconnected and complex nature of building systems, this uncoordinated control can often lead to inefficient solutions as controllers can compete with one another in achieving their desired outputs. Many advanced control strategies have been proposed, with Model Predictive Control (MPC) being a front-runner to address the challenges that building energy systems put forth. However, with MPC, the development of the objective function to be minimized is essential in the performance of the system. Of particular interest would be an objective function that accounts for the economic cost of utilities at the component level as well as the economic cost of occupant discomfort. Such an objective function would enable the identification of possible energy and cost savings, serving to guide building energy managers and researchers as to where their efforts for increases in performance and efficiency should be focused. This paper details the development of steady-state optimal control method based on data collected from a building on Texas A&M University's campus, compares the simulation of a standard control implementation versus the proposed supervisory controller, and examines the impact of including the cost of occupant discomfort in the control strategy. First, a background on recent efforts in control of building energy systems is presented, focusing on economic optimization. Then information about the building and development of the steady-state control method are given, followed by the simulation methods. Simulation results are then detailed, followed by a discussion of the importance of occupant comfort in building control strategies. The paper ends with a discussion of the study's outcome as

well as future work.

6.3 Background

Control of buildings presents several unique challenges. Buildings consist of numerous interconnected energy systems that affect and depend on one another. For example, in a large building there may be multiple chillers that are used to chill a secondary fluid, such as water. This chilled water is then pumped to various systems and areas of buildings where heat exchangers in air handling units (AHUs) use the chilled water to cool air streams. A network of fans and ducts then deliver the cooled air to the desired locations. The flow of this cooled air into the zones can be controlled by variable air volume (VAV) units, in which there may be another heat exchanger that utilizes heated water to warm the air, if necessary. The heated water for this process is provided by a different set of centralized pumps and heat exchangers. The zones themselves are connected to one another, either by conduction through barriers or shared doorways/open spaces. All these interconnections and couplings result in coordinated control problems.

Providing additional difficulties, nonlinear dynamics evolving over multiple time scales occur across these various building systems. While changes in damper position in a VAV or fan speed in an AHU are relatively fast (on the order of seconds), changes in desired chilled water temperatures can take longer (minutes), or changes in room air temperature can be even slower (hours). There are also slow, overarching changes to take into consideration such as the shift in solar loads as the sun moves throughout the day, gradual changes in outdoor air temperature due to change in weather or seasons, and the slow deterioration of equipment through use over time. These all contribute to shifting systems behaviors and disturbances that can cause undesired performance in systems. There are also discrete changes to take into account, such as whether an area/room is occupied, how many people are in said room, or changes in real-time pricing of utilities. In addition, the sensors of the

system are distributed (not always equally), centralized monitoring is being performed, and devices are driven by localized controls. All of this occurs within constraints, either due to hardware limitations, limited resources, or issues of health and comfort.

While the challenges of building control are numerous, one control method that has emerged as a capable solution is model predictive control (MPC). MPC has been chosen as the most appropriate control method in various building thermal control research projects such as Opti-Control in Switzerland [34], Intelligent Buildings and Rational Management of Renewable Energy (MIGRER) in France [35], and MPC for UC Merced Campus in the U.S. [36]. MPC, or receding horizon control, predicts the change in the dependent variables of a modeled system by changing independent variables. Using the current state information, dynamic models of the system, and an objective function, MPC will determine the changes in the independent variables that will minimize the user-defined objective function while honoring given constraints on both dependent and independent variables. Once this series of changes is determined, the controller will apply the first determined control action, and then repeat the calculations for the next time step. Figure 6.4 displays how a typical reference tracking MPC implementation would behave. It can be seen at time k that the controller determines what the predicted output would be along with its optimal control trajectory. After completing the computations, the control would be applied and the system would move on to time $k + 1$, repeating the predictions and optimization with the new measurements. Of importance are the two horizons within MPC: the prediction horizon, which is the length of time for which the system outputs are predicted, and the control horizon, which is the number of control inputs that are determined in the prediction computation. The prediction horizon is often limited by computational capabilities, while the control horizon is selected so that the system dynamics are allowed to diminish [37]. More details about MPC can be found in [38].

MPC applications for buildings has been studied mostly in simulation [39, 40, 41, 42,

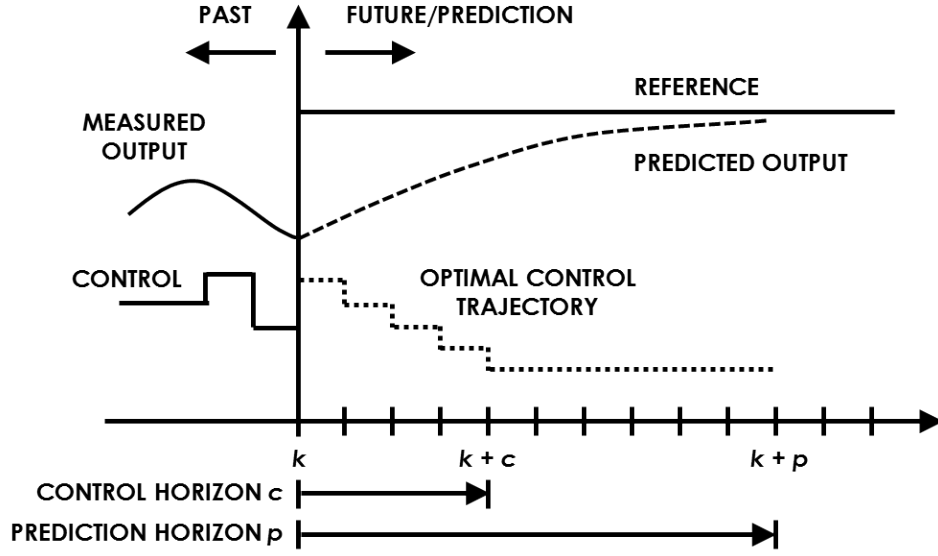


Figure 6.4: How MPC, or receding horizon control, typically functions.

43, 44, 45, 46, 47, 48], with a few experimental efforts [34, 35, 36, 49, 50, 51]. In simulation, MPC has been adapted for controlling building systems such as floor heating [52], water heating [53], cooling [36], and ventilation [54], among others. The most common types of cost functions include quadratic cost, linear cost, and probabilistic cost [40]. The forms of these functions are shown in Table 6.1.

For most applications of MPC for buildings that have been reported in the literature, the cost takes the form of economic MPC (E-MPC), where the objective function is a linear combination of the monetary cost of building energy consumption [43]. In these

Table 6.1: Common types and forms of MPC cost functions.

Cost Function Type	Mathematical Representation
Quadratic	$l_i(x_i, u_i) = x_i^T Q x_i + u_i^T R u_i$
Linear	$l_i(x_i, u_i) = c^T u_i$
Probabilistic	$l_i(x_i, u_i) = \mathbb{E}[q_i(x_i, u_i)]$

applications when E-MPC is used, the amount of energy consumed is being minimized while authors occasionally account for occupant comfort limits through constraints on the variables. In [46], a linear E-MPC cost function was used that had a time-varying cost of electricity vector for several actuators, which included positioning of blinds, the level of electrical lighting, chiller production, operation of the cooling tower, and heat from radiators. This cost function also included a scaling factor for the different actuators. Occupant comfort was maintained by constraints placed on the room temperatures and the lighting levels. This proposed control strategy is able to respond to real-time changes in utility pricing; however, occupant comfort becomes a second priority with only the ranges of temperature and lighting levels ensuring the control actions don't drift too far from the user-defined comfort zone, as opposed to actually optimizing occupant comfort. The experiment detailed in [49] had a similar cost function, except that the method optimized the cost of energy usage by manipulating different control variables: the temperature setpoint for the water coming from the cooling tower as well as the temperature and mass flow setpoints of the water coming from the chillers. Again, comfort is maintained within a range but not optimized. Papers [39, 41, 43, 50] all employed the linear E-MPC objective function while actuating heat flux, input power to a heat pump compressor, indoor air temperature setpoint and a thermal energy storage system, and temperature setpoints, respectively. The four previous works place constraints on the MPC optimization with temperature limits relating to comfort ranges, placing the optimization focus on control action and utility price. Oldewurtel performed simulations on six different combinations of subsystems including actuated blinds, electric lighting, radiators, mechanical ventilation, floor heating, evaporative cooling, and chilled ceilings, focusing on stochastic MPC while only ensuring comfort through zone temperature limits [44]. Corbin, *et al.* [45] detailed two case studies in which the first used a linear E-MPC cost function with comfort temperature limits while the second minimized the sum of electricity used by all HVAC

equipment with a comfort penalty. This comfort penalty was defined as an area-weighted sum of the number of zone occupied hours outside of a predicted mean vote (PMV) threshold of ± 0.5 . PMV is an index that determines the thermal comfort of an average individual dependent upon a variety of factors, including air temperature, relative humidity, relative air velocity, metabolic rate, clothing insulation level, work output, and several other variables [4]. Corbin, *et al.* worked to optimize occupant comfort in their second case-study, but do so with the focus of reducing the cost of energy usage, neglecting the economic aspect associated with occupant comfort and productivity.

The authors of [47] also included a discomfort cost with their monetary energy cost that was based on different lower and upper thermal limits; however, the physical meaning of this discomfort cost is arbitrary as the cost increases to unity until the temperature limits are exceeded and then becomes significantly large, not following any physical or measured relationship. The cost function in [48] included regulation of occupant comfort based on PMV, though it took the quadratic cost form, with the first term being the difference between the predicted PMV of the zone and the PMV setpoint for the zone, quantity squared, multiplied by a weighting factor. The second term consisted of the square of the change in control action, or increment, multiplied by a weighting factor. With this form, the MPC will balance maintaining the desired zone PMV while limiting large control action rates, in this case changes in water flow and air velocity. This will help keep occupants comfortable but the economic cost of the control actions is not accounted for. Morosan, *et al.* [42] used a linear objective function that penalizes the error between the predicted room temperature and the future room temperature reference as well as the energy usage to condition the room. In their efforts, comfort is accounted for as a comfort index that acts as a penalty when the room temperature does not meet its setpoint. However, in the presented simulations the temperature setpoints are arbitrarily chosen and not dependent on any comfort information. Additionally, this method, like the previous ones mentioned,

does not account for the economic aspect of occupant comfort. One objective function from the literature that appears more unique than others was used in [51]. This objective function consisted of three different linear terms: 1) a weighting coefficient multiplied by the predicted percentage of dissatisfied (PPD) people, which PPD can be calculated from PMV, 2) a weighting coefficient multiplied by the summation of cost of energy consumed by the heating and cooling devices, and 3) a weighting coefficient multiplied by the summation of the green house gas intensities of the various energy sources (electricity and natural gas). This objective function displays the power of MPC to determine optimal control actions with respect to a user's desired metrics, in this case occupant comfort, monetary cost of energy, and environmental impact of energy sources. While providing great flexibility in allowing the building operator to prioritize the three metrics with the weighting factors, the respective economic impact of the three areas is not represented in the objective function due to differing units and arbitrary weights.

Overall, previous methods have accounted for the economic cost of energy usage and/or attempted to maintain occupant comfort through optimization constraints or optimizing comfort itself, but none have accounted for the economic aspect of comfort on occupant productivity alongside utility costs. This research effort aims to develop a novel control method that optimizes a building's economic cost, due to both the consumption of utilities and the economic cost of loss of productivity due to occupant discomfort. Through analysis of the performance of this control method, the authors intend to identify areas having the most potential for savings and guide what the priorities of building managers and researchers should be for future work.

6.4 Development of Economic Objective Function for Advanced Building Systems Control

Considering the literature and previous works, a general component-level objective function of the form shown in Equation 6.1 was chosen. The quadratic terms (first and third) were included to provide for standard convex optimization when desired, where e represents the error for the system, Q the weighting placed upon the error, u the control action, and S the weighting placed on control action. The linear terms (second and fourth) were included to facilitate calculating cost purely in economic terms (dollars), as opposed to nonsensical units such as dollars squared. R and T can be formulated in such a manner to transform e and u into economic cost, as will be demonstrated in the subsequent sections.

$$J_{component} = e^T Q e + e^T R + u^T S u + u^T T \quad (6.1)$$

6.4.1 The Utilities Business Office

Working with the Utilities Energy Management (UEM) Office at Texas A&M University, limited access was granted to the Utilities Business Office (UBO) through the building energy management system for the purpose of collecting data and eventual implementation of advanced controllers. As such, individual component objective functions were developed for specific equipment in the UBO; however, the UBO represents a typical office building and thus the work can be generalized to other commercial buildings. What follows is a description of the UBO and its behavior to inform the development of the component objective functions. The UBO is a rectangular, single-story building consisting of 11 zones, 10 of which are actively controlled. The general layout can be seen in Figure 6.5. In this initial development the decision was made to focus solely on the cooling aspects of the system because: 1) due to the location and climate of the building,

the majority of the year is spent in cooling mode, and 2) simplified operating conditions will help to validate the developed control strategy in its first implementation. The process flow and current control implementation is displayed in Figure 6.6. The rest of this section details each of the subsystems and currently implemented controls.

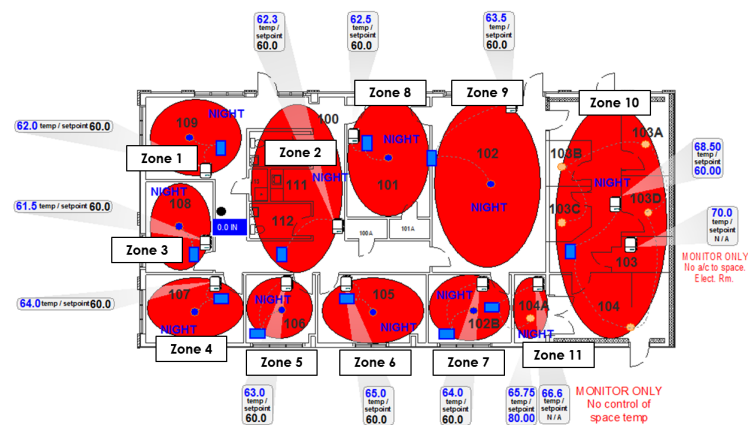


Figure 6.5: Zone layout for the Utilities Business Office (UBO) at Texas A&M University.

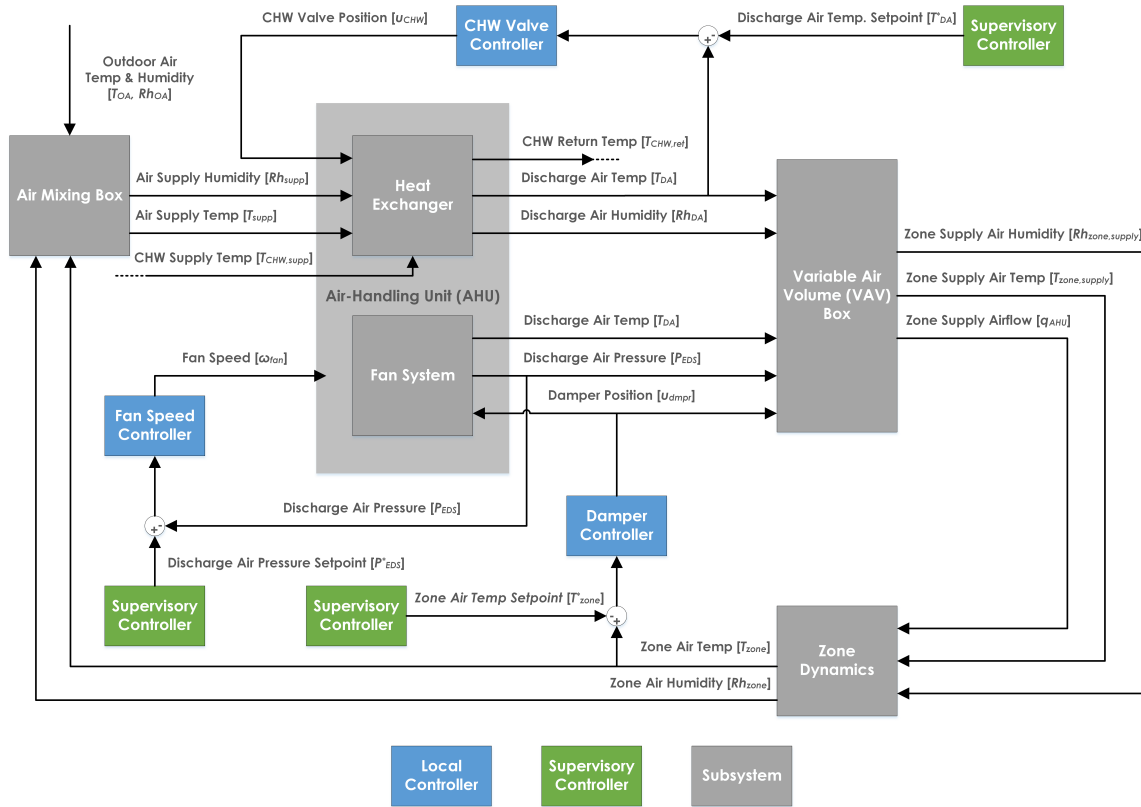


Figure 6.6: Process flow and current control implementation for the UBO.

The UBO is serviced by a single, rooftop air handling unit (AHU). The AHU consists of a variable air volume (VAV) fan, a chilled water coil, an outdoor air damper, a return air damper, a discharge air temperature sensor, and an end static pressure sensor. The organization of these components can be seen in Figure 6.7. During normal operation, the VAV fan works to maintain an end static pressure given by a pressure demand calculation dependent upon the individual damper positions of the zone terminal boxes, moving the air through the system. The pressure demand calculation is the feedback signal for a proportional-integral-derivative (PID) loop that produces a reference equal to the desired end static pressure. This reference is fed to another PID loop that actuates the fan speed to maintain the given end static pressure. For conditioning the air, chilled water provided by

central plant is passed through the AHU's chilled water coil to lower the temperature of the moving air as well as help reduce the air's humidity. The amount of chilled water passing through the coil is controlled by a valve which is actuated by a PID control loop. This control loop is driven by a difference in cooling demand setpoint and a cooling demand calculation that is a weighted combination of the average and maximum cooling loopout values from the individual zone control loops. This loop's output is the input (the desired discharge air temperature) for a second loop that actuates the chilled water valve.

Figure 6.8 details the chilled water loop. A chilled water pump works to maintain a specific supply pressure to provide the required chilled water flow for the AHU. Worth noting is that the chilled water supply and return temperatures are available to measure within the energy management system.

As described earlier, the UBO consists of 11 zones, 10 of which are actively controlled.

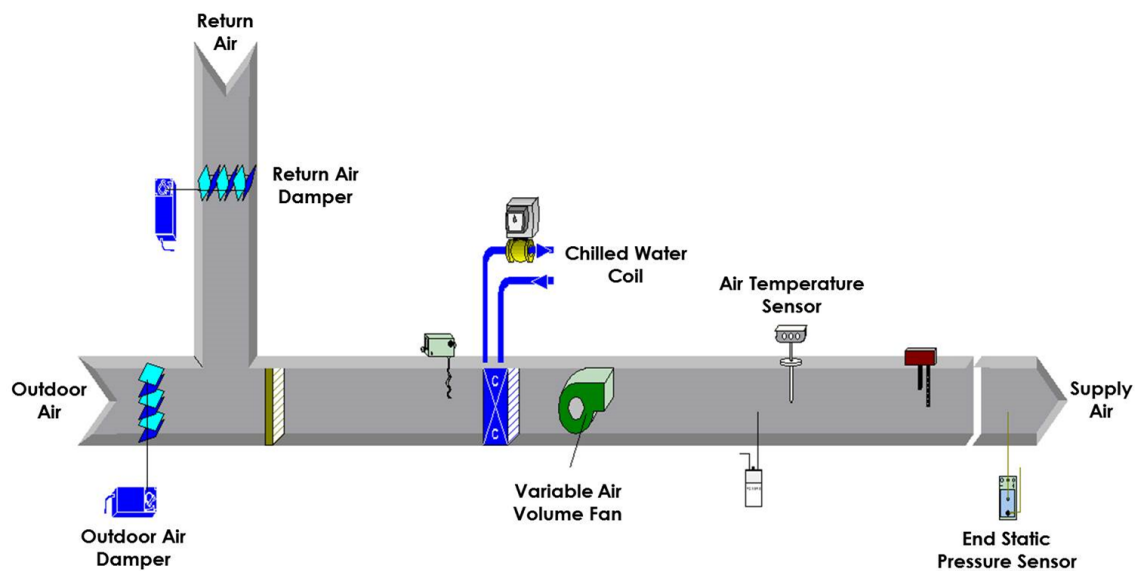


Figure 6.7: Rooftop Air Handling Unit (AHU) for the Utilities Business Office (UBO) at Texas A&M University.

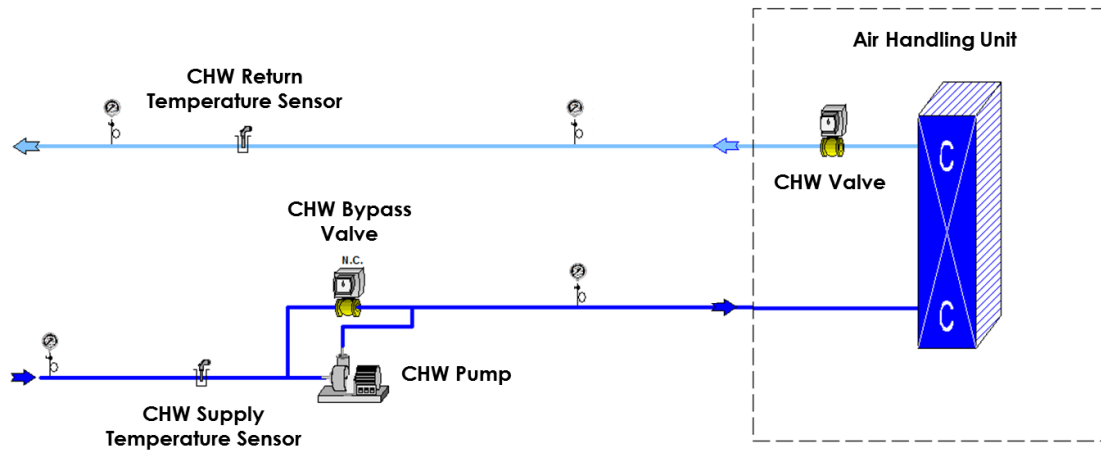


Figure 6.8: Chilled water loop for the Utilities Business Office (UBO) at Texas A&M University.

Each controllable zone is serviced by a VAV terminal box equipped with hot water reheat capabilities, an example of which is shown in Figure 6.9. The flow of conditioned air into the room is regulated by a damper in the terminal box whose position is determined by a PID control loop. The error signal for the control loop is the difference between the respective room temperature setpoint and the measured room temperature. The room temperature setpoint is determined by whether the room is occupied or unoccupied as well as whether the zone is in heating mode or cooling mode. A deadband control method is employed such that if the room is occupied, the zone VAV will heat the room to 70°F or cool the room to 74°F. If the room is unoccupied, the VAV will heat the room to 60°F or cool the room to 85°F. If the occupancy sensors in all the rooms read unoccupied, then the main AHU will turn off and the room temperatures will freely fluctuate.

The scope of this research effort was limited to developing economic cost functions for the AHU and the individual zones. Examining the AHU, two sources of economic cost become apparent: 1) the cost of electricity used by the AHU fan to move the air, and 2)

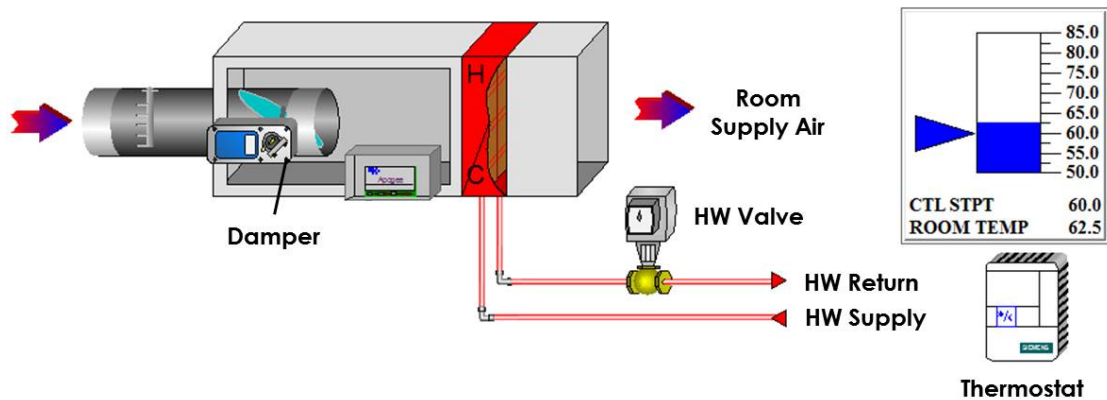


Figure 6.9: Variable Air Volume (VAV) box in the Utilities Business Office (UBO) at Texas A&M University.

the cost associated with the production of chilled water for cooling of the air. At the zone level, there is no equipment that consumes a significant amount of power or resources. The damper in the VAV terminal box is the only actuated component and the power required to move it is negligible. As mentioned in the background section, select previous building energy optimizations have included some form of occupant comfort, whether simply as constraints on the optimization, or as a measure to be minimized; however, to the knowledge of the the authors, no previously simulated or implemented building energy optimization has considered the economic cost of occupant comfort. Thus, an economic objective function that minimizes the cost of occupant discomfort as a measure of the loss of productivity was developed at the individual zone levels. The following sections detail the development of the three objective functions.

6.4.2 AHU Fan Economic Objective Function Development

As described above, the fan in the AHU works to maintain an end static pressure in the duct to move the required amount of air to condition the individual zones. To accomplish this, the fan motor requires electricity and there is a cost associated with the power used.

A simple way to measure the power consumed by a fan motor would be to use a power meter on the electrical lines to the fan and measure the power consumed; however, power meters are not often included in existing building energy systems. While a new one can be installed and integrated with the building control network, this results in additional cost to the operator and increases the data burden on the limited communication resources available within the building system's network. Therefore, the power consumed by the fan was instead determined by other available data, specifically the change in pressure across the fan and the volume flow rate of air being moved by the fan. With these two values, the work being performed by the fan on the air can be determined and converted into a measure of power, as power is work over time. The equation for work performed by a fan is given in Equation 6.2:

$$P_{fan}[W] = 0.1175 \left[\frac{m^3 \min}{ft^3 \sec} \frac{Pa}{in.H_2O} \right] \cdot \frac{q_{AHU} \cdot \Delta P}{\mu_f \cdot \mu_b \cdot \mu_m} [cfm \cdot in.H_2O] \quad (6.2)$$

where P_{fan} is the power consumed by the fan [W], 0.1175 is a conversion factor for imperial units, q_{AHU} is the air flow rate through the AHU [cfm], ΔP is the change in pressure across the fan [in. H₂O], and μ_f , μ_b , and μ_m are efficiencies for the fan blade, the fan belt, and the fan motor, respectively. The efficiencies were all assumed to be 0.9 based on comparable values. The change in pressure across the fan was taken from the end static pressure sensor in the AHU. As for the volume flow rate, the flow rate meters included in each of the VAVs were used in summation to calculate the total air flow rate through the AHU, assuming minimal duct losses and leaks.

Recalling the need for the supervisory controller to overlay the local controllers for the most practical implementation as well as the general form of the proposed economic objective function in Equation 6.1, the most appropriate control action u for the AHU fan objective function would be the end static pressure setpoint. Assuming the local PID

controllers have zero steady state error and sample significantly faster than the supervisory controller, the conjecture can be made that the end static pressure will equal the end static pressure setpoint. As such, assuming an electric utility rate of \$0.12 per kWh, the fan power can be calculated as:

$$J_{fan}[\$] = \frac{0.1175}{1000} \left[\frac{m^3}{ft^3} \frac{min}{sec} \frac{Pa}{in.H_2O} \frac{kW}{W} \right] \cdot \frac{C_{elec} \cdot q_{AHU} \cdot P_{EDS}^* \cdot t_s}{\mu_f \cdot \mu_b \cdot \mu_m} \left[\frac{\$}{kWh} cfm \cdot in.H_2O \cdot hr \right] \quad (6.3)$$

where C_{elec} is the rate of electricity cost [\$/kWh], P_{EDS}^* [cfm] is the end static pressure setpoint, and t_s [hr] is the sampling time of the supervisory controller. Reformulating Equation 6.3 to fit Equation 6.1, the result is given by Equation 6.4:

$$\begin{aligned} J_{fan} &= e^T Q e + e^T R + u^T S u + u^T T \\ Q &= 0 \quad R = 0 \quad S = 0 \\ u &= P_{EDS}^* \\ T &= \frac{0.1175 \cdot C_{elec} \cdot q_{AHU} \cdot t_s}{1000 \cdot \mu_f \cdot \mu_b \cdot \mu_m} \end{aligned} \quad (6.4)$$

thus, establishing an objective function that can minimize the cost of electricity used by the AHU's VAV fan by optimizing the end static pressure setpoint. The overall trend of the cost of electricity used by the fan as end static pressure is varied from 0.2 to 1.8 in. H₂O is shown in Figure 6.10 assuming an electric utility rate of \$0.12 per kWh.

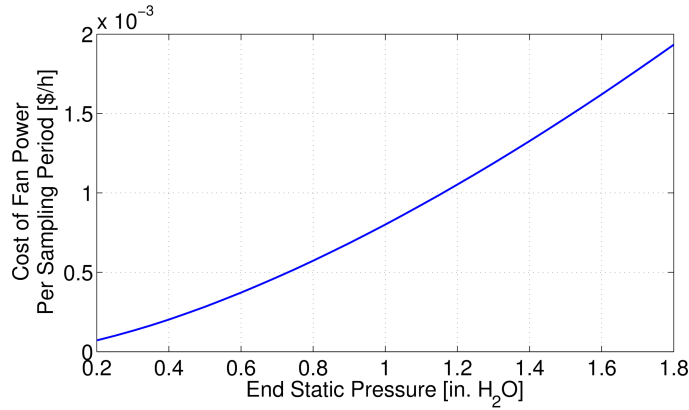


Figure 6.10: Trend of the objective cost of the fan as the end static pressure changes.

6.4.3 AHU Chilled Water Economic Objective Function Development

As described in the building operation details, chilled water is used in the AHU to condition the zone supply air. The UEM Office at Texas A&M University maintains utility usage data, specifically the cost per unit of energy of the respective utility. For chilled water, this is the dollar cost associated with producing one mmBtu of chilled water at the campus wide chilled water temperature of 45°F. In the AHU, the discharge air temperature setpoint is tracked by a PID loop that actuates the valve metering how much chilled water flows through the chilled water coil. The energy associated with chilled water usage can be determined by Equation 6.5:

$$\dot{Q} = \dot{m} \cdot c \cdot \Delta T \quad (6.5)$$

where \dot{Q} is the rate of change of heat, or energy, \dot{m} is the mass flow rate, c is the specific heat of the respective fluid, and ΔT is the change in temperature of the fluid. If a mass flow rate sensor is available on the AHU for the chilled water, then this measurement can be used to calculate the rate of change of energy of the chilled water and thus the overall

cost; however, mass flow rate sensors are not usually installed on chilled water lines at the AHU level. If this is the case, then a volume flow rate can be used such that:

$$\dot{m} = \rho \cdot q \quad (6.6)$$

where ρ is the density of water and q is the volume flow rate. With the UBO, the maximum flow rate through the chilled water valve was determined to be $3.41e^{-3} \text{ m}^3/\text{s}$. Assuming a linear valve/flow relationship, the chilled water valve position multiplied by the maximum possible flow will give the current volume flow rate of the chilled water. Using data values for the discharge air temperature setpoint and the chilled water valve position, a fit was generated to transform the setpoint to a valve position. Combining the relationships gives Equation 6.7

$$J_{CHW}[\$] = \alpha \cdot q_{max} \cdot \rho \cdot c \cdot \Delta T_{h_2o} \cdot t_s \cdot 0.00341214 \left[\frac{mmBtu}{kWh} \right] \cdot \frac{3682}{241} \left[\frac{\$}{mmBtu} \right]$$

$$\alpha = (3.021T_{DA}^{*2} - 109.4T_{DA}^* + 1002) \quad (6.7)$$

where α is the fitted relationship between the discharge air temperature setpoint (T_{DA}^* [°C]) and the chilled water valve position [%], q_{max} [m³/s] is the maximum flow through the valve (at 100% opening), ρ [kg/m³] is the density of water, c [kJ/kg°C] is the specific heat of water, ΔT_{h_2o} [°C] is the change in temperature of the supply and return chilled water, t_s [h] is the sample time of the supervisory controller, 0.00341214 [mmBtu/kWh] is a conversion factor from kWh to mmBtu, and the last term is the economic cost for the UBO of the consumed chilled water. Reformulating Equation 6.7 to fit Equation 6.1, where u equals the discharge air temperature setpoint, gives:

$$\begin{aligned}
J_{CHW} &= e^T Q e + e^T R + u^T S u + u^T T \\
Q &= 0 \quad R = 0 \quad S = 0 \\
u^T T &= \alpha \cdot q_{max} \cdot \rho \cdot c \cdot \Delta T_{h_2o} \cdot t_s \cdot 0.00341214 \left[\frac{mmBtu}{kWh} \right] \cdot \frac{3682}{241} \left[\frac{\$}{mmBtu} \right] \quad (6.8) \\
\alpha &= (3.021T_{DA}^{*2} - 109.4T_{DA}^* + 1002)
\end{aligned}$$

where the entire linear cost term $u^T T$ is equal to Equation 6.7. This result actually lends itself well to traditional convex optimization of the setpoint T_{DA}^* , due to the quadratic nature of the fit. The overall shape of the objective function can be seen as the AHU discharge air temperature is varied in Figure 6.11.

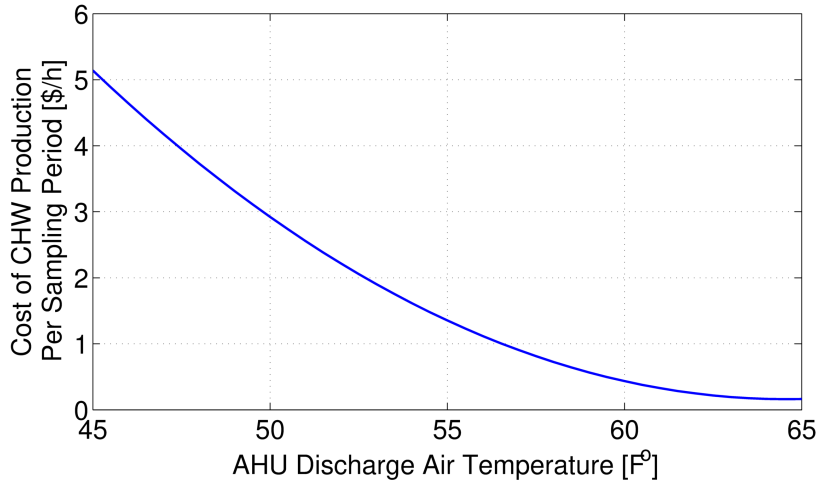


Figure 6.11: Trend of the objective cost of the chilled water used by the AHU as the AHU discharge air temperature changes.

6.4.4 Zone Occupant Comfort Economic Objective Function Development

In this section, a method of determining the economic cost of occupant discomfort is presented. In order to measure an occupant's level of discomfort, many have relied on the use of predicted mean vote (PMV), developed by Fanger in the 1970's [4]. PMV is a measure on the American Society of Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE) thermal sensation scale of -3 to 3, where negative numbers represent being too cold, positive numbers represent being too warm, and a value of zero represents being comfortable. Performing a large study of people, Fanger collected data regarding occupant's votes and created an equation to determine the PMV across a variety of environmental factors. These factors include air temperature, air relative humidity, relative air speed, mean radiant temperature, an occupant's insulation level due to clothing, an occupant's metabolic rate, an occupant's work output, among several other variables. The details of the equation can be found in [4]. From PMV, Fanger determined the Predicted Percentage of Dissatisfied (PPD) of people. The relationship of PPD to PMV can be seen in Figure 6.12.

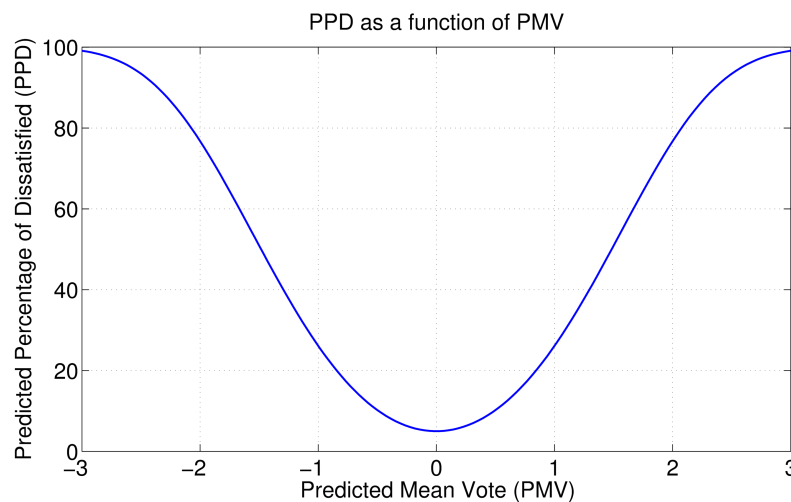


Figure 6.12: Relationship between PPD and PMV as calculated by Fanger [4].

One can see that at -3 and 3 PMV approximately 100% of the population would be dissatisfied. Worth noting is that even at zero PMV, 5% of the population, on average, will still be dissatisfied, attesting to the fact that each individual has specific preferences. While PMV determines how many people will be dissatisfied, a relationship to tie this to an economic cost is still needed. Fortunately, researchers have investigated the effect of occupant comfort on worker productivity, as mentioned in the literature review. One effort in particular tied PMV to a measure of Loss Of Productivity (LOP) [%]. By using regression analysis, a direct relation can be calculated between a worker's loss of performance and the PMV of an indoor climate by including the calculations of equivalent thermal situations from Gagge's two-layer human model [87] and Fanger's comfort equation [4]. For a detailed explanation of the relationship, see [10]. The results of Roelofsen's work [10] are two sets of coefficients for a regression fit for the cold side of the PMV comfort zone and for the warm side of the PMV comfort zone. The regression is a 6th-order fit shown in Equation 6.9:

$$LOP = b_0 + b_1PMV + b_2PMV^2 + b_3PMV^3 + b_4PMV^4 + b_5PMV^5 + b_6PMV^6 \quad (6.9)$$

where LOP is the loss of productivity and b_0, \dots, b_6 are the regression coefficients. The value of the Roelofsen's coefficients are repeated in Table 6.2, for reference. Roelofsen constructed the regression on the cold side to be zero at -0.5 PMV and for the warm side to be zero at 0 PMV, leaving a region between -0.5 to 0 PMV where LOP is zero. This is because several studies found a region of conditions near and below 0 PMV that had a negligible effect on productivity. Figure 6.13 shows the change in LOP as PMV varies. To tie LOP to an economic cost, a simple multiplication of the lost productivity percentage by the amount of salary and employed earns over the sample period was used, shown in

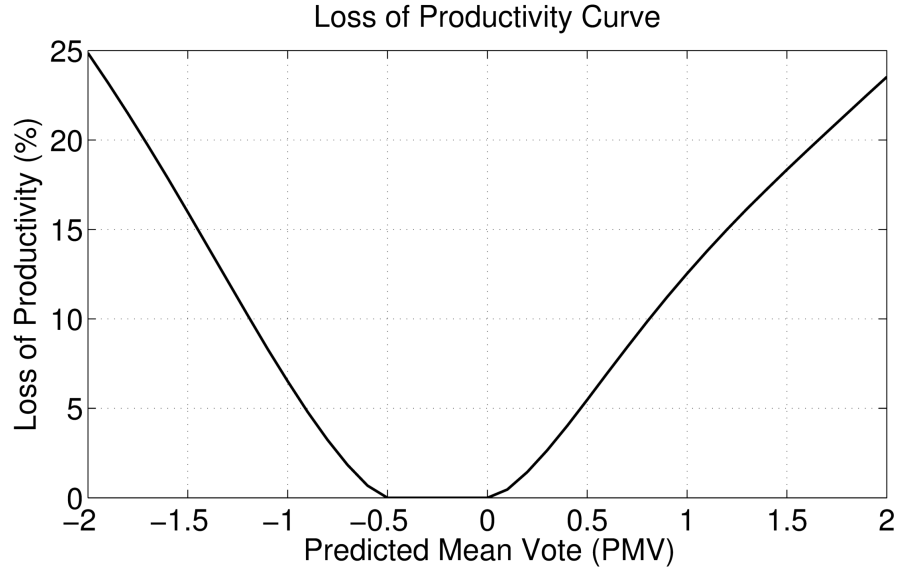


Figure 6.13: Shape of Loss Of Productivity function.

Equation 6.10

$$\beta[\$] = LOP[\%] \cdot \left(\frac{p_{year}[\$] \cdot t_s[h]}{52[weeks] \cdot 40[h]} \right) \quad (6.10)$$

where β [\$] is the lost productivity in wages, p_{year} [\$] is the annual salary for the zone, and t_s [h] is the sampling time. A 40 hour work week for the entire year was used as a

Table 6.2: Regressions for Loss Of Productivity fit of PMV from [10].

Regression Coefficients	Cold Side of PMV Comfort Zone	Warm Side of PMV Comfort Zone
b_0	1.2802070	-0.15397397
b_1	15.995451	3.8820297
b_2	31.507402	25.176447
b_3	11.754937	-26.641366
b_4	1.4737526	13.110120
b_5	0.0	-3.1296854
b_6	0.0	0.29260920

conservative assumption. In reality, there will be some variation due to holidays, vacation, and overtime. Considering the individual zones with respect to Equation 6.1, it is noted that while there is no control action u at the zone level that consumes energy, there is an error signal present. That is the error e in Equation 6.1, which is given by:

$$e = T_{zone} - T_{zone}^* \quad (6.11)$$

where T_{room} and T_{zone}^* are the zone temperature and zone temperature setpoint, respectively. The error is defined in this manner as this publication focuses on the systems when in cooling mode; thus, the error will be mostly positive during cooling mode as the zone temperature will generally be above or at the setpoint temperature. If the zone were in heating mode, the sign of the error term would need to be reversed. Recognizing that the zone temperature setpoint can be optimized by the supervisory controller to minimize the LOP by optimizing the zone's PMV, and that the error signal is a difference of temperatures, the sensitivities of the above relationships can be determined and combined to give an economic objective function. In determining PMV, only the zone air temperature and zone air relative humidity are changing. As such, an accurate fit of the PMV equation was created and used to reduce computation time and complexity. A metabolic rate of 70 [W/m²], a clothing insulation factor of 0.75 [m²K/W], and a relative air velocity of 0.2 [m/s] were assumed. Additionally, the mean radiant temperature was assumed to be equal to the zone air temperature. The fit is defined as:

$$PMV = 0.5542 \cdot Rh_{zone} + 0.23 \cdot T_{zone} - 5.44 \quad (6.12)$$

where Rh_{zone} [%] is the relative humidity of the zone and T_{zone} [°C] is the zone air temperature. Equation 6.12 is used to determine the sensitivity of PMV to changes in air temperature. The resulting combination of sensitivities is shown in Equation 6.13:

$$\begin{aligned}
J_{room} &= e^T Q e + e^T R + u^T S u + u^T T \\
Q &= 0 \quad S = 0 \quad T = 0 \\
R &= \left[\frac{\partial PMV}{\partial T_a} \right] \cdot \left[\frac{\partial LOP}{\partial PMV} \right] \cdot \left[\frac{\partial \beta}{\partial LOP} \right]
\end{aligned} \tag{6.13}$$

where the sensitivities are determined to be:

$$\left[\frac{\partial PMV}{\partial T_a} \right] = 0.23 \tag{6.14}$$

$$\begin{aligned}
\left[\frac{\partial LOP}{\partial PMV} \right] &= b_1 + 2b_2 PMV + 3b_3 PMV^2 + \\
&4b_4 PMV^3 + 5b_5 PMV^4 + 6b_6 PMV^5
\end{aligned} \tag{6.15}$$

$$\left[\frac{\partial \beta}{\partial LOP} \right] = \left(\frac{p_{year} \cdot t_s}{52\text{wks} \cdot 40\text{hrs}} \right) \tag{6.16}$$

where PMV is the predicted mean vote, p_{year} is an occupant's annual salary, and t_s is the sampling time of the supervisory controller, in hours. The cost calculation assumes that if more than one occupant is in a zone, the sum of the occupant's salaries is used for p_{year} and that a standard 40 hour work week is used. For Equation 6.15, the coefficients vary as described by Table 6.2. If the zone's PMV falls within the range of -0.5 to 0, then Equation 6.15 is equal to zero. Thus, an economic objective function that minimizes the cost of productivity loss by optimizing the zone temperature setpoint was developed.

The overall trend of the objective function as the zone air temperature varies is shown in Figure 6.14. The abrupt changes occur at the PMV values of -0.5 and 0. This is due to the fit of the LOP equation. It is worth noting that this curve will shift depending on other variables in the objective function that are varied, such as annual salary and relative humidity of the zone. Also, depending on the value of the user-defined setpoint,

the objective cost can become negative. This does not mean that the zone is earning money, but occurs due to the structure of the objective function. An optimization competition can occur between the error term and the coefficient R . As the error is defined as the difference in the zone temperature and zone temperature setpoint, the objective cost will be zero when the zone reaches this defined setpoint; however, if this setpoint is not equal to the optimal comfort temperature, as determined by the PMV function, then the objective cost will also be zero if the zone air temperature is equal to the optimal comfort temperature. This behavior presents an interesting control question. To operate at the most cost effective point, as defined by the objective function, it would require the zone temperature setpoint to be set to the PMV optimal temperature, but in doing so, the ability for occupants or building managers to provide the system feedback about their comfort is removed. So which is more important: the ability for occupants to choose their zone temperatures or allow the system to determine what is best for the occupants? This question deserves additional investigation but is beyond the scope of this publication. Fortunately, with how

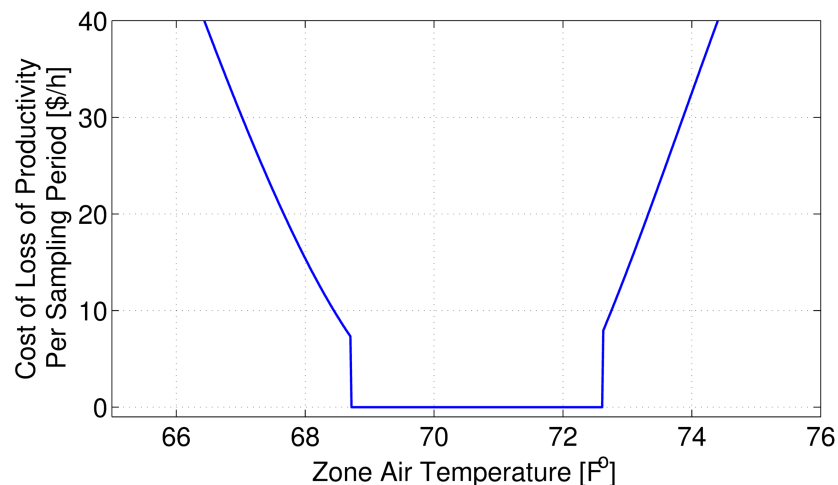


Figure 6.14: The change in objective cost of the loss of productivity due to discomfort due to changes in zone air temperature during a 15 minute period.

the objective function is structured currently, the system will propose a compromise: a temperature between the user-defined setpoint and the PMV optimal comfort temperature.

For a centralized implementation, the objective function then becomes:

$$J_{total} = J_{fan} + J_{CHW} + \sum_{i=1}^n J_{zone,i} \quad (6.17)$$

where n is the number of zones in the system. Equation 6.17 is the system objective function that will be minimized in the optimization.

6.5 Simulation Design

A model of the UBO was first created with the aid of SketchUp [88], a 3D modeling software. Using dimensions taken from the building, the single story layout was replicated. Utilizing the plug-in from OpenStudio, thermal zones, boundaries, and interactions were defined. This model was then exported as an input file (IDF) for EnergyPlus [89], an open-source energy simulation program that has been developed by the Building Technologies Office (BTO) within the U.S. Department of Energy (DOE). Using EnergyPlus, HVAC equipment was added to the model based on the equipment present in the UBO. While EnergyPlus excels at modeling and simulation, it is not immediately accessible for controller development and implementation. As such, controllers for the AHU and the zone level VAVs were created in MATLAB. To enable co-simulation between EnergyPlus and MATLAB, two programs were used. The first was MLE+ [90], an open-source MATLAB toolbox for creating the necessary configuration files and providing functions to connect EnergyPlus and MATLAB with an easy-to-use graphical interface. MLE+ utilizes the Building Controls Virtual Test Bed (BCVTB) [91] as the communication backend between EnergyPlus and MATLAB, providing the co-simulation functionality.

The model developed in SketchUp is shown in Figure 6.15. The individual zones were setup as defined in the building layout and exterior doors & windows were placed as accu-

rately as possible. The AHU and zone VAVs were added using EnergyPlus' HVACTemplate objects. A central electric chiller was also added to supply chilled water to the AHU. The chilled water output temperature is regulated to 45°F, the same as the supply chilled water temperature for the UBO. Currently, EnergyPlus does not offer modeling of pressure with variable air volume systems, thus another solution was necessary to simulate the UBO's control and physical limitations of end static pressure and flow in the AHU. In order to accomplish this, it was assumed that the dynamics of the fan speed and end static pressure were fast enough compared to the simulation timestep (1 minute) to be considered instantaneous. Additionally, the assumption that the fan would supply the requested end static pressure, constrained by the physical limitations of the AHU and ducting was made. To determine this constraint, data from the real UBO building was analyzed and a maximum possible work performed by the fan was calculated (910 W). During the optimizations, the constraint is calculated by using Equation 6.18. To determine the total volume flow through the AHU, individual models of the zone VAVs were generated from

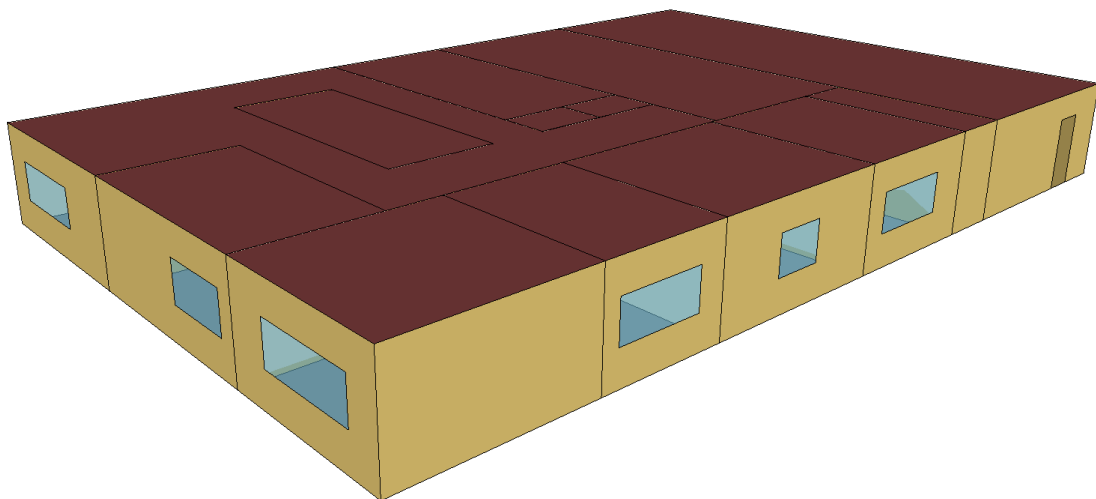


Figure 6.15: SketchUp model of the Utilities Business Office (UBO) at Texas A&M University.

data based on VAV damper position and end static pressure. The effect of outdoor air temperature was also considered on the VAVs as the AHU draws in outdoor air, but was shown to be minimal. This is most likely due to the fact that AHU is able to meet its discharge air temperature setpoint, even at varying flows, effectively isolating the VAV supply air from the outdoor air conditions. The flows from the VAV models are summed to obtain the total flow through the AHU, assuming minimal duct losses. The constraint can be written as:

$$910[W] \geq 0.1175 \left[\frac{m^3 \text{ min}}{ft^3 \text{ sec}} \frac{Pa}{in.H_2O} \right] \cdot \frac{q_{AHU} \cdot P_{EDS}}{\mu_f \cdot \mu_b \cdot \mu_m} [cfm \cdot in.H_2O] \quad (6.18)$$

Thus, the optimization will only ever choose an end static pressure setpoint that is physically achievable by the system. This end static pressure is then passed to the VAV models which, along with the commanded damper positions, produce individual zone air volume flows.

The overall control hierarchy can be seen in Figure 6.16. The supervisory controller supplies the zone temperature setpoints (T_{ZONES}^*) to the respective PID reference inputs and the end static pressure setpoint (P_{EDS}^*) to the VAV models. The discharge air temperature setpoint (T_{AHU}^*) is supplied directly to EnergyPlus as the control for the chilled water valve is implemented using an appropriate setpoint manager within EnergyPlus. The zone temperature error is fed to the cooling PID controller which outputs a desired percentage of maximum flow setting (0% to 100%). This signal is passed through block G which maps the maximum flow percentage to the range of minimum flow to maximum flow. If the minimum flow setting for a zone is zero, then the signal remains unchanged. The desired flow percentage is then passed to the flow PID controller which then produces a desired damper position. This damper position is then used in the VAV models as previously described. The output from the VAV models, desired zone air volume flows, are

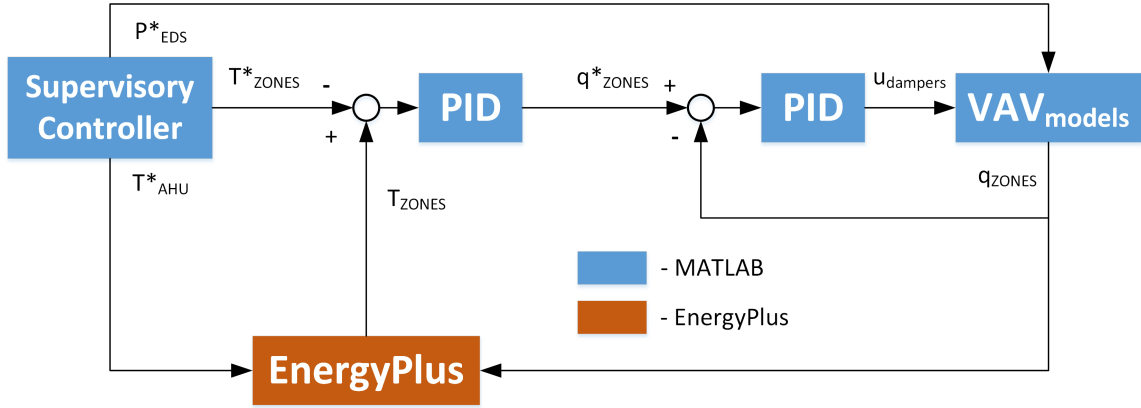


Figure 6.16: Control hierarchy used in simulation of the UBO.

converted to air flow fractions and then sent to zone VAVs in EnergyPlus where the room dynamics are simulated for one timestep.

For the supervisory controllers, a prediction model is necessary to determine the future zone temperatures as the setpoints are optimized. A previously developed modeling algorithm was employed to generate models for the individual zones [93]. One of the main reasons this algorithm was chosen is because of its ease in producing models, requiring the user to only select input and output data. The full details of this process are beyond the scope of this publication, but more information can be found in [93]. Briefly, data from the EnergyPlus simulation is analyzed to develop discrete, linear models of the selected parameters. These models take the form of:

$$\begin{aligned}
 x_i(k+1) &= A_i x_i(k) + B_i u_i(k) + K_i e_i(k) \\
 y_i(k) &= C_i x_i(k)
 \end{aligned}
 \tag{6.19}$$

where A , B , C , and K are the identified system matrices, u is the model input vector, y is the predicted output, and e is the error defined as the difference between the current value and the previous predicted value. The algorithm automatically identifies signifi-

cant coupling interactions between zones and includes the respective zone temperatures ($T_{zone\ dist.,j}$) as measured disturbances. In addition to the temperature of these disturbance zones, other parameters such as outdoor air temperature (T_{OA}), outdoor air relative humidity (Rh_{OA}), AHU discharge air temperature (T_{AHU}), end static pressure (P_{EDS}), and zone temperature setpoints ($T_{zone,i}^*$) are used as inputs to ARX, ARMAX, Output Error, and Box-Jenkins modeling methods with the output being the respective zone temperatures ($T_{zone,i}$). The best fitting model is selected and then the individual models are combined into a centralized model of the entire system. Steady-state predictions from this centralized model are then used by the supervisory controller to optimize the UBO's 12 setpoints (discharge air temperature setpoint, end static pressure setpoint, and 10 zone temperature setpoints) to minimize the economic cost functions previously described. The optimization occurs every 15 minutes.

Steady-state relationships were chosen over dynamic relationships for the initial implementation and validation of the proposed economic objective function strategy. This choice takes advantage of the fact that in the building energy systems, the control variable's dynamics (AHU discharge air temperature, end static pressure, VAV damper position) change quickly versus the other system variables (outdoor air temperature, outdoor relative humidity, room temperature), which change relatively slowly over the optimized timestep. The exact model inputs and outputs are shown in Table 6.3.

Models were identified for two operational cases: 1) where the zone VAV damper still had actuator range (i.e. the damper was not fully open), and 2) where the zone VAV damper is fully open. The separate models were necessary as the effect of the model inputs varies greatly between the two cases. In case 1, the effect of T_{AHU}^* and P_{EDS}^* are minimal compared to $T_{zone,i}^*$. This is due to the fact that the VAV damper is actuated by a PID controller with $T_{zone,i}^*$ as the reference. The PID controller is able to reject changes in T_{AHU}^* and P_{EDS}^* by changing the damper position to achieve $T_{zone,i}^*$. However, in case 2

Table 6.3: Inputs and outputs for the generated models.

Model Input	Model Output
T_{OA}	
Rh_{OA}	
$T_{zone,i}^*$	$T_{zone,i}$
T_{AHU}^*	
P_{EDS}^*	
$T_{zone\ dist.,j} - T_{zone,i}$	

when the damper is fully open, T_{AHU}^* and P_{EDS}^* become the inputs of significance as they directly effect the zone temperature, determining the amount and the temperature of the incoming conditioned air.

A decision process was necessary to determine when each model should be used. This decision was made based off of two conditions. The first determined if the predicted zone temperature using the case 1 models was greater than the prediction from the case 2 models. This condition served to verify whether the predicted temperature from the case 1 models was currently achievable with the state of the AHU. If the case 1 predicted temperature was lower than the case 2 predicted temperature, then the VAV wouldn't be able to achieve the case 1 predicted temperature with the current T_{AHU}^* and P_{EDS}^* values. The second checked if the current VAV damper position was less than 95%, or in other words if the VAV still had actuator range of the damper. The value of 95% was used as opposed to 100% to serve as a threshold and help prevent the system from oscillating between cases. If both these conditions were true, then the case 1 models were used; otherwise, the case 2 models were used. To help ensure smooth transfer between the two models and more accurate predictions, errors were calculated between the previous predictions the measured temperatures and included in the current prediction.

6.6 Results

Simulations were completed to determine the steady-state optimal control method's performance compared to the current control strategies in place in the UBO. An additional simulation was completed to demonstrate the current control method's ability to track LOP optimal ($PMV = -0.25$) zone temperature setpoints. By using LOP optimal temperature setpoints, a more direct performance difference can be determined between the current control method and the proposed steady-state optimal control method. Lastly, a simulation showcasing the proposed optimal control method's ability to prioritize certain zones over other zones was completed.

6.6.1 UBO Simulation with Current Building Controls

The current control method was simulated on the UBO under two operational cases: 1) with the building operator defined zone temperature setpoints ($23\text{ }^{\circ}\text{C}$), and 2) with LOP optimal zone temperature setpoints (the air temperature at which $PMV = -0.25$). The numerical results of the first case will be included in the discussion later in the paper while this section will focus on the results from the second case. Figure 6.17 shows one day of the zone temperatures for the UBO building using the current control method and PMV optimal temperature setpoints. The outdoor air temperature (dashed line) is included in the plot for reference.

The current control method with demand calculations and local PID control shows the ability to track the LOP setpoints fairly accurately. Worth noting is that beginning around 2 PM, Zone 1's temperature starts to drift upwards away from the optimal temperature. This is due to Zone 1's damper being fully open combined with the fact that the AHU fan has reached its power limit and the discharge air temperature is not decreasing fast enough to provide the additional required cooling. Figure 6.18 shows the system's end static pressure (dashed green line) and the total air flow (solid blue line) in the AHU. Shortly after 1 PM,

the end static pressure begins to decrease as the total air flow continues to increase. This is the point where the AHU fan has reached its maximum power capabilities. As the zone dampers continue to open, there is less obstruction to the passage of air, decreasing the pressure and increasing the flow.

Figure 6.19 shows the chilled water flow and discharge air temperature of the AHU. The discharge air temperature gradually decreases throughout the day (dashed green line), responding to the increase in cooling demand. As the temperature drops, more chilled water is required to cool the air, displayed by the increase in the mass flow rate of the chilled water (solid blue line).

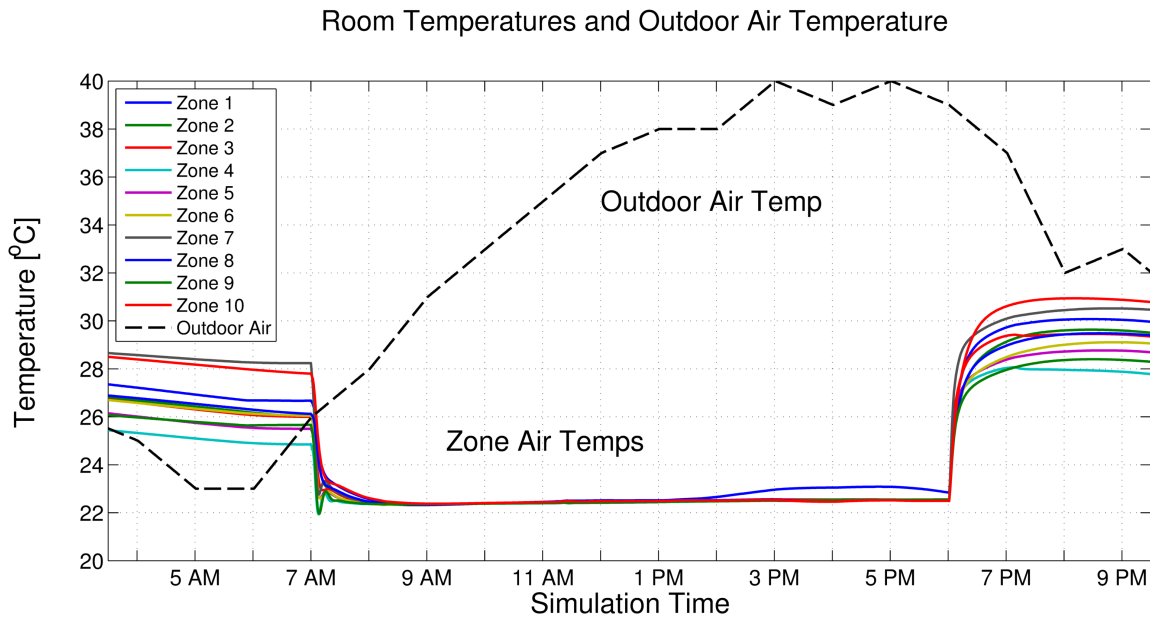


Figure 6.17: Zone and outdoor air temperatures for the UBO using the current currently implemented methods with PMV optimal temperature setpoints.

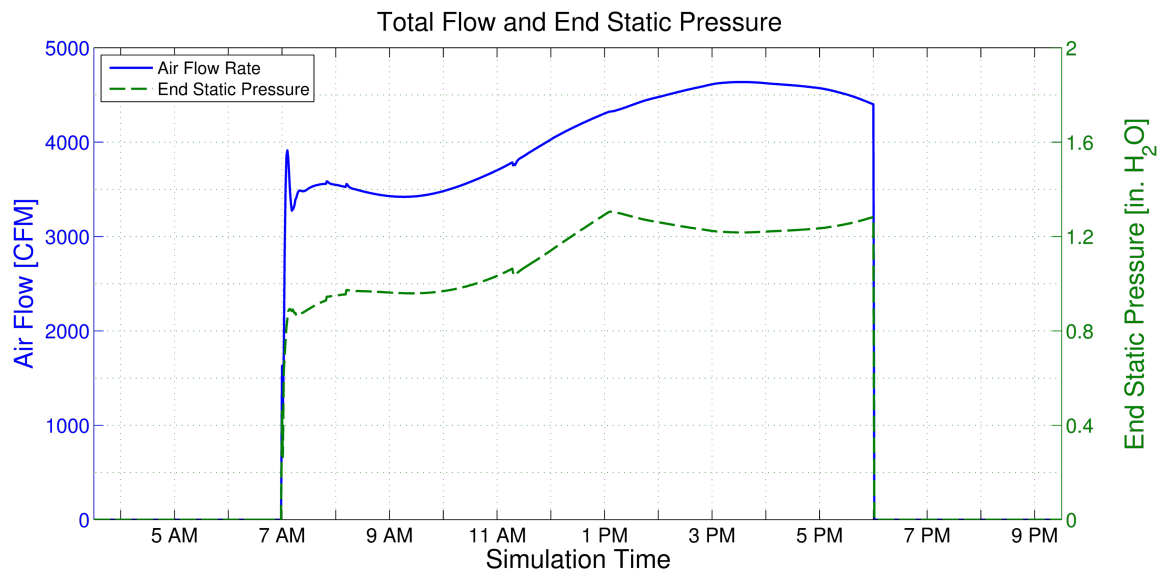


Figure 6.18: Total air flow and end static pressure in the AHU for the UBO using the currently implemented control methods with PMV optimal temperature setpoints.

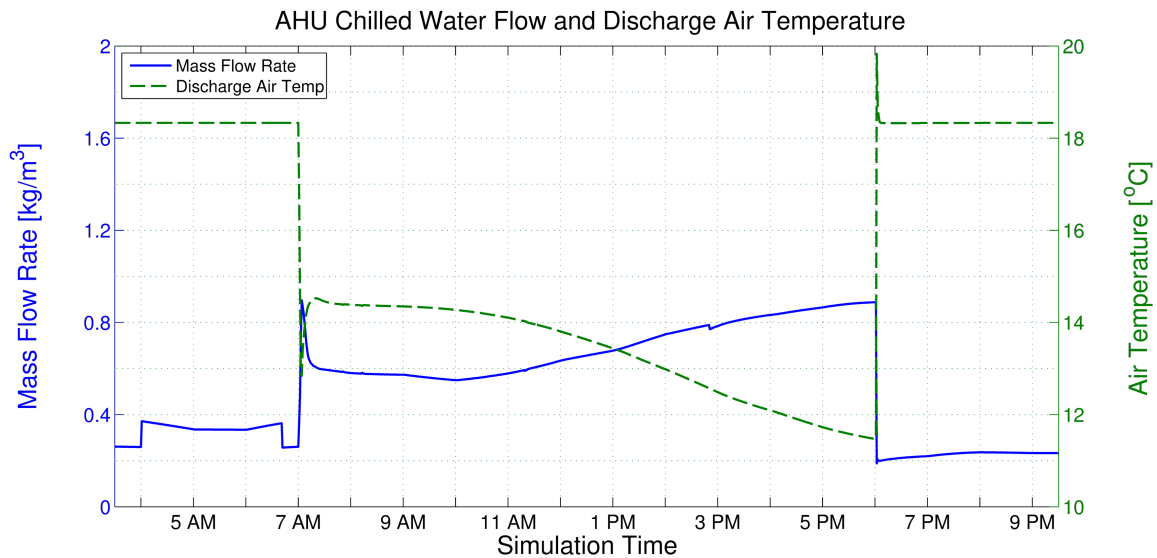


Figure 6.19: Chilled water flow and discharge air temperature for the UBO using the currently implemented control methods with PMV optimal temperature setpoints.

6.6.2 Steady-State Optimal Control Simulation

The proposed steady-state optimal control method was simulated on the UBO with the user-defined temperature setpoints equal to the PMV optimal temperature. All the zone temperatures can be seen in Figure 6.20. Compared to Figure 6.17, the zone temperatures appear to vary slightly more through out the day. This is not because the zones temperatures aren't optimal, but because of the range of PMV (-0.5 to 0) for zero loss of productivity. This range of PMV's allows the optimization a band in the individual zone temperatures while minimizing the utility cost of the chilled water and electricity and leveraging the coupling that exists between zones.

Figure 6.21 shows the end static pressure and the total air flow through the AHU. Comparing to the current control method simulation, the air flows follow relatively similar paths, with the pressure in the steady-state method simulation taking a higher value but remaining more constant throughout the day. Figure 6.22 shows a lower discharge air temperature for the steady-state case. While this results in increased flow rates of the chilled water, the cost may not necessarily be higher as the return chilled water temperature may be lower, meaning the chiller has to cool the water over a smaller difference in temperatures. This lower discharge air temperature helps the steady-state optimal control method to achieve more reduction in the cost of lost productivity due to discomfort, enabling lower temperatures in the zones.

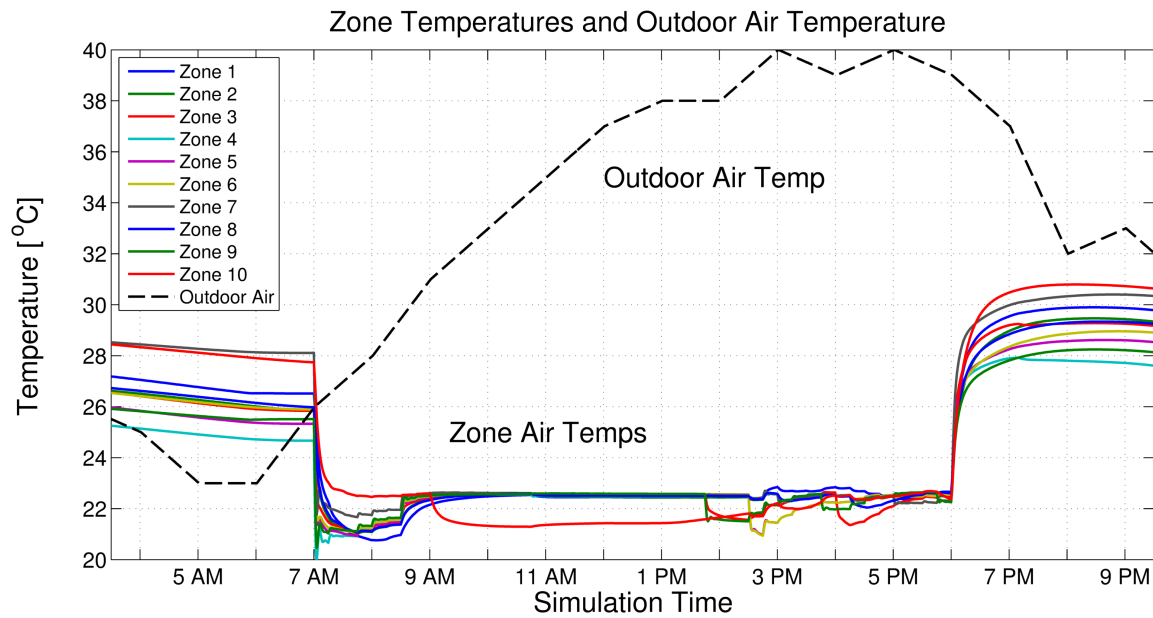


Figure 6.20: Zone and outdoor air temperatures for the UBO using the steady-state control method with PMV optimal temperature setpoints.

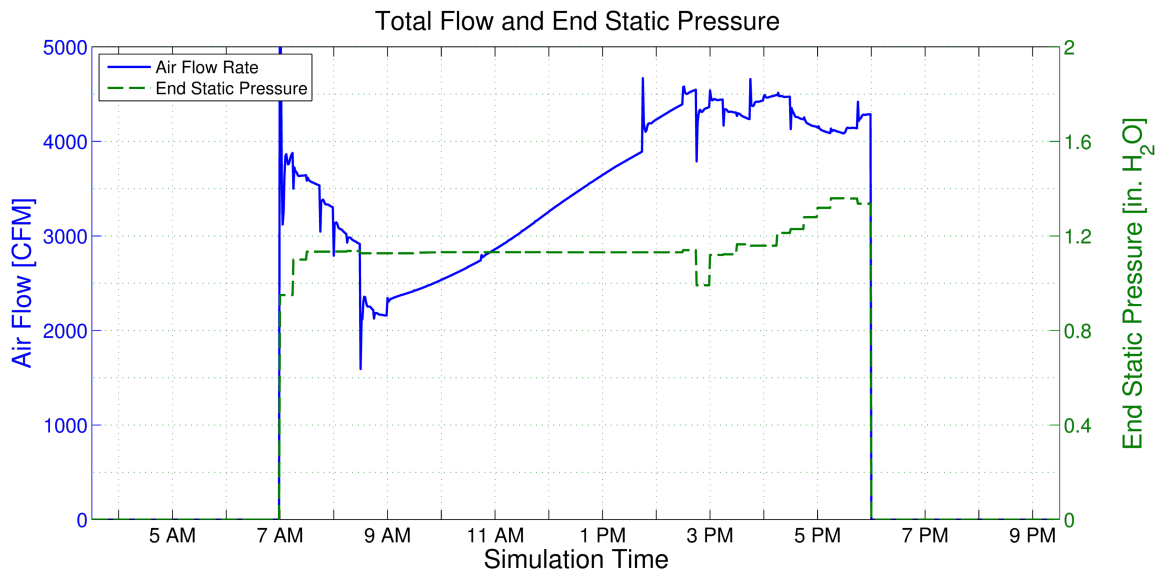


Figure 6.21: Total air flow and end static pressure in the AHU for the UBO using the steady-state control method with PMV optimal temperature setpoints.

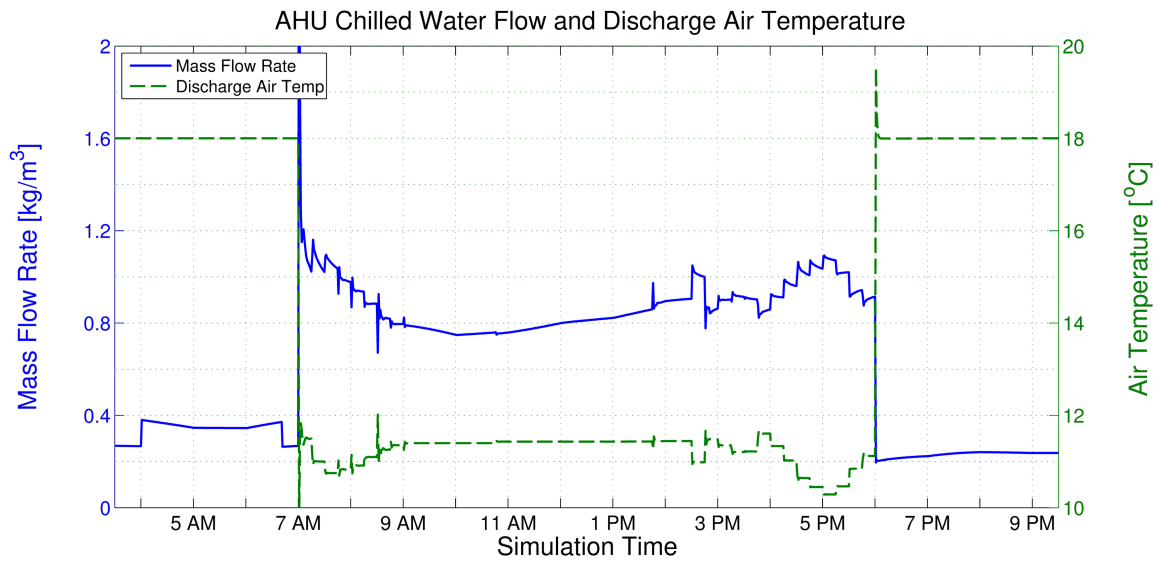


Figure 6.22: Chilled water flow and discharge air temperature for the UBO using the steady-state control method with PMV optimal temperature setpoints.

6.6.3 Very Important Person (VIP) Simulation

To demonstrate one of the proposed steady-state algorithms capabilities, a simulation in which one zone was valued significantly more over the other zones in the building was completed. This can occur in the situation where there is a very important person (VIP) that requires comfortable conditions to be maintained, or in the case where other rooms are less important to maintain at a specific comfort level and can be warmer to reduce utility usage. In this simulation, Zone 5 was chosen as the VIP zone. Figure 6.23 shows all of the zone temperatures. The other zones are higher in temperature throughout the day, while Zone 5 is maintained at a lower temperature. Several zone temperatures can be seen rising above the 0 PMV threshold after 1 PM, when the cooling demand for the day is the greatest. This departure from the optimal LOP range between -0.5 and 0 PMV is due to the optimization balancing the cost of discomfort in the zones with the cost of the utilities.

Figure 6.24 provides further insight into the maintaining of comfort in Zone 5. The zone temperature is shown with the solid blue line and two thresholds are displayed: 1) the dashed red represents the 0 PMV threshold, and 2) the dashed green represents the -0.5 PMV threshold. After the building is initially occupied, the zone temperature is maintained between the two thresholds resulting in zero loss of productivity for Zone 5.

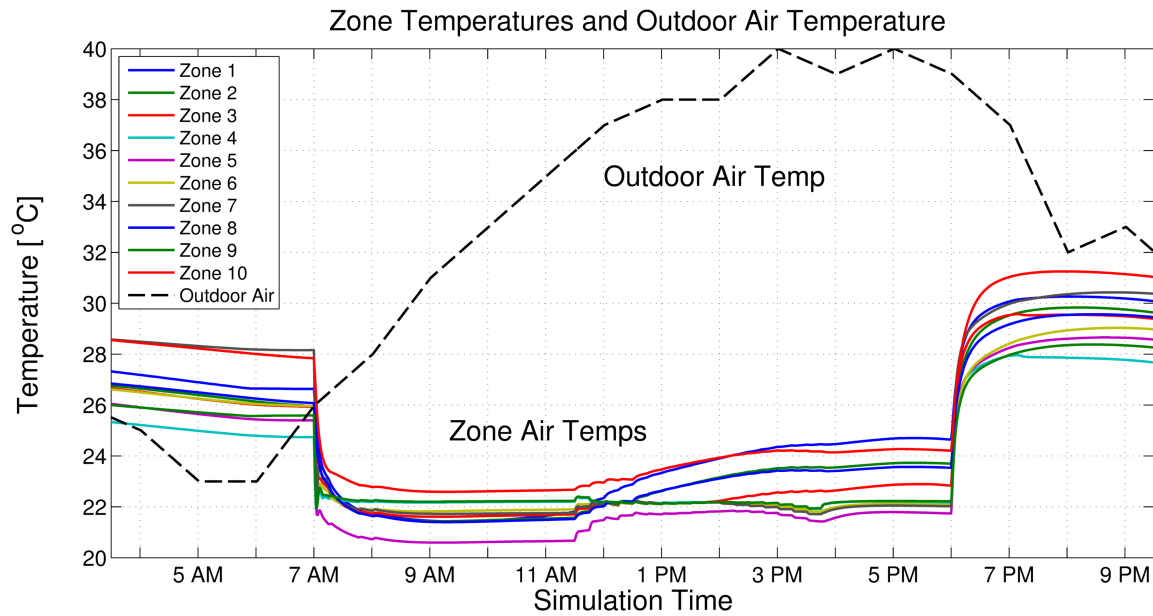


Figure 6.23: Zone and outdoor air temperatures for the UBO using the steady-state optimal control method with a VIP zone.

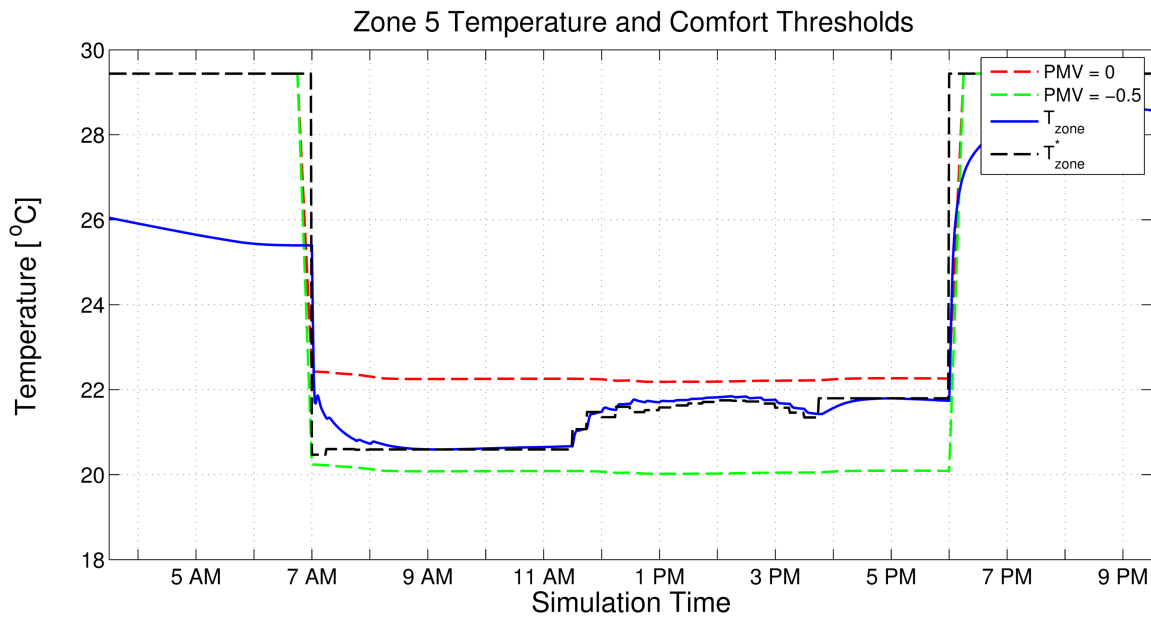


Figure 6.24: Zone and outdoor air temperatures for the UBO using the steady-state optimal control method with a VIP zone.

6.7 Discussion

Table 6.4 shows the annualized costs from the two simulations performed with the current control method and the simulation performed with the proposed steady-state optimal control method. The costs from the simulations were annualized using cooling degree days for the College Station, TX area. The first simulation in the table is the UBO as it is currently operated. The building technician defined temperatures of 23 °C were used as the zone temperature setpoints. While this simulation used the least amount in utilities, it also had the greatest cost in terms of loss productivity due to discomfort. The technician defined setpoints are above the PMV optimal range for loss productivity, so this is to be expected.

The second simulation in the table is the UBO and its current control method but with PMV optimal temperature setpoints (setpoints that give 0 PMV). Worth noting is the

significant decrease of 93.1% in the cost of loss productivity just by changing the user-defined setpoints. This of course comes at an increase (approximately 15.4%) in utility cost; however, the total cost was reduced by \$5,307.85, or 38.1%. The third simulation is the UBO with the proposed steady-state optimal control method. This method resulted in the greatest decrease of the cost of lost productivity of 95.6% with a slightly higher cost in utilities of 5.4%. The steady-state optimal control method also gave the greatest decrease in overall cost, saving \$6,189.48, or 44.5% of the original cost. This translates to utility savings of \$704.47, or 8.7%, and total cost savings of \$881.63, or 10.2%, over the current control method with PMV optimal setpoints.

A significant observation is that just by changing the current zone temperature setpoints, the UBO building operators could have immediate savings in terms of increased productivity for a slight increase in utility cost with no change in control methods. Furthermore, additional savings can be had through the use of the advanced steady-state optimal control method. Other benefits of using the advanced controller are that over time, the modeling identification algorithm used will update and improve the steady-state prediction models automatically, providing for the potential for further savings over time. Also, the models will adapt as seasonal climate shifts occur and equipment efficiency changes

Table 6.4: Annualized economic costs of the different simulation scenarios.

	Zone			Total	
	Comfort Cost	CHW Cost	Fan Cost	Utility Cost	Total Cost
Current Control with User Setpoints	\$6,869.38	\$4,093.61	\$1,255.67	\$7,046.03	\$13,915.41
Current Control with PMV Setpoints	\$476.37	\$4,562.48	\$1,677.62	\$8,131.19	\$8,607.56
Optimal Predicted Steady-State Setpoints with PMV Setpoints	\$299.21	\$4,383.28	\$1,226.63	\$7,426.72	\$7,725.93

while the current control method would require manual tuning as the system parameters change to maintain the same level of performance. Currently in the UBO, user overrides of the current control system and setpoints are common. While these overrides may reduce in frequency with a change in zone temperature setpoints to PMV optimal values, the impact of overrides would still be greater with the current control system compared to the proposed steady-state optimal controller. The steady-state optimal controller balances the optimal economic zone setpoint with the user-defined setpoint, reducing the impact of overrides. The advanced controller also has the added benefit of allowing the building operator to easily prioritize zones to maximize comfort in by adjusting the weight of the annual salary of respective zones. The authors acknowledge that not all building operational situations call for maximum comfort and productivity, but propose that the importance of occupant comfort and its significant economic impact on businesses and organizations merits further investigation. As building design and control move forward, optimizing occupant comfort should be considered a priority as opposed to a standard to be met.

6.7.1 Future Directions

The presented work has raised some interesting questions, such as the importance of occupant comfort and its associated economic cost of loss productivity versus utility cost. Further study into this relationship is necessary and can include additional economic and psychological measures, such as the impact of productivity in specific working environments, as well as investigating if integrating user feedback into the control would be beneficial or not. Also, while the simulations showed the proposed steady-state controller to be successful in reducing the overall cost, verification on the actual building is needed. It would also be of interest to investigate the performance of steady-state predictions versus dynamic predictions and determine exactly how much benefit there is from implementing

the more computationally difficult dynamic models of fully implemented MPC.

6.8 Conclusion

This paper presented a novel economic steady-state optimal control method for control of energy systems in buildings. The control method used economic objective functions that were derived based on systems found in the Utilities Business Office at Texas A&M University to minimize the economic cost associated with operating a building. Specifically, the cost of utilities (electricity and chilled water) were optimized alongside the cost of loss productivity due to occupant discomfort. Co-simulations of the steady-state optimization controller were performed with EnergyPlus and MATLAB. The simulation results showed improved comfort performance and economic savings with the use of the steady-state optimization controller over the current control method implemented in the Utilities Business Office.

7. PAPER E: DYNAMIC MODEL PREDICTIVE CONTROL VS STEADY-STATE PREDICTIVE OPTIMIZATION OF BUILDING ENERGY SYSTEMS

7.1 Synopsis

Optimal control of energy systems in buildings is a current area of interest; however, full implementation of Model Predictive Control (MPC) with dynamic prediction models becomes increasingly difficult as larger and more complex systems are controlled. The purpose of this paper is two-fold: 1) to introduce an economic MPC strategy that minimizes utility cost and the cost of loss productivity due to occupant discomfort, and 2) investigate the performance increase gained by using dynamic models over steady-state relationships to help determine when the added difficulty, communication burden, and computational cost is justified. A model is developed of a real building on Texas A&M University's campus in EnergyPlus. Dynamic MPC and steady-state optimization are implemented on the simulated building and co-simulation with the control implemented in MATLAB is performed. Operational cases including constant and random occupancy as well as standard and optimal setback temperatures are investigated. The proposed algorithm's ability to determine optimal setback temperatures as well as prioritize certain zone's comfort over others is demonstrated. It is found that the performance increase from dynamic models is marginal in some of the proposed cases and that steady-state predictions can provide similar performance results as a fully dynamic solution.

7.2 Introduction

Efficiency in use and consumption of natural resources has increased in importance as the global population and spread of modern technology has grown. In the U.S. alone, total energy consumption has nearly tripled over the last 65 years from 34.6 quadrillions Btus (quads) in 1950 to 97.4 quads in 2015 [12]. Of this energy consumed, renewable ener-

gies comprise approximately 10% with the rest coming from non-renewable sources [12]. In response to growing energy needs and potential scarcity of natural resources, many nations have put forth specific renewable energy targets. These targets aim to reduce humanity’s dependence on non-renewable energies, increasing energy security and ensuring a sufficient infrastructure for future development. The European Union’s (EU) Renewable Energy Directive established a binding goal of 20% final energy consumption from renewable sources by 2020 [13]. For the U.S. the Department of Energy has set a goal for the nation to have 20% of its electricity sourced from wind energy by the year 2030 [14]. While renewable energy sources will continue to increase over the coming years, improving efficiency in how energy sources are consumed can also have a significant impact in alleviating energy demand and scarcity of natural resources.

The U.S. consume 19% of the total world energy consumption, as shown in Figure 7.1. Approximately 40% of all U.S. energy goes to building operations in the commercial and residential sectors [15]. Dividing the energy used in the commercial and residential building sectors down by source shown in Figure 7.1, the data shows that approximately 75% comes from fossil fuels. As a result, energy usage in buildings account for 40% of the

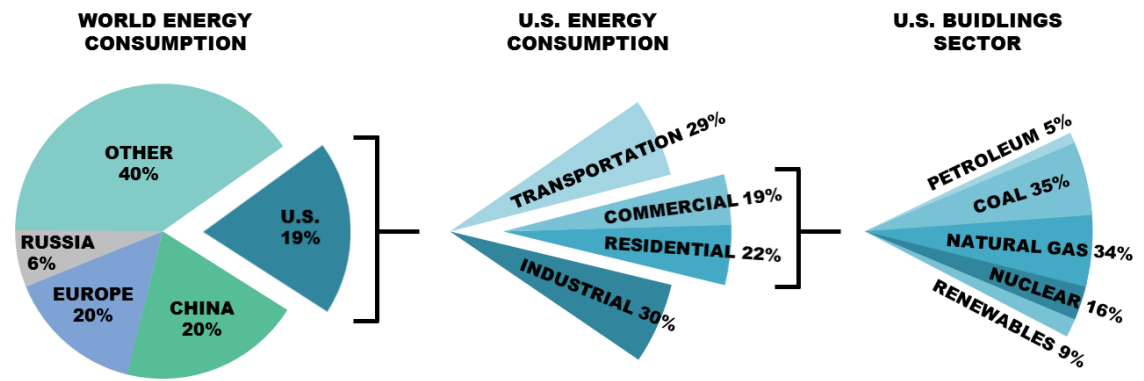


Figure 7.1: U.S. energy consumption and breakdown of energy used in different sectors [2].

total U.S. carbon emissions [16]. Also, buildings' share of U.S. energy consumption has increased from approximately 34% in 1980 to approximately 40% in 2010 [2]. Specifically, the commercial sector has shown an increased consumption percentage over time while the residential share has remained relatively constant. According to [2], the commercial buildings sector consumes nearly 20 quads (20 quadrillion BTUs).

Within the buildings sector, the commercial site energy consumption can be broken down into specific end uses, as shown in Figure 7.2. Examining the top end uses reveals that space heating (27%), lighting (14%), space cooling (10%), water heating (7%), and ventilation (6%) are responsible for 64% of commercial building energy usage. These categories can be combined more generally to refer to services required mostly when a building is occupied (space conditioning and lighting). The smaller categories and the all-encompassing "other" category represent 25%, while the other 11% is an adjustment

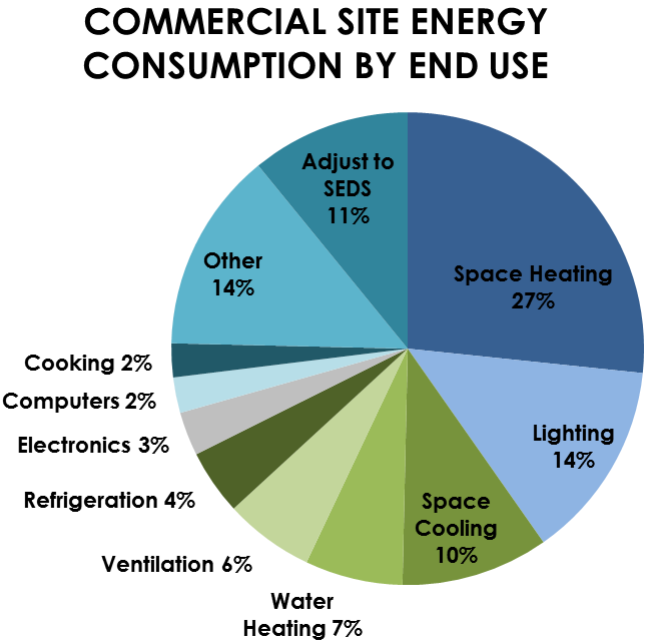


Figure 7.2: Site energy consumption by end use for U.S. commercial buildings [2].

that the Energy Information Administration uses to relieve discrepancies between data sources, specifically defined as “Energy attributable to the commercial buildings sector, but not directly to specific end-uses” [15]. In 2010 utilities cost businesses and building owners in the commercial sector \$179.4 billion. When this cost is broken down by use, the same five categories described previously are at the top: lighting (\$35.4 billion), space heating (\$27.5 billion), space cooling (\$25.3 billion), ventilation (\$15.9) billion, and water heating (\$7.3 billion). As such, buildings are a prime target for reductions in energy use and increases in energy efficiency.

Current practice in building energy systems is to implement control with low-level controllers (proportional-integral or proportional-integral-derivative) in a decentralized fashion. In some cases, there may be some supervisory control, but mostly the systems operate independently. Because of the physically interconnected and complex nature of building systems, this uncoordinated control can often lead to inefficient solutions as controllers can compete with one another in achieving their desired outputs. Many advanced control strategies have been proposed, with Model Predictive Control (MPC) being a front-runner to address the challenges that building energy systems put forth. While much effort has gone into developing MPC for building energy systems, several difficulties have arisen that have slowed the progress of practical solutions. Specifically, for MPC to perform well, reliable models are needed; however, acquiring accurate models of buildings beyond a handful of rooms or a few pieces of HVAC equipment is rather difficult. And as systems grow in size and number, the complexities of obtaining dynamic models of the respective systems can become prohibitive. Additionally, as systems become more complex, the computation and communication burdens to solve the MPC problem quickly exceed current hardware and infrastructure capabilities. This paper endeavors to investigate the performance gained from using fully dynamic models over steady-state relationships and whether that performance increase, if any, is worth the additional time and effort. First,

a background on controlling energy systems in buildings, MPC, and recent efforts is presented. Then information about the simulation of a building is given, along with how models of this building were generated. Simulation results are then detailed, with a comparison of performance between the dynamic and steady-state implementations. The paper concludes with a discussion of the study's outcome as well as future work.

7.3 Background

Control of energy systems in buildings provides several unique challenges. Buildings consist of numerous interconnected energy systems that affect and depend on one another. For example, in a large building there may be multiple chillers that are used to chill a secondary fluid, such as water. This chilled water is then pumped to various systems and areas of buildings where heat exchangers in air handling units (AHUs) use the chilled water to cool air streams. A network of fans and ducts then deliver the cooled air to the desired locations. The flow of this cooled air into the zones can be controlled by variable air volume (VAV) units, in which there may be an additional heat exchanger that utilizes heated water to reheat the air, if necessary. The heated water for this process is provided by a different set of centralized pumps and heat exchangers. The zones themselves are connected to one another, either by conduction through barriers or shared doorways/open spaces. All these interconnections and couplings can make for complex coordinated control problems.

Providing additional difficulties, nonlinear dynamics and multiple time scales occur across these various building systems. Equipment such as chilled water valves, fans, and dampers can have nonlinear behavior. For example, a nonlinear relationship can exist between air flow into a zone and damper position, dependent upon operating conditions and the damper used (single blade, opposed blade, etc.). End static pressure, often regulated by fan speed, depends on duct efficiencies and losses as well as zone damper positions. As for time scales, changes in damper position in a VAV or fan speed in an AHU are relatively fast

(on the order of seconds) compared to changes in desired chilled water temperatures (minutes) or changes in zone air temperature (hours). There are also slow, overarching changes to take into consideration such as the shift in solar loads as the sun moves throughout the day, gradual changes in outdoor air temperature due to change in weather or seasons, and the deterioration of equipment through use over time. These all contribute to shifting systems behaviors and disturbances that can cause undesired performance in systems. There are also discrete changes to take into account, such as whether an area/room is occupied, how many people are in said room, or changes in real-time pricing of utilities. In addition, the sensors of the system are distributed (not always equally), centralized monitoring is being performed, and devices are driven by localized controls. In addition to all of this, the systems must operate within constraints due to hardware limitations, limited resources, and/or issues of health and comfort.

While the challenges of building control are numerous, one control method that has emerged as a capable solution is model predictive control (MPC). MPC has been chosen as the most appropriate control method in various building thermal control research projects such as Opti-Control in Switzerland [34], Intelligent Buildings and Rational Management of Renewable Energy (MIGRER) in France [35], and MPC for UC Merced Campus in the U.S. [36]. MPC, or receding horizon control, predicts the change in the dependent variables of a modeled system by changing independent variables. Using the current state information, dynamic models of the system, and an objective function, MPC will determine the changes in the independent variables that will minimize the user-defined objective function while honoring given constraints on both dependent and independent variables. Once this series of changes is determined, the controller will apply the first determined control action, and then repeat the calculations for the next time step. Figure 7.3 displays how a typical reference tracking MPC implementation would behave. It can be seen at time k that the controller determines what the predicted output would be along with its op-

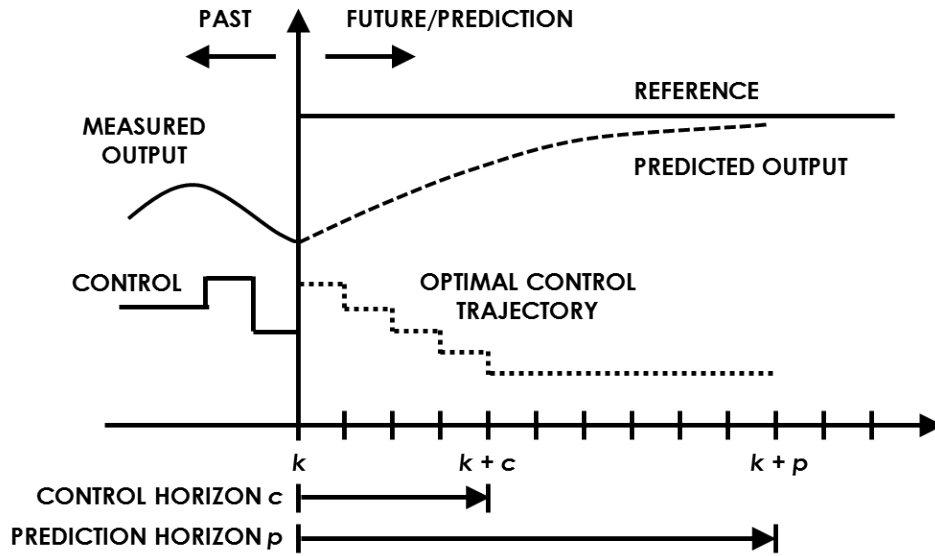


Figure 7.3: How MPC, or receding horizon control, typically functions.

timal control trajectory. After completing the computations, the control would be applied and the system would move on to time $k + 1$, repeating the predictions and optimization with the new measurements. It is important to note the two horizons within MPC: the prediction horizon, which is the length of time for which the system outputs are predicted, and the control horizon, which is the number of control inputs that are determined in the prediction computation. The prediction horizon is often limited by computational capabilities, while the control horizon is selected so that the system dynamics are allowed to diminish [37]. An extensive overview of the recent advances of MPC can be found in [38].

MPC in buildings has been studied mostly in simulation [39, 40, 41, 42, 43, 44, 45, 46, 47, 48], with some experimental efforts [34, 35, 36, 49, 50, 51]. In simulation, MPC has been adapted for controlling building systems such as floor heating [52], water heating [53], cooling [36], and ventilation [54], among others. MPC has shown improved performance over the standard controls that are normally used in buildings. In [42], the authors were able to demonstrate a 13.4% reduction in energy consumption and a 36.7%

increase in thermal comfort with MPC compared with typical PI controllers. This is partially due to the fact that MPC can account for coupling between systems. MPC was applied to zone temperature control in a large university building in [108] and [109], showcasing 29% reduction in energy usage while maintaining the same thermal comfort levels in both studies.

For most applications of MPC for buildings that have been reported in the literature, the objective function that is minimized focuses on energy usage, occupant comfort, or some combination of the two. This can take the form of economic MPC (E-MPC), where the objective function is a linear combination of the monetary cost of building energy consumption [43]. Generally in the applications where E-MPC is used, the amount of energy consumed is minimized while occupant comfort is maintained between upper and lower limits by constraints on the variables. In [46], a linear E-MPC cost function was used that had a time-varying cost of electricity vector for several actuators, which included positioning of blinds, the level of electrical lighting, chiller production, operation of the cooling tower, and heat from radiators. Occupant comfort was maintained by constraints placed on the room temperatures and the lighting levels. This proposed control strategy is able to respond to real-time changes in utility pricing; however, occupant comfort becomes a second priority temperature and lighting levels are constrained within specified limits as opposed to optimizing occupant comfort. Avci, et al. [55], also simulated a MPC controller that determined on/off control actions as well as reference temperatures for AC units to different zones. Their optimization leveraged real-time pricing of utilities and occupant preference to minimize usage of utilities and the difference between reference temperatures and room temperatures (error). In this effort, the authors developed an algorithm to choose the temperature reference setpoint based on the current utility cost and the occupants desired temperature range. While this helps to minimize cost to the user as well as maintain comfort within the user's defined range, this method does not provide a

minimal economical solution in that the cost depends on the user's own comfort decisions, which can either vary greatly or require updating with weather or seasonal changes.

The experiment detailed in [49] had a similar cost function, except that it was optimizing the cost of energy usage by manipulating the temperature setpoint for the water coming from the cooling tower as well as the temperature and mass flow setpoints of the water coming from the chillers. Again, occupant comfort was just maintained between upper and lower limits and not optimized. Papers [39, 41, 43, 50] all employed the linear E-MPC objective function while actuating heat flux, input power to a heat pump compressor, indoor air temperature setpoint and a thermal energy storage system, and temperature setpoints, respectively. The 4 previous works place constraints on the MPC optimization with temperature limits relating to comfort ranges, placing the optimization focus on control action and utility price. Oldewurtel performed simulations on six different combinations of subsystems including actuated blinds, electric lighting, radiators, mechanical ventilation, floor heating, evaporative cooling, and chilled ceilings while maintaining comfort between temperature limits [44]. Corbin, *et al.* [45] detailed two case studies in which the first used a linear E-MPC cost function while the second minimized the sum of electricity used by all HVAC equipment with a comfort penalty. This comfort penalty was defined as an area-weighted sum of the number of zone occupied hours outside of a predicted mean vote (PMV) threshold of ± 0.5 . PMV is an index that determines the thermal comfort of an average individual dependent upon a variety of factors, including air temperature, relative humidity, relative air velocity, metabolic rate, clothing insulation level, work output, and several other variables [4]. Corbin, *et al.* worked to optimize occupant comfort in their second case-study, but do so with the focus of reducing the cost of energy usage, neglecting the economic aspect associated with occupant comfort and productivity.

The authors of [47] also included a discomfort cost with their monetary energy cost that was based on different lower and upper thermal limits; however, the physical mean-

ing of this discomfort cost is arbitrary as the cost increases to unity until the temperature limits are exceeded and then becomes significantly large, not following any physical or measured relationship. The cost function in [48] included regulation of occupant comfort based on PMV, though it took the quadratic cost form, with the first term being the difference between the predicted PMV of the zone and the PMV setpoint for the zone, quantity squared, multiplied by a weighting factor. The second term consisted of the square of the change in control action, or increment, multiplied by a weighting factor. With this form, the MPC will balance maintaining the desired zone PMV while limiting large control action rates, in this case changes in water flow and air velocity. This will help keep occupants comfortable but the economic cost of the control actions is not accounted for. Morosan, *et al.* [42] used a linear objective function that penalizes the error between the predicted room temperature and the future room temperature reference as well as the energy usage to condition the room. In their efforts, comfort is accounted for as a comfort index that acts as a penalty when the room temperature does not meet its setpoint. However, in the presented simulations the temperature setpoints are arbitrarily chosen and not dependent on any comfort information. Additionally, this method, like the previous ones mentioned, does not account for the economic aspect of occupant comfort. One objective function from the literature that appears more unique than others was used in [51]. This objective function consisted of three different linear terms: 1) a weighting coefficient multiplied by the predicted percentage of dissatisfied (PPD) people, which PPD can be calculated from PMV, 2) a weighting coefficient multiplied by the summation of cost of energy consumed by the heating and cooling devices, and 3) a weighting coefficient multiplied by the summation of the green house gas intensities of the various energy sources (electricity and natural gas). This objective function displays the power of MPC to determine optimal control actions with respect to a user's desired metrics, in this case occupant comfort, monetary cost of energy, and environmental impact of energy sources. While providing great

flexibility in allowing the building operator to prioritize the three metrics with the weighting factors, the economic impact of the three areas can not be optimized due to differing units and arbitrary weights. Overall, previous methods have accounted for the economic cost of energy usage and/or attempted to maintain occupant comfort through optimization constraints or optimizing comfort itself, but none have accounted for the economic aspect of comfort on occupant productivity alongside utility costs.

The purpose of this research effort is to present a control method that minimizes the economic cost of both utilities and occupant comfort. Occupant comfort is monetized through the loss of productivity due to an occupant's discomfort. Additionally, this paper investigates the performance gains of implementing a full MPC controller, computing optimal control actions over a horizon based on dynamic models, compared to a steady-state optimal control method, that makes a single prediction at each time step deciding the optimal control actions based on steady-state relationships. The steady-state optimal control method leverages the fact that many of the control inputs in buildings (damper positions, fan speed, chilled water flow rates) change relatively quickly compared to the system outputs and disturbances (zone temperatures, outdoor air temperature, outdoor relative humidity, etc.); thus, the system reaches a pseudo-steady-state by the next optimization timestep with little change in the environmental variables. In doing so, the implementation of the controller is simplified and the computation time/burden is reduced. The comparison of the two methods will help to determine when the full controller is more beneficial versus the simplified steady-state method.

7.4 Description of Simulated Design and Implementation of Control

7.4.1 The Utilities Business Office

The Utilities Energy Management (UEM) Office at Texas A&M University granted the authors access to the Utilities Business Office (UBO), a small office building on campus.

Data was collected about the building through the building energy management system for the purpose of modeling and eventual implementation and testing of advanced controllers. The UBO is a rectangular, single-story building consisting of 11 zones, 10 of which are actively controlled. The building's layout can be seen in Figure 7.4. This paper focuses solely on the cooling aspects of the system due to: 1) the fact that the majority of the year is spent in cooling mode, and 2) limiting to one operation mode will aid in identifying performance differences between dynamic and steady-state models.

The UBO is serviced by a single, rooftop air handling unit (AHU). The AHU consists of a variable air volume (VAV) fan, a chilled water coil, an outdoor air damper, a return air damper, a discharge air temperature sensor, and an end static pressure sensor. The organization of these components can be seen in Figure 7.5. During normal operation, the VAV fan works to maintain an end static pressure given by a pressure demand calculation

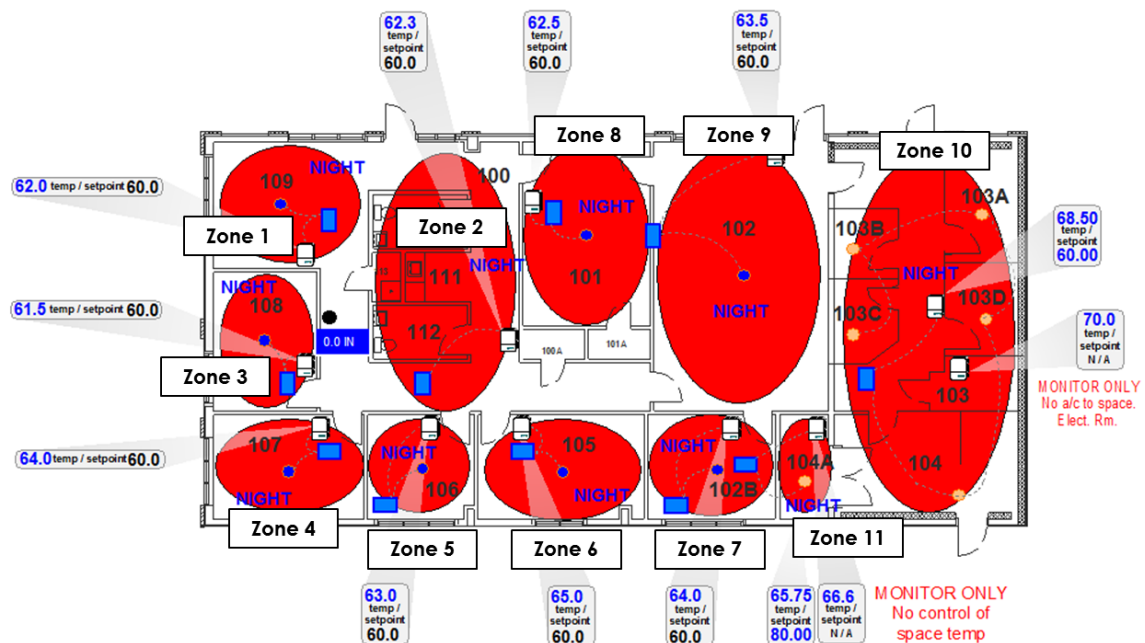


Figure 7.4: Zone layout for the Utilities Business Office (UBO) at Texas A&M University.

dependent upon the individual damper positions of the zone terminal boxes, moving the air through the system. The pressure demand calculation is the feedback signal for a proportional-integral-derivative (PID) loop that produces a reference equal to the desired end static pressure. This reference is fed to another PID loop that actuates the fan speed to maintain the given end static pressure. For conditioning the air, chilled water provided by central plant is passed through the AHU's chilled water coil to lower the temperature of the moving air as well as help reduce the air's humidity. The amount of chilled water passing through the coil is controlled by a valve which is actuated by a PID control loop. This control loop is driven by a difference in cooling demand setpoint and a cooling demand calculation that is a weighted combination of the average and maximum cooling loopout values from the individual zone control loops. This loop's output is the input (the desired discharge air temperature) for a second loop that actuates the chilled water valve.

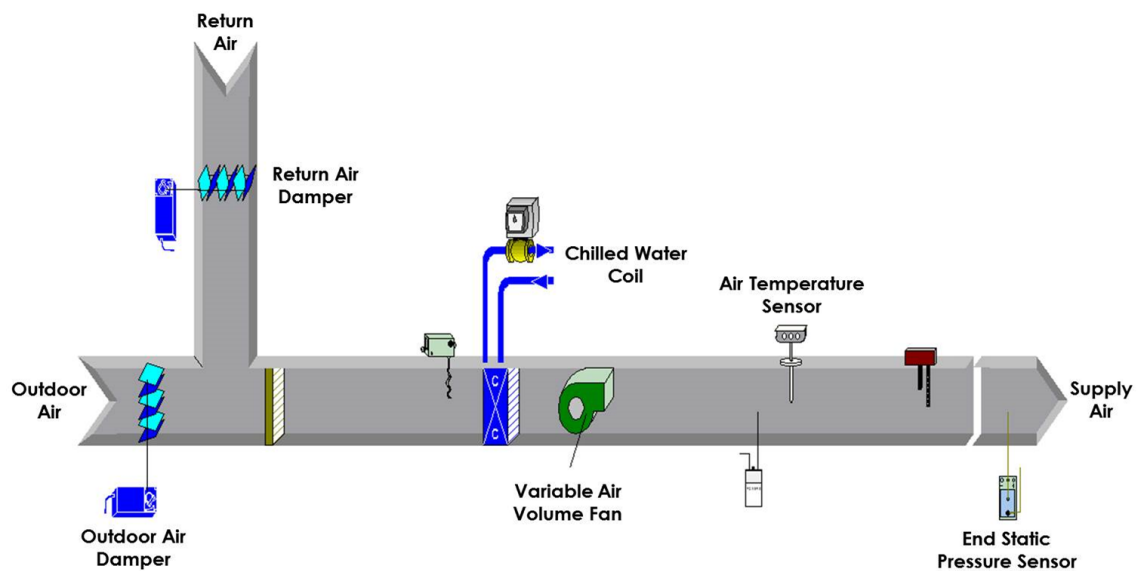


Figure 7.5: Rooftop Air Handling Unit (AHU) for the Utilities Business Office (UBO) at Texas A&M University.

Figure 7.6 details the chilled water loop. A chilled water pump works to maintain a specific supply pressure to provide the required chilled water flow for the AHU. Worth noting is that the chilled water supply and return temperatures are available to measure within the energy management system.

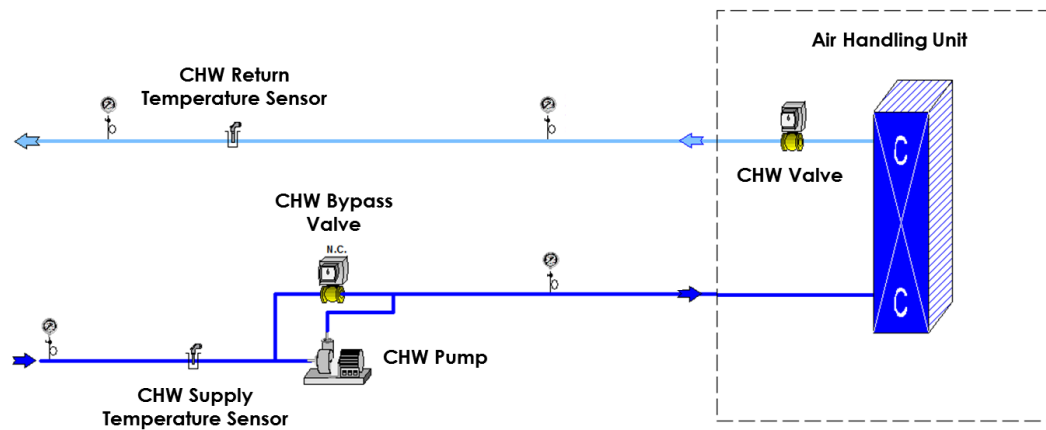


Figure 7.6: Chilled water loop for the Utilities Business Office (UBO) at Texas A&M University.

The UBO consists of 11 zones, 10 of which are actively controlled. Each controllable zone is serviced by a VAV terminal box equipped with hot water reheat capabilities, an example of which is shown in Figure 7.7. The flow of conditioned air into the room is regulated by a damper in the terminal box whose position is determined by a PID control loop. The error signal for the control loop is the difference between the respective room temperature setpoint and the measured room temperature. The room temperature setpoint is determined by whether the room is occupied or unoccupied as well as whether the zone is in heating mode or cooling mode. A deadband control method is employed such that if the room is occupied, the zone VAV will heat the room to 70°F or cool the room to 74°F.

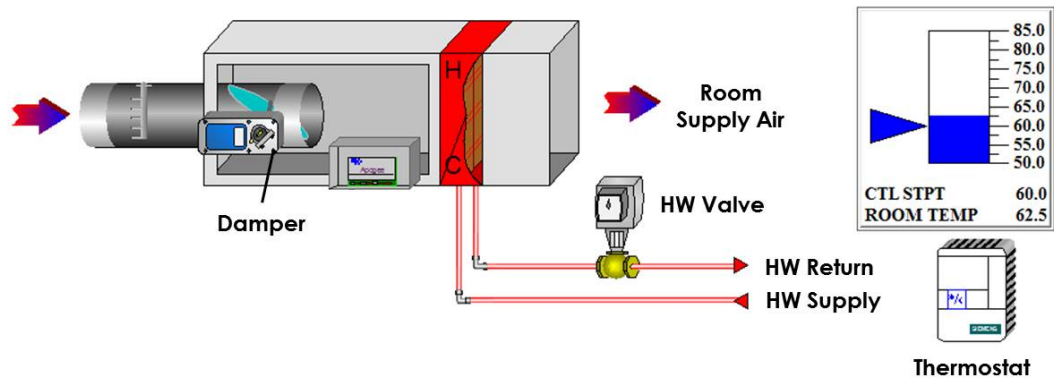


Figure 7.7: Variable Air Volume (VAV) box in the Utilities Business Office (UBO) at Texas A&M University.

If the room is unoccupied, the VAV will heat the room to 60°F or cool the room to 85°F. If the occupancy sensors in all the rooms read unoccupied, then the main AHU will turn off and the room temperatures will freely fluctuate.

In this research effort, the scope of the supervisory controller was limited to determining setpoints for the AHU and the individual zones. Specifically, for the AHU two setpoints are optimized: the end static pressure setpoint and the discharge air temperature setpoint. For each of the zones, the zone temperature setpoint is optimized. The objective functions used in the optimization are economic functions derived in a previous work for the UBO. They are briefly described in the following subsections, but additional details about the derivation of the objective functions can be found in [110].

7.4.2 AHU Fan Economic Objective Function

As described above, the fan in the AHU works to maintain an end static pressure in the duct to move the required amount of air to condition the individual zones. To accomplish this, the fan motor requires electricity and there is a cost associated with the power used. A simple way to measure the power consumed by a fan motor would be use a power meter

on the electrical lines to the fan and measure the power consumed; however, power meters are not often included in existing building energy systems and integrating a new one has its own associated costs to install and connect the device to the network. Therefore, the power consumed by the fan was instead determined by other available data, specifically the change in pressure across the fan and the volume flow rate of air being moved by the fan. With these two values, the work being performed by the fan on the air can be determined and converted into a measure of power, as power is work over time. The objective function for the fan is defined as follows:

$$J_{fan} = u_{fan}^T T_{fan} \quad (7.1)$$

$$u_{fan} = P_{EDS}^* \quad T_{fan} = \frac{0.1175 \cdot C_{elec} \cdot q_{AHU} \cdot t_s}{1000 \cdot \mu_f \cdot \mu_b \cdot \mu_m}$$

where P_{EDS}^* is the end static pressure setpoint, C_{elec} is the rate of electricity cost [\$/kWh], q_{AHU} is the total air flow through the AHU, t_s is the sampling time of the supervisory controller, and μ_f , μ_b , μ_m are efficiencies related to the fan blade, belt, and motor respectively. The asterisk * denotes a setpoint.

7.4.3 AHU Chilled Water Economic Objective Function

In order to condition the air in order to provide cooling to the zones, chilled water is passed through a chilled water coil over which air moves in the AHU. There is an associated cost with the production of this chilled water which comes from a central chiller. This cost is determined with Equation 7.2:

$$J_{CHW} = u_{CHW}^T T_{CHW}$$

$$u_{CHW}^T T_{CHW} = \alpha \cdot q_{max} \cdot \rho \cdot c \cdot \Delta T_{h2o} \cdot t_s \cdot 0.00341214 \left[\frac{mmBtu}{kWh} \right] \cdot \frac{3682}{241} \left[\frac{\$}{mmBtu} \right]$$

$$\alpha = (3.021T_{DA}^{*2} - 109.4T_{DA}^* + 1002) \quad (7.2)$$

where α is a fitted relationship between the discharge air temperature setpoint (T_{DA}^*) and the chilled water valve position, q_{max} is the maximum flow through the valve (at 100% opening), ρ is the density of water, c is the specific heat of water, ΔT_{h2o} is the change in temperature of the supply and return chilled water, t_s is the sample time of the supervisory controller in hours, 0.00341214 is a conversion factor from kWh to mmBtu, and the last term is the economic cost ratio for the UBO of the consumed chilled water.

7.4.4 Zone Occupant Comfort Economic Objective Function

The last economic objective function that was developed for the UBO is the cost of occupant discomfort within each zone. This is determined by a relationship between predicted mean vote (PMV) [4], a loss of productivity (LOP) due to discomfort [10], and an occupants salary. In this work, an approximation of PMV was found to be accurate and reduced computation time. A fit was generated that varies based on zone air temperature and zone relative humidity, expressed as:

$$PMV = 0.5542 \cdot Rh + 0.23 \cdot T_{zone}^* - 5.44 \quad (7.3)$$

where Rh is the zone relative humidity and T_{zone}^* is the zone temperature setpoint determined by the optimization. The comfort economic objective function used is shown in Equation 7.4:

$$J_{room} = e^T R \quad (7.4)$$

$$e = T_{zone} - T_{user,zone}^* \quad R = \left[\frac{\partial PMV}{\partial t_a} \right] \cdot \left[\frac{\partial LOP}{\partial PMV} \right] \cdot \left[\frac{\partial \beta}{\partial LOP} \right]$$

where T_{zone} and $T_{user,zone}^*$ are the zone temperature and user-defined zone temperature

setpoints, respectively. The sensitivities are determined to be:

$$\left[\frac{\partial PMV}{\partial t_a} \right] = 0.23 \quad (7.5)$$

$$\left[\frac{\partial LOP}{\partial PMV} \right] = b_1 + 2b_2 PMV + 3b_3 PMV^2 + 4b_4 PMV^3 + 5b_5 PMV^4 + 6b_6 PMV^5 \quad (7.6)$$

$$\left[\frac{\partial \beta}{\partial LOP} \right] = \left(\frac{p_{year} \cdot t_s}{52wks \cdot 40hrs} \right) \quad (7.7)$$

where PMV is the predicted mean vote, β [\$] is the productivity lost in wages, p_{year} [\$] is an occupant's annual salary, and t_s is the sampling time of the supervisory controller, in hours. For details on the derivation of the sensitivities see [110]. The cost calculation assumes that if more than one occupant is in a zone, the sum of the occupant's salaries is used for p_{year} and that a standard 40 hour work week is used. It is worth noting that there are two separate zone temperature setpoints: one defined by the optimization and the other defined by the user (occupant, building technician, etc.). The purpose of this is to allow for some level of occupant feedback, if necessary, on the zone temperatures. The optimization will find the optimal economic compromise between the loss of productivity optimal temperature and the user preference. For the simulations reported in this paper, the user-defined zone temperature setpoint was set equal to the LOP optimal temperature ($PMV = -0.25$). The coefficients b_1, \dots, b_6 come from the regression fit of the LOP curve [10]. There are two sets of the coefficients: one for the cold side of comfort ($PMV < -0.5$) and one for the warm side of comfort ($PMV > 0$). In between -0.5 and 0, the loss of productivity was determined by Roelofsen was determined to be zero [10]. They are listed in table 7.1.

Table 7.1: Regressions for Loss Of Productivity fit of PMV from [10].

Regression Coefficients	Cold Side of PMV Comfort Zone	Warm Side of PMV Comfort Zone
b_0	1.2802070	-0.15397397
b_1	15.995451	3.8820297
b_2	31.507402	25.176447
b_3	11.754937	-26.641366
b_4	1.4737526	13.110120
b_5	0.0	-3.1296854
b_6	0.0	0.29260920

7.4.5 Simulation Design

A model of the UBO was developed in EnergyPlus [89]. While EnergyPlus excels at modeling and simulation, it is not immediately accessible for controller development and implementation. As such, the controllers for the AHU and the zone level VAVs were created in MATLAB. To enable co-simulation between EnergyPlus and MATLAB, two programs were used. The first was MLE+ [90], an open-source MATLAB toolbox for creating the necessary configuration files and providing functions to connect EnergyPlus and MATLAB with an easy-to-use graphical interface. The second was the Building Controls Virtual Test Bed (BCVTB) [91]. MLE+ utilizes the BCVTB as the communication backend between EnergyPlus and MATLAB, providing the co-simulation functionality.

The individual zones were setup as defined by the building layout (Figure 7.4). The AHU and zone VAVs were added using EnergyPlus' HVACTemplate objects. A central electric chiller was also added to supply chilled water to the AHU. The chilled water output temperature is regulated to 45°F, the same as the supply chilled water temperature for the UBO. Currently, EnergyPlus does not offer modeling of pressure with variable air volume systems. To simulate the UBO's control and physical limitations of end static pressure and flow in the AHU, it was assumed that the dynamics of the fan speed and

end static pressure were fast enough compared to the simulation timestep (1 minute) to be considered instantaneous. Additionally, it was assumed that the fan would supply the requested end static pressure, constrained by the physical limitations of the AHU and ducting. To determine this constraint, data from the real UBO building was analyzed and a maximum possible work performed by the fan was calculated (910 W). During the optimizations, the constraint is calculated by using Equation 7.9. To determine the total volume flow through the AHU, individual models of the zone VAVs were generated from data based on VAV damper position and end static pressure. The VAV data was fitted with the following structure:

$$q_{VAV} = (a \cdot u_d^2 + b \cdot u_d) \cdot \sqrt{P_{EDS}} \quad (7.8)$$

where q_{VAV} is the air flow through the VAV (cfm), a and b are identified coefficients for each zone VAV, u_d is the damper position (%), and P_{EDS} is the end static pressure (in. h₂O). The flows from the VAV models are summed to obtain the total flow through the AHU, assuming minimal duct losses. The constraint on the end static pressure can be written as follows:

$$\frac{0.1175 \cdot q_{AHU} \cdot P_{EDS}}{\mu_f \cdot \mu_b \cdot \mu_m} \leq 910 \quad (7.9)$$

Thus, the optimization will only ever choose an end static pressure setpoint that the system is physically capable of. This end static pressure is then passed to the VAV models which, along with the commanded damper positions, produce individual zone air volume flows.

The overall control hierarchy can be seen in Figure 7.8. The supervisory controller supplies the zone temperature setpoints (T_{ZONES}^*) to the respective PID reference inputs and the end static pressure setpoint (P_{EDS}^*) to the VAV models. The discharge air tem-

perature setpoint (T_{AHU}^*) is supplied directly to EnergyPlus as the control for the chilled water valve is implemented using an appropriate setpoint manager within EnergyPlus. For the MPC controller, these setpoints are optimized every 5 minutes of simulation time over a prediction horizon of 30 minutes. The steady-state optimal control method optimizes the setpoints every 15 minutes. The zone temperature error is fed to the cooling PID controller which outputs a desired percentage of maximum flow setting (0% to 100%). This signal is passed through block G which maps the maximum flow percentage to the range of minimum flow to maximum flow. If the minimum flow setting for a zone is zero, then the signal remains unchanged. The desired flow percentage is then passed to the flow PID controller which then determines a desired damper position. This damper position is then used in the VAV models as previously described. The output from the VAV models (desired zone air volume flows) are converted to air flow fractions and sent to zone VAVs in EnergyPlus where the room dynamics are simulated for one simulation timestep.

7.4.6 Zone Temperature Models

For the supervisory controllers, models are necessary to predict the room temperatures as the setpoints are optimized. A previously developed modeling algorithm was employed

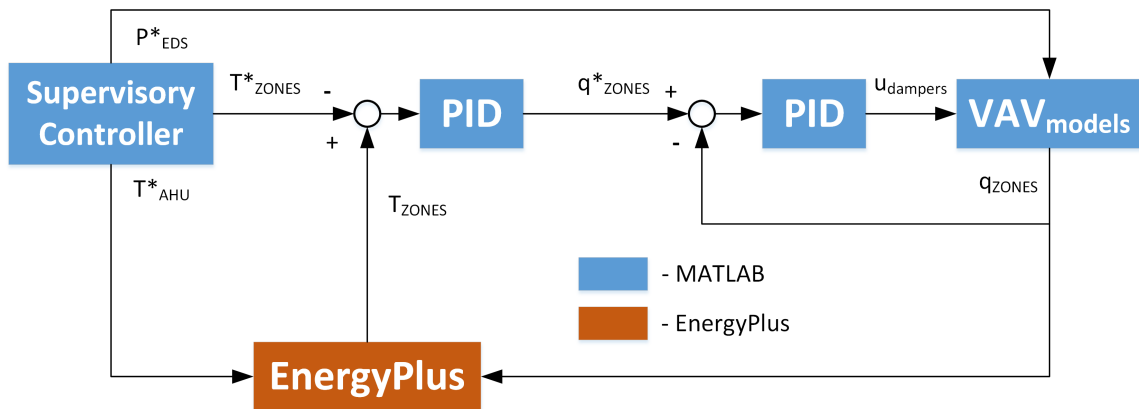


Figure 7.8: Control hierarchy used in simulation of the UBO.

to generate models for the individual zones. The full details of this process are beyond the scope of this publication, but more information can be found in [93]. Briefly, data from the EnergyPlus simulation is analyzed to determine significant coupling interactions between zones. The temperature of these zones along with other inputs such as outdoor air temperature, outdoor air relative humidity, AHU discharge air temperature, end static pressure, and zone temperature setpoints are used as inputs to ARX, ARMAX, and Box-Jenkins modeling methods with the output being the respective zone temperatures. The best fitting model is selected and then the individual models are combined into a centralized model of the entire system. Predictions from this centralized model are then used by the supervisory controller to optimize the UBO's 12 setpoints (discharge air temperature setpoint, end static pressure setpoint, and 10 zone temperature setpoints) to minimize the economic cost functions previously described. The identified models can be written in a discrete, linear state-space form as follows:

$$\begin{aligned}x_i(k+1) &= A_i x_i(k) + B_i u_i(k) + K_i e_i(k) \\ y_i(k) &= C_i x_i(k)\end{aligned}\tag{7.10}$$

where x_i is the state vector at a given timestep k , u_i is the input vector corresponding to each model, and y_i denotes the zone air temperature. Normally $e_i(k)$ is considered as a random component representing a white-noise disturbance. In this implementation, $e_i(k)$ represents the error between the previous prediction and the current measured value, relying on the assumption that the error is approximately Gaussian. The prediction results from the dynamic model for Zone 4 are shown in Figure 7.9. The 30 minute ahead prediction recreates the measured temperature fairly accurately, with only minor differences. The other zone models show similar predictive capabilities.

The inputs and outputs provided to the modeling algorithm for each zone are listed in

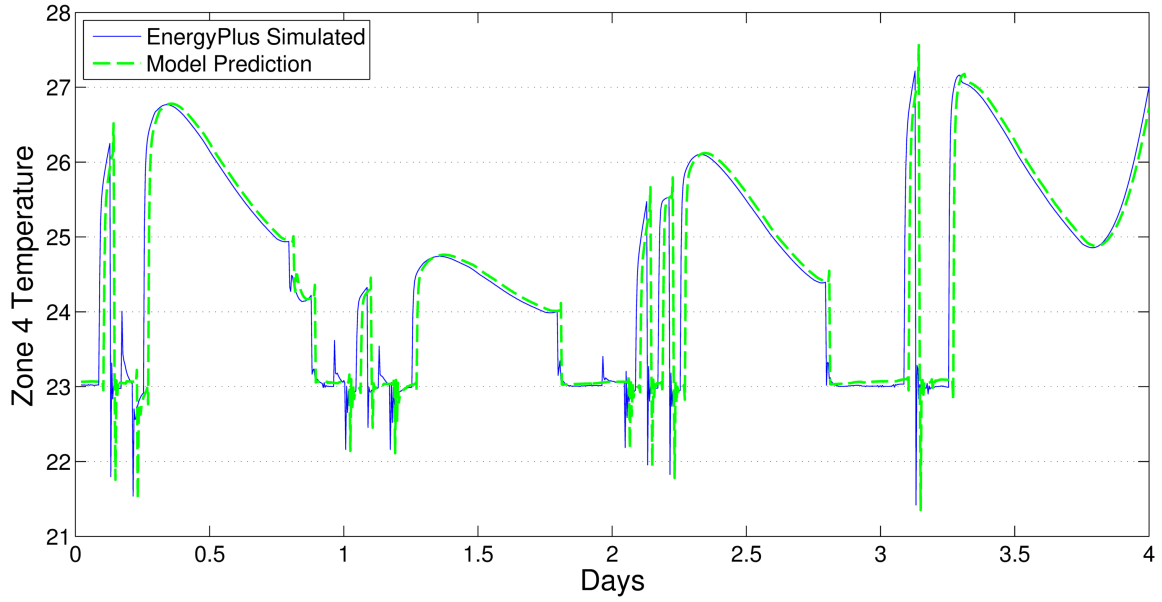


Figure 7.9: Zone 4 temperature prediction 30 minutes ahead using identified models.

Table 7.2. Because the last input includes the output of the models, the identified models required modification so that the output is not part of the input vector. This is accomplished by subtracting the model output from the input vector and substituting that input vector back into the identified state-space model, resulting in modifications to the state matrix A and the error matrix K . For specific details on this formulation, see [93].

Models were identified for two operational cases: 1) where the zone VAV damper

Table 7.2: Inputs and outputs for the generated models.

Model Input	Model Output
T_{OA}	$T_{zone,i}$
Rh_{OA}	
$T_{zone,i}^*$	
T_{AHU}^*	
P_{EDS}^*	
$T_{zone\ dist.,j} - T_{zone,i}$	

still had actuator range (i.e. the damper was not fully open), and 2) where the zone VAV damper is fully open. The separate models were necessary as the effect of the model inputs varies greatly between the two cases. In case 1, the effect of T_{AHU}^* and P_{EDS}^* are minimal compared to $T_{zone,i}^*$. This is due to the fact that the VAV damper is actuated by a PID controller with $T_{zone,i}^*$ as the reference. The PID controller is able to reject changes in T_{AHU}^* and P_{EDS}^* by changing the damper position to achieve $T_{zone,i}^*$. However, in case 2 when the damper is fully open, T_{AHU}^* and P_{EDS}^* become the inputs of significance as they directly effect the zone temperature, determining the amount and the temperature of the incoming conditioned air. For the dynamic MPC controller, the models were used in the state-space form described by Equation 7.10. For the steady-state optimal control method, the steady-state outputs were calculated as:

$$y_i(k) = [C_i(I - A_i)^{-1}B_i] u_i(k) + e_i(k) \quad (7.11)$$

It was then necessary in the simulation to determine when each model should be used. This decision was made based off of two conditions. The first determined if the predicted zone temperature using the case 1 models was greater than the prediction from the case 2 models. This condition served to verify whether the predicted temperature from the case 1 models was currently achievable with the state of the AHU. If the case 1 predicted temperature was lower than the case 2 predicted temperature, then the VAV wouldn't be able to achieve the case 1 predicted temperature with the current T_{AHU}^* and P_{EDS}^* values. The second checked if the current VAV damper position was less than 95%, or in other words if the VAV still had actuator range of the damper. The value of 95% was used as opposed to 100% to serve as a threshold and help prevent the system from oscillating between cases. If both these conditions were true, then the case 1 models were used; otherwise, the case 2 models were used.

7.4.7 Summary of Optimization

For the implementation proposed in this paper, the centralized objective function that is minimized can be written as:

$$\min_{[T_{zone,i}^*, T_{DA}^*, P_{EDS}^*]} = \sum_{j=1}^m \left(J_{fan,j}(P_{EDS,j}^*) + J_{CHW,j}(T_{DA,j}^*) + \sum_{i=1}^n J_{zone,i,j}(T_{zone,i,j}^*) \right) \quad (7.12)$$

where m is equal to the length of the prediction horizon and n is equal to the number of zones. The optimization is subject to the constraints:

$$\begin{aligned} \frac{0.1175 \cdot q_{AHU} \cdot P_{EDS}^*}{\mu_f \cdot \mu_b \cdot \mu_m} &\leq 910[W] \\ \Delta T_{DA}^* &\leq 0.5 \\ \Delta P_{EDS}^* &\leq 0.15 \end{aligned} \quad (7.13)$$

where the first inequality is the constraint on the work done by the fan described previously in Equation 7.9, and the second and third inequalities are constraints placed on the change in the AHU discharge air temperature setpoint and the end static pressure setpoint. The second and third constraints are meant to prevent an drastic changes that could place undue stress on the equipment. For the EnergyPlus simulations where the mass flow rate of the chilled water is available, the chilled water cost function was modified. A specific fit for the simulated chilled water valve was generated between the discharge air temperature setpoint and the chilled water mass flow rate. This fit was used in the chilled water cost function, redefined from Equation 7.2 as:

$$J_{CHW} = \alpha \cdot c \cdot \Delta T_{h_2o} \cdot t_s \cdot 0.00341214 \left[\frac{mmBtu}{kWh} \right] \cdot \frac{3682}{241} \left[\frac{\$}{mmBtu} \right] \quad (7.14)$$

$$\alpha = \left(\frac{-332.9 \cdot T_{DA}^{*2} + 1.089e4 \cdot T_{DA}^* - 6.802e4}{T_{DA}^{*2} + 7787 \cdot T_{DA}^* - 7.606e4} \right)$$

The MPC simulations used prediction horizons of 30 minutes, computed in 5 minute intervals. The steady-state control method simulations performed the optimization every 15 minutes, a time that allows for most of the dynamics to dissipate. The EnergyPlus simulation is run at 1 minute timesteps, which is the smallest timestep that the program allows. The MPC simulations assumed perfect knowledge of future outdoor weather conditions and occupancy profiles. This data could be estimated with weather forecasts and occupant's schedules.

7.5 Results

Simulations were performed with both the steady-state optimal control method and the MPC controller in several different operational cases. These cases include constant occupancy throughout the workday, random occupancy with standard zone temperature setbacks (to simulate a more realistic working environment), random occupancy with optimal temperature setbacks computed by the minimization function, and a situation in which one room is prioritized over others, or a Very-Important-Person (VIP) case. The performance of both the algorithms are compared to the that of the current control strategy implemented in the UBO with temperature setpoints that produce a PMV equal to -0.25.

7.5.1 Constant Occupancy Simulation

In this simulation the zones are assumed to be constantly occupied between the hours of 7 AM and 6 PM. The user-defined temperature setpoints are equal to the air temperature that will give a PMV of zero based on the current zone's relative humidity. Figure 7.10

shows how the zone temperatures vary throughout the day using the steady-state control method. Initially, the temperatures drop drastically as the rooms become occupied at 7 AM and the setpoints are lowered. Then majority of the rooms come up to a temperature of approximately 22.5 °C, which is associated with a PMV of slightly less than zero. Some variations in the room temperatures are observable, though the zones approximately stay within the -0.5 to 0 PMV range. These variations are due to the couplings that were identified in the modeling process between the different rooms. The optimization may lower some room setpoints to assist with the cooling of other rooms. This is most notable after approximately 2:30 PM when the greatest cooling demand occurs during the day. Specifically, zone 1's damper has become fully open. In order to ensure Zone 1 receives the cooling it needs, Zone 6's temperature is lowered. The discharge air temperature is also lowered significantly after 4 PM (shown in Figure 7.13). In response to the cooler discharge air temperature, many of the other zones reduce their damper openings resulting in an increase in the end static pressure (shown in Figure 7.15).

Compared to the steady-state control method, the MPC controller exhibited fewer variations in zone temperatures throughout the day, shown in Figure 7.11. The zone temperatures begin to drop at 6:30 AM as the prediction horizon of the MPC controller enables automated pre-cooling of the building. This pre-cooling action allows for the zones to be much closer to their optimal loss of productivity values at the start of occupancy at 7 AM. Again, around 3 PM during the highest cooling demand of the day, Zone 1 can be seen experiencing a slight rise in temperature. During this period, Zone 1's damper is fully open, relying on the surrounding zones as well as the temperature and flow rate of the incoming conditioned air to maintain the zone's temperature. Figure 7.12 shows the increase in Zone 1's temperature (solid blue) as well as the 0 and -0.5 PMV thresholds corresponding to the limits of zero loss of productivity (dashed red and dashed green, respectively). Zone 1's temperature rises slightly above the 0 PMV limit, with the greatest difference equaling

approximately 0.3 °C. This time above the 0 PMV threshold is due to the MPC controller determining the optimal setpoints that minimize the trade-off between increased comfort and increased cooling costs.

Overall, the two methods show similar strategies with the discharge air temperature (Figures 7.13 and 7.14) and the end static pressure (Figures 7.15 and 7.16). One main difference is that the MPC controller employs a higher end static pressure in the beginning of the simulation while the steady-state method has a rise in end static pressure near the end of the workday. Also, the steady-state method tends to display larger changes in the control variables compared to the MPC controller, which could be due in part to the difference in optimization timesteps (15 minutes vs 5 minutes, respectively).

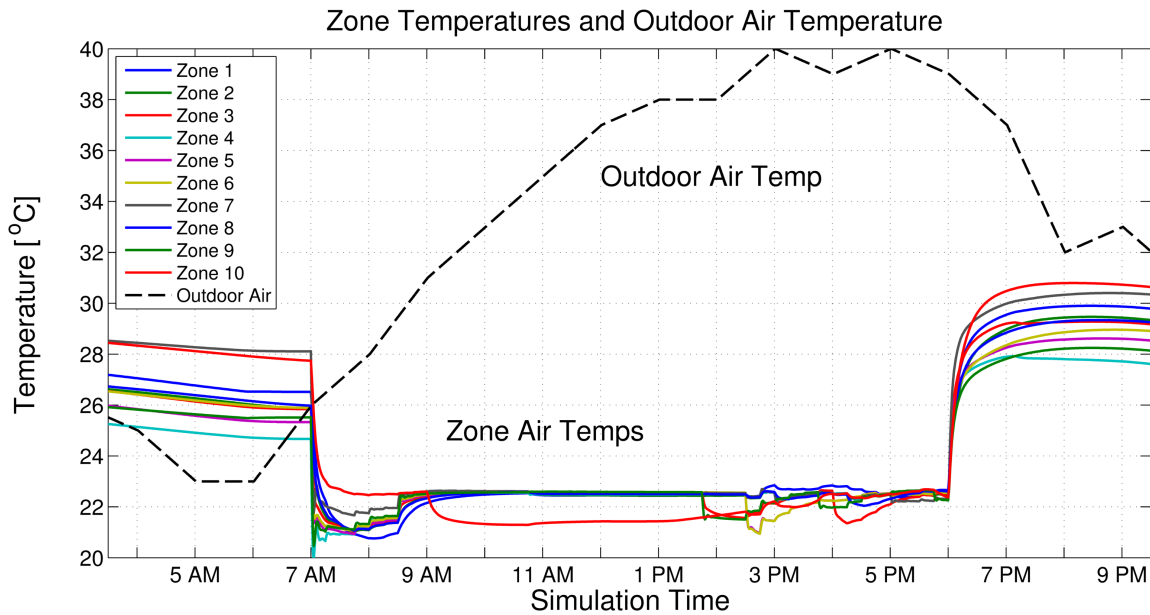


Figure 7.10: Zone temperatures for the steady-state method with constant occupancy.

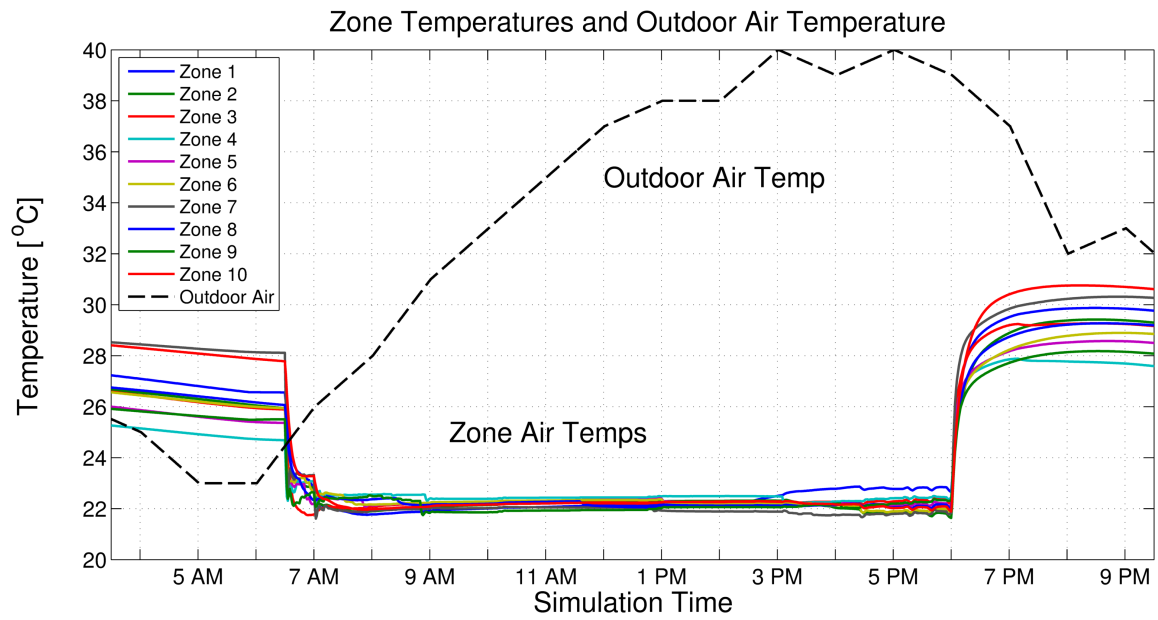


Figure 7.11: Zone temperatures for the MPC controller with constant occupancy.

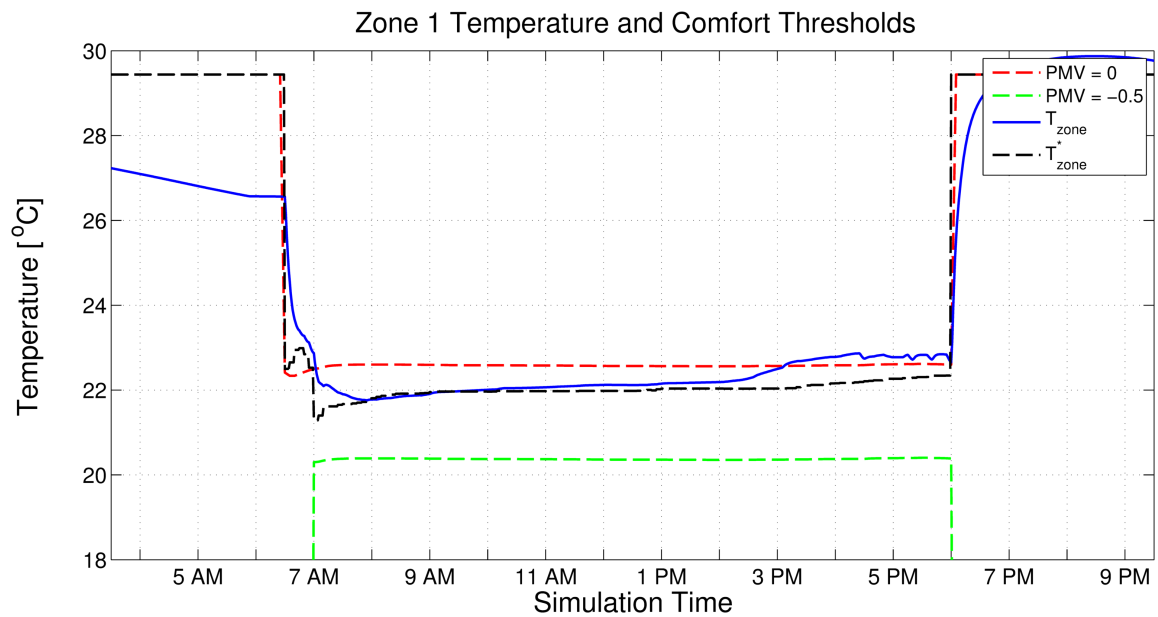


Figure 7.12: Zone 1 temperature for the MPC controller.

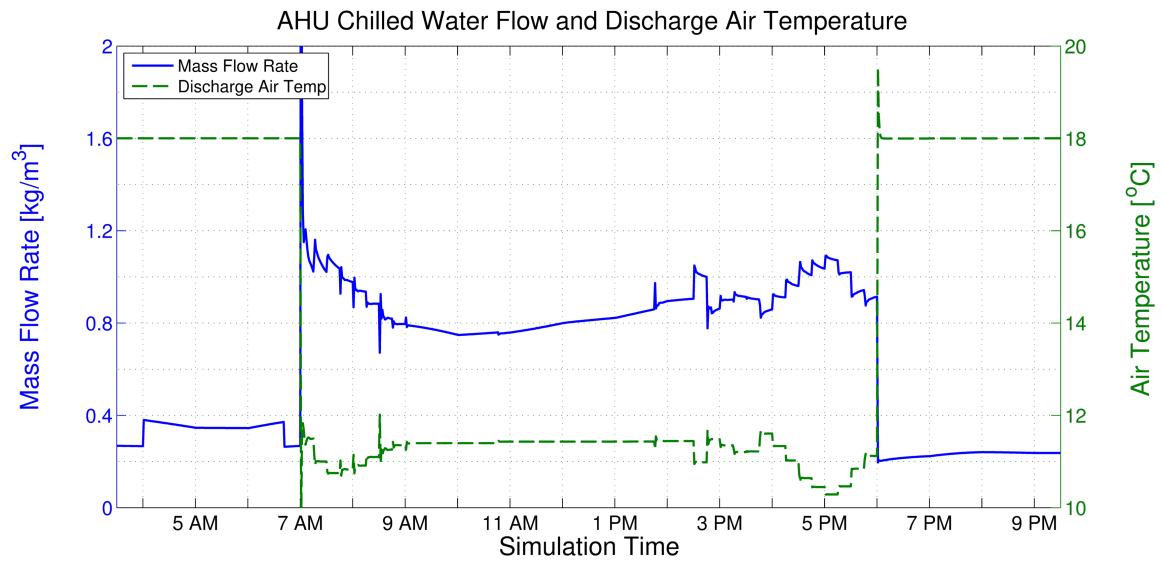


Figure 7.13: Mass flow of chilled water and the discharge air temperature for the steady-state method.

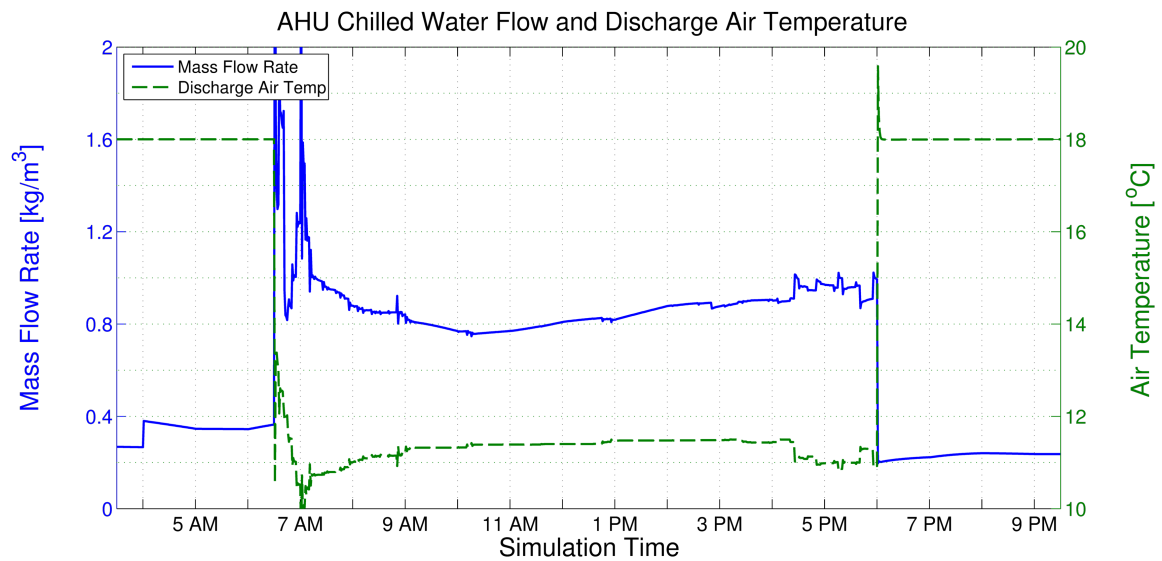


Figure 7.14: Mass flow of chilled water and the discharge air temperature for the MPC controller.

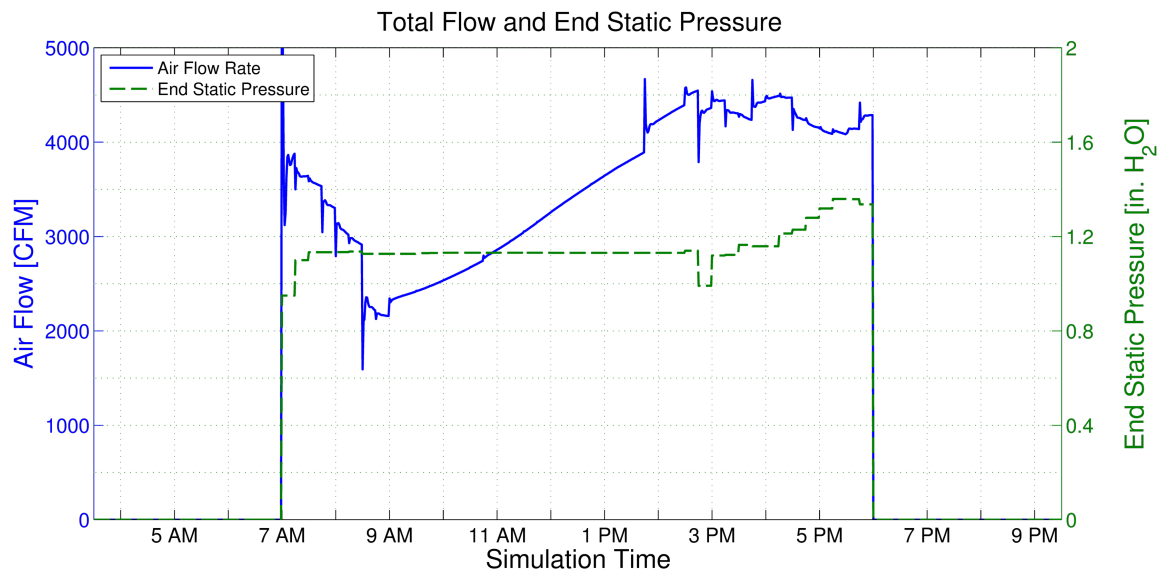


Figure 7.15: AHU air flow and end static pressure for the steady-state method.

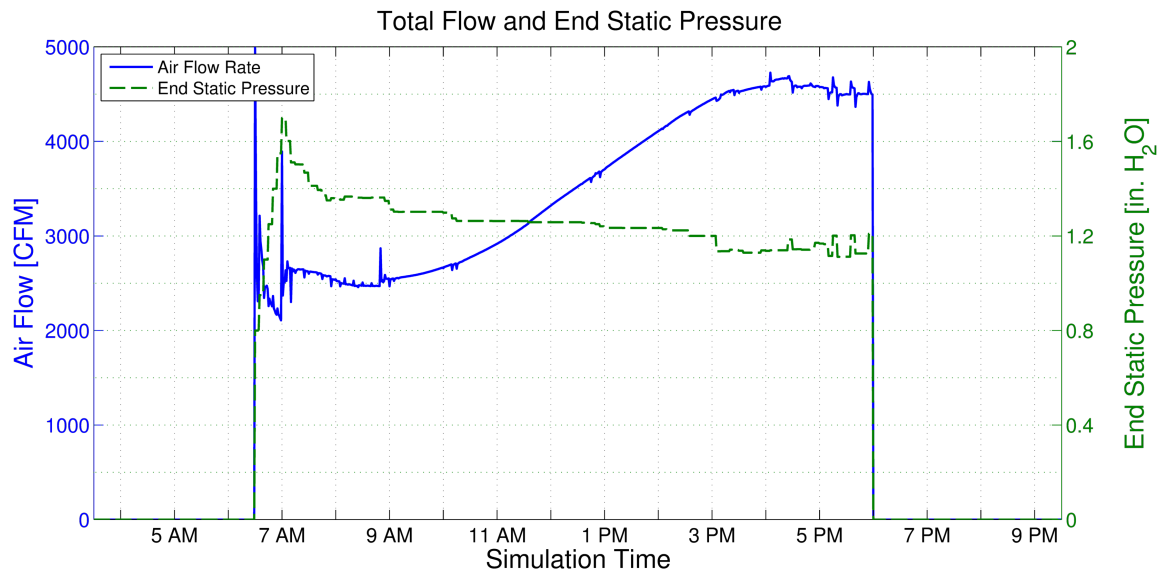


Figure 7.16: AHU air flow and end static pressure for the MPC controller.

The economic costs associated with loss of productivity due to discomfort as well

as utilities from each of the simulations were annualized using cooling degree day data for College Station, TX. The results are shown in Table 7.3. The first row of Table 7.3 shows the costs associated with the current control methodology and technician-defined zone temperature setpoints. The second row indicates that for an increase in utility cost of \$1,085.16 per year, the combined economic cost of discomfort and utilities in the UBO could be reduced by \$5,307.85, or 38.1%. Essentially by just altering the zone temperature setpoints from 23 °C to approximately 22 °C, significant economic savings in productivity can be had from decreases in discomfort. For comparing the steady-state and MPC control methods, the current control method with the PMV optimal setpoints will be used as the baseline. The steady-state control method gives an 8.7% reduction in utility costs and a 10.24% reduction in overall costs from the baseline. The MPC control method gives an 8.4% reduction in utility costs and a 13.3% reduction in overall costs. Although the MPC method has a slightly higher utility cost than the steady-state, it is able to provide better comfort control. This is partly due to the pre-cooling as a result of the prediction horizon, as well as determining optimal trajectories of setpoints versus the steady-state predictions.

Table 7.3: Annualized economic costs for constant occupation profiles.

	Zone			Total	
	Comfort Cost	CHW Cost	Fan Cost	Utility Cost	Total Cost
Current Control w/ User Setpoints	\$6,869.38	\$4,093.61	\$1,255.67	\$7,046.03	\$13,915.41
Current Ctrl. w/ PMV Setpoints	\$476.37	\$4,562.48	\$1,677.62	\$8,131.19	\$8,607.56
Steady-State w/ PMV Setpoints	\$299.21	\$4,383.28	\$1,226.63	\$7,426.72	\$7,725.93
MPC w/ PMV Setpoints	\$15.05	\$4,310.81	\$1,351.90	\$7,449.47	\$7,464.52

7.5.2 Random Occupancy Simulation

Newer building energy management systems will include occupancy sensors in rooms that allow for setbacks to occur when the room is unoccupied. The next simulation that was completed included a random occupancy profile of the zones in the UBO to simulate having occupancy sensors. The setback temperatures used for the zones were the same as those currently used in the UBO (29.4 °C). The zone temperatures from the steady-state and MPC simulations are shown in Figures 7.17 and 7.18. Again, the two control methods show similar results, with the MPC controller implementing pre-cooling. For the MPC controller, perfect knowledge of the future outdoor air temperature, outdoor air relative humidity, and occupancy of the zones are assumed. In application, the outdoor air temperature and relative humidity could be estimated from weather forecasts and the future zone occupancy could be estimated from employee scheduling.

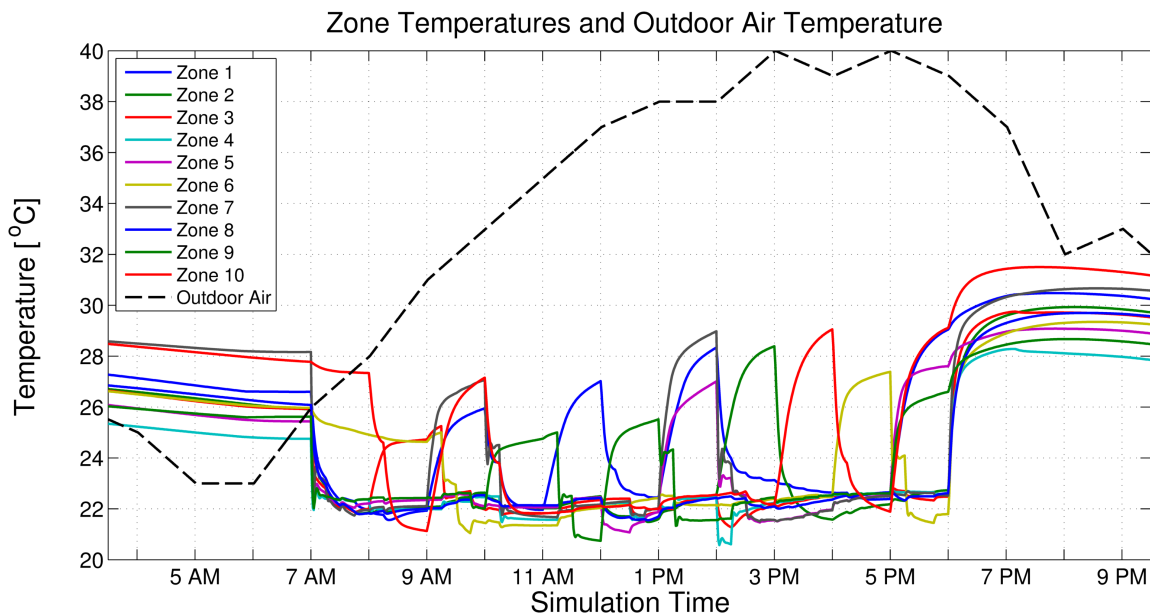


Figure 7.17: Zone temperatures for the steady-state method with random occupancy.

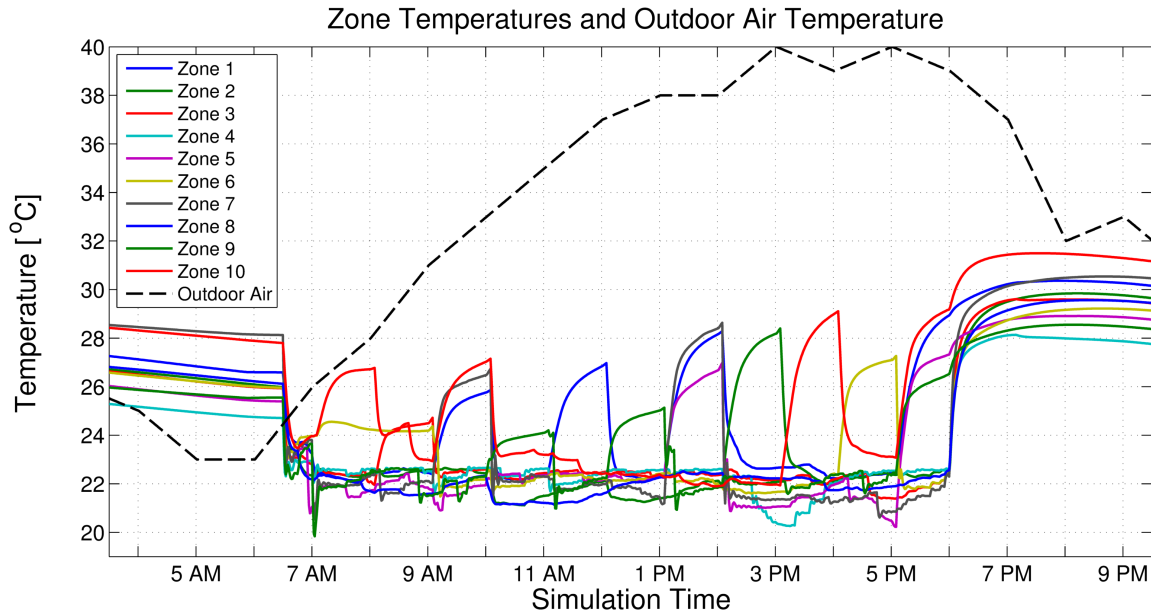


Figure 7.18: Zone temperatures for the MPC controller with random occupancy.

The economic cost results for the random occupancy simulations with the currently used setback temperatures are listed in Table 7.4. The steady-state method did slightly worse than the current control in overall zone comfort cost but managed to reduce both the total utility cost and the combined comfort and utility costs by 7.2% and 4.4%, respec-

Table 7.4: Annualized economic costs for random occupation profiles and standard setback temperatures.

	Zone		Total		
	Comfort Cost	CHW Cost	Fan Cost	Utility Cost	Total Cost
Current Ctrl. w/ PMV Setpoints	\$1,598.30	\$4,051.93	\$1,699.75	\$7,431.14	\$9,029.44
Steady-State w/ PMV Setpoints	\$1,729.78	\$4,045.36	\$1,177.67	\$6,899.78	\$8,629.56
MPC w/ PMV Setpoints	\$1,354.39	\$4,056.91	\$1,296.88	\$7,035.31	\$8,389.71

tively. These savings were mainly due to decreased electricity use by the fan. The MPC controller was able to improve on the overall cost with a reduction of 7.1% while giving 5.3% savings in utilities used from improved comfort and decreased fan use.

7.5.3 Random Occupancy with Optimal Setbacks Simulation

Simulations were completed in which the steady-state and MPC methods were allowed to determine the optimal setback temperatures as opposed to using the setback temperatures currently in place in the UBO. The zone temperatures from both control methods are shown in Figures 7.19 and 7.20. Compared to the zone temperature profiles with the standard setbacks, the use of optimal setbacks resulted in zone temperature deviating less from the optimal comfort zone between 0 and -0.5 PMV for zero loss of productivity.

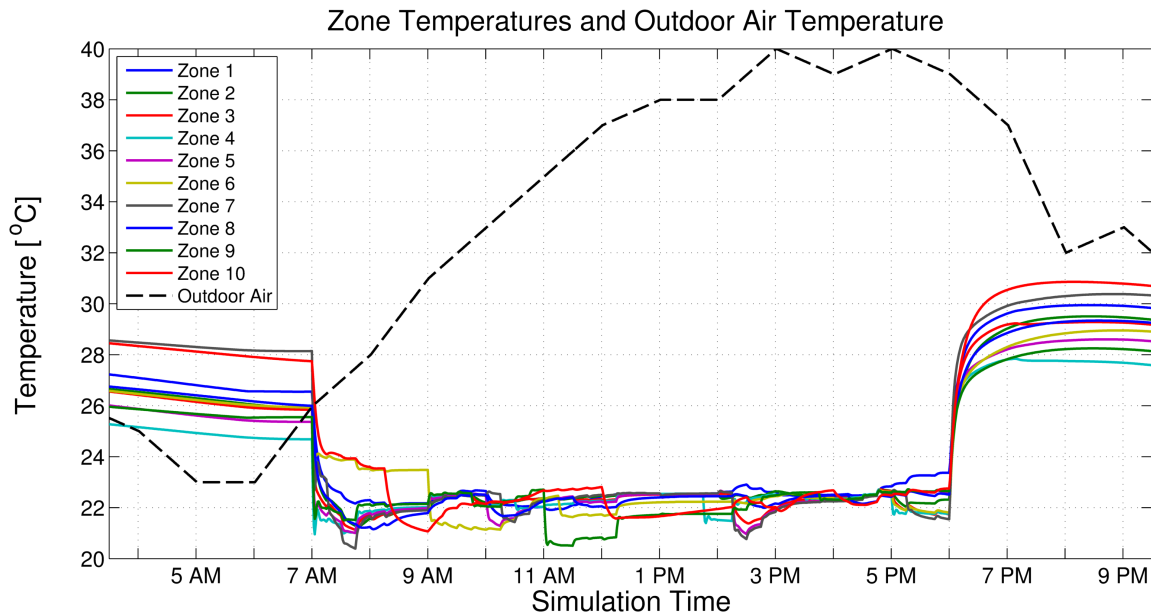


Figure 7.19: Zone temperatures for the steady-state method with random occupancy and optimal zone setbacks.

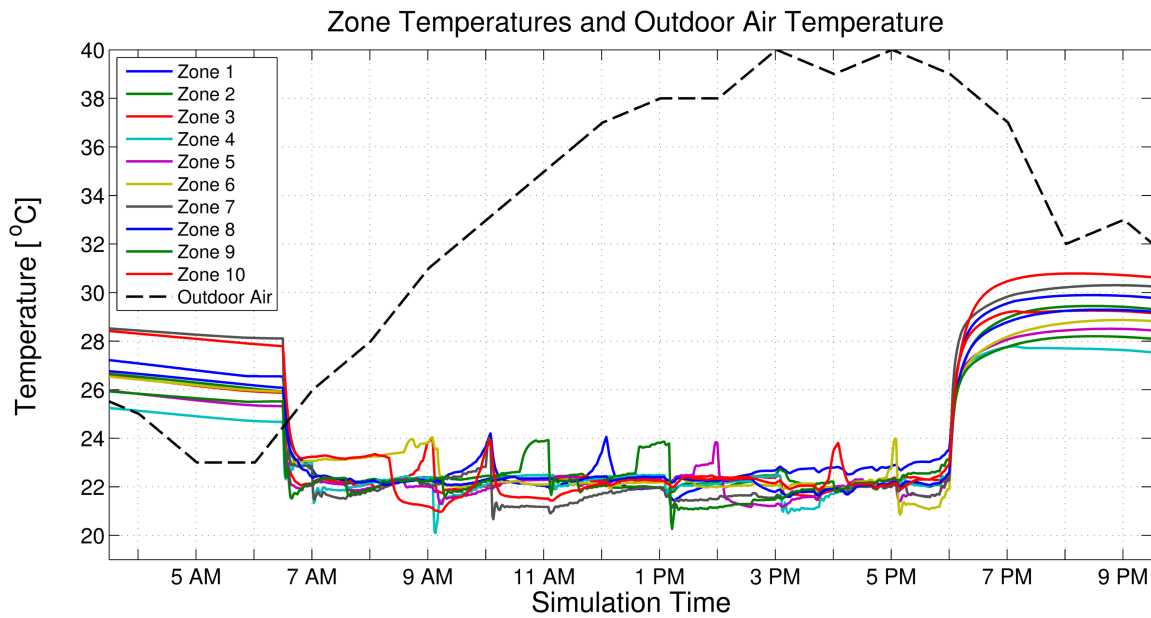


Figure 7.20: Zone temperatures for the MPC controller with random occupancy and optimal zone setbacks.

The economic results of the simulations with random occupancy and optimal setback temperatures are detailed in Table 7.5. Compared to the baseline of the current control using standard setbacks, both the steady-state and MPC methods provide performance improvements. While the decrease in annual utility costs are only \$187.49 (2.5%) and \$135.23 (1.8%) for the steady-state and MPC methods, respectively, the methods are much more efficient with the resources that they use. The steady-state was able to reduce the cost of loss productivity by \$1,297.50 for a combined decrease in discomfort and utility costs of 16.5%. The MPC method reduced the cost of loss productivity by \$1,416.32 to give a combined decrease in total costs of 17.2%. Both algorithms are able to provide significant productivity savings while decreasing the overall utility cost by using slightly more chilled water and less fan power than the current control method. Additionally, the added capability of determining optimal setback temperatures eliminates the need for building technicians to determining and adjusting setbacks, which can be a significant time

investment as the number of zones and equipment grows.

Table 7.5: Annualized economic costs for random occupation profiles and optimal setback temperatures.

	Zone			Total	
	Comfort Cost	CHW Cost	Fan Cost	Utility Cost	Total Cost
Steady-State w/ PMV Setpoints	\$300.79	\$4,320.13	\$1,132.90	\$7,243.65	\$7,544.45
MPC w/ PMV Setpoints	\$181.97	\$4,276.29	\$1,247.17	\$7,295.91	\$7,477.89

7.5.4 Priority Zone Case

Sometimes it may be desirable to prioritize some zones over others with regard to maintaining comfort. It may be that certain zones are not occupied for extended periods of time, so slightly more uncomfortable conditions may be permissible to reduce utility costs. It could be that one zone is the office or work space of a very important person (VIP) that the building operators wish to ensure stay comfortable. A zone may require specific comfort conditions to be maintained for a certain process or piece of equipment. In any case, the ability to prioritize certain zones over others, either permanently or temporarily, is a desirable feature in building controls. Both the steady-state and MPC methods are capable of doing this through adjusting the annual salary assigned to each zone. This value can essentially act as weight to allow the building operator to prioritize certain zones over others. The results from a simulation where the MPC method is used to prioritize Zone 5's comfort over others are shown in Figure 7.21. Several zone temperatures can be seen rising away from their 0 PMV temperature threshold beginning around 12 PM. As the cooling demand of these zones increases with the outdoor air temperature and solar loads,

their temperatures continue to increase; however, Zone 5 is maintained in the zero loss of productivity range of -0.5 to 0 PMV, as shown in Figure 7.22. The discomfort costs between the two methods were essentially equal, but the MPC method was able to reduce annual utility costs by 8.2% from the utility costs of the steady-state method. This result stemmed from reduced usage of both chilled water and fan power.

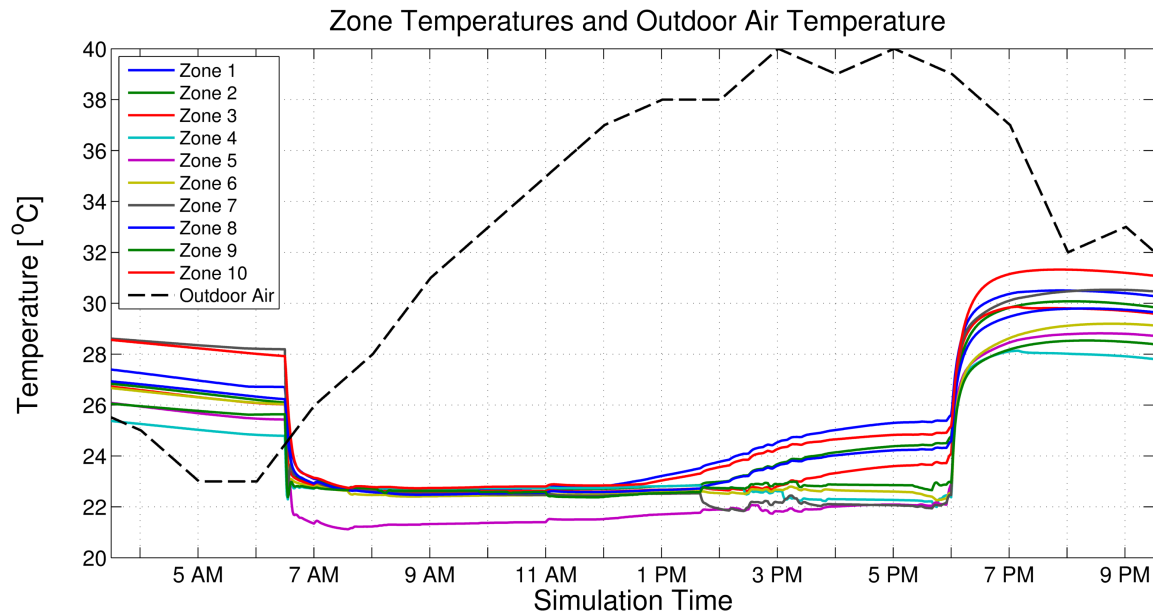


Figure 7.21: Zone temperatures for the MPC method with Zone 5 prioritized over other zones.

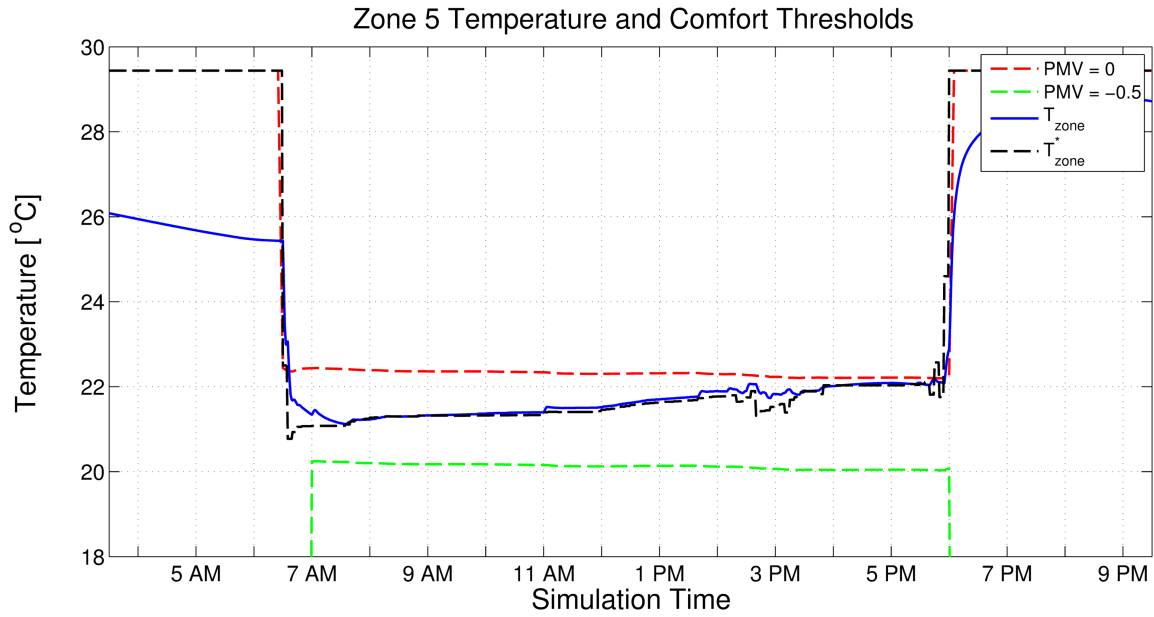


Figure 7.22: Zone 5 temperature with comfort limits as a prioritized zone.

7.6 Overall Comparison

Table 7.6 presents the overall comparison between the steady-state and MPC methods with the current control method using LOP optimal temperature setpoints ($PMV = -0.25$). Based on the results in the table, both the steady-state and MPC methods outperformed the current control methods with regard to total cost savings. The new methods showed the best performance increase (16.5% and 17.2%) when random occupancy profiles were present and the methods were able to determine the optimal setback temperatures. In the case of random occupancy profiles and standard setbacks, the steady-state algorithm had higher costs due to loss productivity than the current control method, but was able to use less chilled water and fan power to produce greater overall savings.

Comparing the steady-state performance to the MPC performance shows that, depending on the operational case, a full MPC implementation gives you marginal improvement. Specifically, in the case of constant occupancy the steady-state method was able to achieve

Table 7.6: Summary of results of the simulations.

Method	Setbacks	Occupancy	Comfort Savings	Utility Savings	Total Savings
Steady-State	N/A	Constant	37.19%	8.66%	10.24%
MPC	N/A	Constant	96.84%	8.38%	13.28%
Steady-State	Current	Random	-8.23%	7.15%	4.43%
MPC	Current	Random	15.26%	5.33%	7.08%
Steady-State	Optimal	Random	81.18%	2.52%	16.45%
MPC	Optimal	Random	88.61%	1.82%	17.18%

approximately 77.1% of the performance of MPC method. With random occupancy and optimal setback temperatures, the difference between the two methods was even smaller with the steady-state achieving 95.8% of the performance of the MPC. This comparison is insightful given that a fully dynamic MPC solution becomes increasingly more complex to implement as systems scale, along with an increase in computational burden and time. Essentially, in control of building energy systems, a steady-state prediction method can provide nearly as well as a full MPC solution with less complexity and computational requirements.

7.7 Conclusions

In this paper, an economic Model Predictive Control (MPC) strategy for control of building energy systems is proposed. The MPC strategy aims to reduce the total energy consumption of the HVAC systems while concurrently minimizing occupant loss of productivity (LOP) due to discomfort. To achieve this, energy consumption and LOP are put on equal terms through the development of economic objective functions. The economic cost associated with the production of chilled water for cooling air is determined along with the economic cost of the power consumed by the fan in the air handling unit (AHU). An occupant's LOP due to discomfort is monetized by leveraging a relationship between

the Predicted Mean Vote (PMV) and LOP to determine a percentage of an occupant's salary that is lost.

The MPC strategy is implemented in a centralized fashion to minimize the combined economic cost of utilities and LOP. The performance of the MPC strategy is compared with a previously developed steady-state optimal control method and a baseline control. The baseline control is taken from the Utilities Business Office (UBO) at Texas A&M University: a small, single floor office building with ten controllable zones and a central roof-top AHU. An EnergyPlus model of the UBO is developed and used in co-simulation with MATLAB to simulate the three control strategies. Several simulations are completed with varying parameters such as constant vs. random occupancy and standard vs. optimal zone setback temperatures. Additionally, the MPC strategy's capability of prioritizing certain zone's comfort over others is demonstrated.

For the UBO it is found that by simply changing the current zone temperature set-points with the currently implemented control that significant productivity savings can be had. The steady-state and MPC strategies outperform the current control method in terms of overall economic cost and utility cost. The MPC strategy outperforms the currently control method in occupant comfort measures in all cases while the steady-state method outperforms the current control in all cases except for when there is random occupancy and the current setback temperatures are used. Additionally, comparison between the steady-state optimal control method and the MPC strategy show that steady-state predictions can, in some cases, provide nearly as good performance as a fully dynamic solution. This insight is significant as implementing a dynamic MPC solution on building energy systems becomes increasingly complex as systems scale and requires greater computational time; thus, steady-state predictions can be used to achieve near optimal control.

Future efforts entail expanding the simulation to multiple buildings of varying size. The UBO is a small office space with relatively low thermal mass; therefore, it would be

worthwhile to investigate the proposed control method's performance on larger buildings with different thermal properties. Also, developing economic objective functions for other building energy systems including heated hot water services, central pumps, and chillers would enable simulations of entire campuses. Lastly, verification of the performance of the proposed control strategy can be completed by applying the advanced control method directly to the UBO.

8. CONCLUSIONS

In this study, the proposed solutions have the ultimate goal of reducing energy consumption, generating economic savings, and maximizing occupant comfort in existing and future buildings. This goal has been pursued through two separate efforts: the creation of Autonomous Robotic Assessments of Energy, and the development of advanced economic optimal controllers for building energy systems that account for user comfort and productivity. While seemingly disparate topics, the two areas can benefit on another in future integrations as the technologies become more refined.

Specifically, an automated energy audit process was presented, termed Autonomous Robotic Assessments of Energy (AuRAE). As part of this process, audit-centric navigational algorithms were developed for both indoor and outdoor building environments. The indoor algorithm is capable of full exploration of a previously unknown space while routing to targets of interest as they become identified in real time. The exterior algorithm is capable navigating the perimeter of a building, including traversing possible gaps due to open bay doors in commercial facilities. These algorithms were simulated and proven to perform successfully in their respective tasks.

Robotic audit capabilities were demonstrated, showing proof of concept and feasibility. A automated lighting identification and analysis package prototype was developed. The package was installed on a ground robot platform and demonstrated to continually create a 2-dimensional map of its surrounding environment while identifying the 3-dimensional locations of lights and recording lighting specific data. Additionally, an environmental baseline measurement package was installed on an aerial vehicle platform and demonstrated to record environmental data while producing a 3D map of the interior of the building and tracking its position. These efforts and results contribute to the automation of energy

audits and the building audit community.

Advanced economic optimal controllers for building energy systems were developed. These methods utilized component level economic objective functions that enables the cost analysis of potential savings with regard to specific building energy equipment. They also accounted for the cost of occupant discomfort in the occupants' loss of productivity, suggesting that it may be more cost effective for businesses to focus on keeping their occupants comfortable over obtaining increasingly small improvements in energy efficiency. Lastly, through simulation, it was determined that employing dynamic models over steady-state predictions in MPC-like controllers provided a marginal increase in performance of approximately 10-20% depending on the situation, suggesting that in some cases it may not be worth the additional difficulty to implement dynamic predictions. These results contribute to the building controls community and the growing interest in the importance of occupant comfort as well as building that adapt in real-time to environmental, economic, and occupant conditions.

8.1 Future Directions

There are several directions for future research based on the results of the presented study. Regarding the Autonomous Robotic Assessments of Energy, implementing the autonomous algorithms on experimental platforms would be extremely useful in further validation of the navigational strategies. With the algorithms implemented, integration of the audit package prototypes would be the logical next step. Additional audit packages can be developed as well. Some possibilities are insulation fault detection, occupant enumeration, and HVAC measurement and diagnosis.

Cooperation between auditing platforms would be a significant addition in terms of performance and capabilities. Investigation into cooperative mapping strategies, optimal exploration, and specialist type platforms could be performed. With a fleet of autonomous

auditors, it would be cost effective to have one robot that performs lighting measurements, another that does thermal imaging, and others with specific capabilities as fewer total sensors would be needed. Ground home-base platforms could be developed to offload computational burdens from aerial platforms and extend their flight time by offering exchangeable power sources.

The developed navigational algorithms use in other applications could be studied. For example, companies are currently working to develop automated solutions for underground mine exploration and mapping. The autonomous exploration strategy could also be employed in search and rescue operations, specifically in cases where human intervention is dangerous or not possible.

With respect to the economic objective functions, implementation and validation on the UBO building would be a significant achievement. Also, objective functions for additional equipment such as water pumps, central plants, and other building energy systems can be developed. Simulations of larger, more complex systems would provide additional opportunity for identifying cost savings potential as well as determining the performance of the proposed control strategy. Implementing the economic objective functions with the distributed MPC algorithm under current development in the lab would be a welcome advancement as well. With more simulation, patterns of optimal control can potentially be identified, leading to the development of heuristic models and strategies for quicker adoption and simplified implementation in industry.

Further investigation into the impact of an occupant's comfort and their environmental conditions on their productivity is needed. This is a largely unexplored area, and as buildings continue to become more automated and technology becomes smarter, understanding of this complex issue will become more critical. Also, the psychology behind allowing forms of user control to influence or override the system and the resulting effects on occupant comfort would be incredibly useful. Another topic regarding occupants is

providing feedback about the operating conditions of the system. For example, is it beneficial to provide the occupant with information about if they were to adjust their desired temperature setting to become a little more uncomfortable, they would receive a certain amount of savings? Or what are the effects of only allowing users to request a room to be warmer or colder, and not including exact numbers in the interaction? There are numerous opportunities for increasing understanding of how to best include users in the design and control of building energy systems.

REFERENCES

- [1] U.S. Energy Information Administration (EIA), “U.S. energy production, consumption has changed significantly since 1908 - Today in Energy,” Nov. 2016. [Online]. Available: <https://www.eia.gov/todayinenergy/detail.php?id=28592>
- [2] U.S. Department of Energy, “Building Share of U.S. Primary Energy Consumption (Percent): <http://buildingsdatabook.eren.doe.gov>,” Mar. 2012. [Online]. Available: <http://buildingsdatabook.eren.doe.gov/TableView.aspx?table=1.1.3-A>
- [3] C. Bay, T. Terrill, and B. Rasmussen, “Autonomous lighting assessments in buildings: part 1 - robotic navigation and mapping,” *Advances in Building Energy Research*, 2016.
- [4] P. Fanger, *Thermal comfort: analysis and applications in environmental engineering*. New York: McGraw-Hill, 1972.
- [5] C. Bay, T. Terrill, and B. Rasmussen, “Autonomous Lighting Audits: Part 1 – Building Navigation and Mapping,” in *ASME 2014 Dynamic Systems and Control Conference*, vol. 1. San Antonio, TX, USA: ASME, Oct. 2014.
- [6] S. S. Ge, X. Lai, and A. Mamun, “Boundary following and globally convergent path planning using instant goals,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 2, pp. 240–254, Apr. 2005.
- [7] ASHRAE, *Procedures For Commercial Building Energy Audits, Second Edition*, 2nd ed. Atlanta, Ga: ASHRAE, 2011.
- [8] C. Bay and B. Rasmussen, “Simulation and validation of interior and exterior navigational strategies for autonomous robotic assessments of energy,” in *American Control Conference (ACC), 2015*, July 2015, pp. 270–275.

- [9] T. Terrill, C. Bay, and B. Rasmussen, “Autonomous lighting assessments in buildings: part 2 – light identification and energy analysis,” *Advances in Building Energy Research*, 2016.
- [10] P. Roelofsen, “The impact of office environments on employee performance: The design of the workplace as a strategy for productivity enhancement,” *ResearchGate*, vol. 1, no. 3, pp. 247–264, July 2002. [Online]. Available: https://www.researchgate.net/publication/242347378/The_impact_of_office_environments_on_employee_performance_The_design_of_the_workplace_as_a_strategy_for_productivity_enhancement
- [11] S. Mullainathan and E. Shafir, *Scarcity: Why Having Too Little Means So Much*. New York: Times Books, Sept. 2013.
- [12] U.S. Energy Information Administration (EIA), “Total Energy - Data,” 2016. [Online]. Available: <http://www.eia.gov/totalenergy/data/browser/#/?f=A&start=1949&end=2015&charted=4-6-7-14>
- [13] European Commission, “Renewable energy - Moving towards a low carbon economy,” 2009. [Online]. Available: [/energy/en/topics/renewable-energy](http://energy.en/topics/renewable-energy)
- [14] Department of Energy, “20% Wind Energy by 2030: Increasing Wind Energy’s Contribution to U.S. Electricity Supply,” 2008. [Online]. Available: <http://energy.gov/eere/wind/20-wind-energy-2030-increasing-wind-energys-contribution-us-electricity-supply>
- [15] U.S. Energy Information Administration, “Annual Energy Review 2011,” U.S. Department of Energy, Tech. Rep., Sept. 2012. [Online]. Available: <http://www.eia.gov/totalenergy/data/annual/>

- [16] —, “Emissions of Greenhouse Gases in the United States 2008,” U.S. Department of Energy, Tech. Rep., 2009. [Online]. Available: <http://www.eia.gov/oiaf/1605/archive/gg09rpt/index.html>
- [17] Existing Commercial Buildings Working Group, “Inventory of U.S. Greenhouse Gas Emissions and Sinks: 1990-2009,” U.S. Environmental Protection Agency, Tech. Rep., Apr. 2012. [Online]. Available: <http://www.epa.gov/climatechange/ghgemissions/usinventoryreport.html>
- [18] Department of Energy, “Industrial Assessment Centers (IACs) Database,” 2015. [Online]. Available: <http://energy.gov/eere/amo/industrial-assessment-centers-iacs>
- [19] NEMI, “Building commissioning, testing, adjusting, and balancing,” National Energy Management Institute, Tech. Rep., 2005.
- [20] PECE, “Final report—California commissioning market characterization study,” Pacific Gas and Electric Company, Tech. Rep., 2000.
- [21] Association of Energy Engineers, “REA - Certified Residential Energy Auditor,” 2017. [Online]. Available: <http://www.aeecenter.org/i4a/pages/index.cfm?pageid=3477>
- [22] G. Granosik, M. G. Hansen, and J. Borenstein, “The OmniTread serpentine robot for industrial inspection and surveillance,” *Industrial Robot: An International Journal*, vol. 32, no. 2, pp. 139–148, Apr. 2005. [Online]. Available: <http://www.emeraldinsight.com/doi/abs/10.1108/01439910510582264>
- [23] C. Balaguer, A. Gimenez, J. M. Pastor, V. M. Padron, and M. Abderrahim, “A Climbing Autonomous Robot for Inspection Applications in 3d Complex Environments,” *Robotica*, vol. 18, no. 3, pp. 287–297, May 2000. [Online]. Available: <http://dx.doi.org/10.1017/S0263574799002258>

- [24] D. Borrmann, A. Nuchter, M. Dakulovic, I. Maurovic, I. Petrovic, D. Osmankovic, and J. Velagic, "The Project ThermalMapper - Thermal 3d Mapping of Indoor Environments for Saving Energy," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 31–38, Jan. 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474667016335844>
- [25] O. Oreifej, J. Cramer, and A. Zakhor, "Automatic Generation of 3d Thermal Maps of Building Interiors," in *ASHRAE Transactions*, 2014. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.645.4516&rep=rep1&type=pdf>
- [26] A. Carrio, J. Pestana, J.-L. Sanchez-Lopez, R. Suarez-Fernandez, P. Campoy, R. Tendero, M. Garcia-De-Viedma, B. Gonzalez-Rodrigo, J. Bonatti, J. G. Rejas-Ayuga, R. Martinez-Marin, and M. Marchamalo-Sacristan, "UBRISTES: UAV-Based Building Rehabilitation with Visible and Thermal Infrared Remote Sensing," in *Robot 2015: Second Iberian Robotics Conference*, ser. Advances in Intelligent Systems and Computing, L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano, and V. Munoz-Martinez, Eds. Springer International Publishing, 2016, no. 417, pp. 245–256, dOI: 10.1007/978-3-319-27146-0_19. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-27146-0_19
- [27] M. L. Mauriello, L. Norooz, and J. E. Froehlich, "Understanding the Role of Thermography in Energy Auditing: Current Practices and the Potential for Automated Solutions," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 1993–2002. [Online]. Available: <http://doi.acm.org/10.1145/2702123.2702528>
- [28] C. Wang, Y. K. Cho, and M. Gai, "As-Is 3d Thermal Modeling for Existing

- Building Envelopes Using a Hybrid LIDAR System,” *Journal of Computing in Civil Engineering*, vol. 27, no. 6, pp. 645–656, 2013. [Online]. Available: [http://dx.doi.org/10.1061/\(ASCE\)CP.1943-5487.0000273](http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000273)
- [29] D. Roca, J. Armesto, S. Lagueta, and L. Diaz-Vilarino, “Lidar-equipped uav for building information modelling,” in *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL. Gottingen, Germany: Copernicus GmbH, 2014, pp. 523–527. [Online]. Available: <http://search.proquest.com/docview/1757013568/abstract/AA1F76D501A040D6PQ/1>
- [30] Y. Ham and M. Golparvar-Fard, “An automated vision-based method for rapid 3d energy performance modeling of existing buildings using thermal and digital imagery,” *Advanced Engineering Informatics*, vol. 27, no. 3, pp. 395–409, Aug. 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S147403461300027X>
- [31] H. Son, S. Lee, and C. Kim, “Automated 3d Model Reconstruction to Support Energy-efficiency,” *Procedia Engineering*, vol. 145, pp. 571–578, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877705816300510>
- [32] K. Kulakowski, P. Matyasik, and S. Ernst, “Modeling indoor lighting inspection robot behavior using Concurrent Communicating Lists,” *Expert Systems with Applications*, vol. 41, no. 4, Part 1, pp. 984–989, Mar. 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417413004600>
- [33] C. R. Price and B. P. Rasmussen, “Effective Tuning of Cascaded Control Loops for Nonlinear HVAC Systems,” p. V002T29A002, Oct. 2015. [Online]. Available: <http://dx.doi.org/10.1115/DSCC2015-9806>
- [34] D. Gyalistras and The OptiControl Team, “Final Report: Use of Weather and Occupancy Forecasts for Optimal Building Climate Control (OptiControl),” Terrestrial

Systems Ecology ETH, Zurich, Switzerland, Tech. Rep., 2010.

- [35] C. Ghiaus and I. Hazyuk, “Calculation of optimal thermal load of intermittently heated buildings,” *Energy and Buildings*, vol. 42, no. 8, pp. 1248–1258, Aug. 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378778810000484>
- [36] Y. Ma, F. Borrelli, B. Hancey, A. Packard, and S. Bortoff, “Model Predictive Control of thermal energy storage in building cooling systems,” in *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009*, Dec. 2009, pp. 392–397.
- [37] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, July 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0967066102001867>
- [38] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control Theory and Design*. Madison, WI: Nob Hill Pub, Aug. 2009.
- [39] I. Hazyuk, C. Ghiaus, and D. Penhouet, “Optimal temperature control of intermittently heated buildings using Model Predictive Control: Part II – Control algorithm,” *Building and Environment*, vol. 51, pp. 388–394, May 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0360132311003921>
- [40] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari, “Use of model predictive control and weather forecasts for energy efficient building climate control,” *Energy and Buildings*, vol. 45, pp. 15–27, Feb. 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378778811004105>

- [41] R. Halvgaard, N. K. Poulsen, H. Madsen, and J. B. Jorgensen, "Economic Model Predictive Control for building climate control in a Smart Grid," in *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, Jan. 2012, pp. 1–6.
- [42] P.-D. Morosan, R. Bourdais, D. Dumur, and J. Buisson, "Building temperature regulation using a distributed model predictive control," *Energy and Buildings*, vol. 42, no. 9, pp. 1445–1452, Sept. 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378778810000915>
- [43] C. R. Touretzky and M. Baldea, "Integrating scheduling and control for economic MPC of buildings with energy storage," *Journal of Process Control*, vol. 24, no. 8, pp. 1292–1300, Aug. 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0959152414001164>
- [44] F. Oldewurtel, A. Parisio, C. N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth, "Energy efficient building climate control using Stochastic Model Predictive Control and weather predictions," in *Proceedings of the 2010 American Control Conference*, June 2010, pp. 5100–5105.
- [45] C. D. Corbin, G. P. Henze, and P. May-Ostendorp, "A model predictive control optimization environment for real-time commercial building application," *Journal of Building Performance Simulation*, vol. 6, no. 3, pp. 159–174, May 2013. [Online]. Available: <http://dx.doi.org/10.1080/19401493.2011.648343>
- [46] F. Oldewurtel, A. Ulbig, A. Parisio, G. Andersson, and M. Morari, "Reducing peak electricity demand in building climate control using real-time pricing and model predictive control," in *49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 1927–1932.
- [47] D. Molina, C. Lu, V. Sherman, and R. G. Harley, "Model Predictive and Genetic Algorithm-Based Optimization of Residential Temperature Control in the Presence

- of Time-Varying Electricity Prices,” *IEEE Transactions on Industry Applications*, vol. 49, no. 3, pp. 1137–1145, May 2013.
- [48] J. D. Alvarez, J. L. Redondo, E. Camponogara, J. Normey-Rico, M. Berenguel, and P. M. Ortigosa, “Optimizing building comfort temperature regulation via model predictive control,” *Energy and Buildings*, vol. 57, pp. 361–372, Feb. 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378778812005610>
- [49] Y. Ma, F. Borrelli, B. Hancey, B. Coffey, S. Bengea, and P. Haves, “Model Predictive Control for the Operation of Building Cooling Systems,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 796–803, May 2012.
- [50] S. C. Bengea, A. D. Kelman, F. Borrelli, R. Taylor, and S. Narayanan, “Implementation of model predictive control for an HVAC system in a mid-size commercial building,” *HVAC&R Research*, vol. 20, no. 1, pp. 121–135, Jan. 2014. [Online]. Available: <http://dx.doi.org/10.1080/10789669.2013.834781>
- [51] S. R. West, J. K. Ward, and J. Wall, “Trial results from a model predictive control and optimisation system for commercial building HVAC,” *Energy and Buildings*, vol. 72, pp. 271–279, Apr. 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378778813008542>
- [52] H. Karlsson and C.-E. Hagentoft, “Application of model based predictive control for water-based floor heating in low energy residential buildings,” *Building and Environment*, vol. 46, no. 3, pp. 556–569, Mar. 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0360132310002672>
- [53] G. S. Duburcq, “Advanced control for intermittent heating,” in *Proceedings of Clima 2000 Conference*, Brussels, Belgium, Sept. 1997. [Online]. Available: <http://www.aivc.org/resource/advanced-control-intermittent-heating>

- [54] S. Yuan and R. Perez, "Multiple-zone ventilation and temperature control of a single-duct VAV system using model predictive strategy," *Energy and Buildings*, vol. 38, no. 10, pp. 1248–1261, Oct. 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S03787778806000764>
- [55] M. Avci, M. Erkoç, A. Rahmani, and S. Asfour, "Model predictive HVAC load control in buildings using real-time electricity pricing," *Energy and Buildings*, vol. 60, pp. 199–209, Feb. 2013. [Online]. Available: <https://miami.pure.elsevier.com/en/publications/model-predictive-hvac-load-control-in-buildings-using-real-time-e>
- [56] J. Towler, "BSRIA Diamond Group Forum – A glimpse of the future?" Mar. 2015. [Online]. Available: <https://blogs.bsria.co.uk/2015/03/09/bsria-diamond-group-forum-a-glimpse-of-the-future/>
- [57] D. Schleicher, L. Bergasa, M. Ocana, R. Barea, and M. Lopez, "Real-Time Hierarchical Outdoor SLAM Based on Stereovision and GPS Fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 440–452, Sept. 2009.
- [58] S. Thrun, M. Diel, and D. Hähnel, "Scan alignment and 3d surface modeling with a helicopter platform," in *In Proceedings of the International Conference on Field and Service Robotics*, 2003.
- [59] S. Ahrens, D. Levine, G. Andrews, and J. How, "Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments," in *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, May 2009, pp. 2643–2648.
- [60] K. Celik, S.-J. Chung, and A. K. Somani, "MVCSLAM: Mono-Vision Corner SLAM for Autonomous Micro-Helicopters in GPS Denied Environments," in *AIAA Guidance, Navigation and Control Conference and Exhibit*. Honolulu, Hawaii: American Institute of Aeronautics and Astronautics, Aug. 2008.

- [61] S. Grzonka, G. Grisetti, and W. Burgard, "A Fully Autonomous Indoor Quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, Feb. 2012.
- [62] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007.
- [63] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3d motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Nov. 2011, pp. 155–160.
- [64] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *AAAI/IAAI*, 2002, pp. 593–598.
- [65] D. Nister, "Preemptive RANSAC for live structure and motion estimation," in *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, Oct. 2003, pp. 199–206 vol.1.
- [66] H. Strasdat, J. Montiel, and A. J. Davison, "Scale Drift-Aware Large Scale Monocular SLAM," in *Robotics: Science and Systems*, vol. 2, 2010, p. 5.
- [67] J. Stuhmer, S. Gumhold, and D. Cremers, "Real-time Dense Geometry from a Handheld Camera," in *Proceedings of the 32Nd DAGM Conference on Pattern Recognition*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 11–20. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1926258.1926261>
- [68] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, Nov. 2011. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/rob.20412/abstract>

- [69] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey, “Outdoor mapping and navigation using stereo vision,” in *Experimental Robotics*. Springer, 2008, pp. 179–190.
- [70] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the RGB-D SLAM system,” in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 1691–1696.
- [71] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, “Real-time 3d visual SLAM with a hand-held RGB-D camera,” in *Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, Vasteras, Sweden, 2011.
- [72] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D Mapping: Using Depth Cameras for Dense 3d Modeling of Indoor Environments,” in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and G. Sukhatme, Eds. Springer Berlin Heidelberg, Jan. 2014, no. 79, pp. 477–491. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-28572-1_33
- [73] S. M. Lavalle, “Rapidly-Exploring Random Trees: A New Tool for Path Planning,” Iowa State University, Tech. Rep., 1998.
- [74] J. Kuffner and S. LaValle, “RRT-connect: An efficient approach to single-query path planning,” in *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00*, vol. 2, 2000, pp. 995–1001 vol.2.
- [75] G. Oriolo, M. Vendittelli, L. Freda, and G. Troso, “The SRT method: randomized strategies for exploration,” in *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, vol. 5, Apr. 2004, pp. 4688–4694 Vol.5.

- [76] J. Espinoza, A. Sanchez, and M. Osorio, "Exploring unknown environments with mobile robots using SRT-Radial," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*, Oct. 2007, pp. 2089–2094.
- [77] C.-F. Juang and C.-H. Hsu, "Reinforcement Ant Optimized Fuzzy Controller for Mobile-Robot Wall-Following Control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3931–3940, Oct. 2009.
- [78] A. Bemporad, M. Di Marco, and A. Tesi, "Wall-following controllers for sonar-based mobile robots," in , *Proceedings of the 36th IEEE Conference on Decision and Control, 1997*, vol. 3, Dec. 1997, pp. 3063–3068 vol.3.
- [79] N. J. Cowan, J. Lee, and R. J. Full, "Task-level control of rapid wall following in the American cockroach," *Journal of Experimental Biology*, vol. 209, no. 9, pp. 1617–1629, May 2006. [Online]. Available: <http://jeb.biologists.org/content/209/9/1617>
- [80] Y. Ando and S. Yuta, "Following a wall by an autonomous mobile robot with a sonar-ring," in , *1995 IEEE International Conference on Robotics and Automation, 1995. Proceedings*, vol. 3, May 1995, pp. 2599–2606 vol.3.
- [81] R. Carelli and E. Oliveira Freire, "Corridor navigation and wall-following stable control for sonar-based mobile robots," *Robotics and Autonomous Systems*, vol. 45, no. 3–4, pp. 235–247, Dec. 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889003001489>
- [82] T. Yata, L. Kleeman, and S. Yuta, "Wall following using angle information measured by a single ultrasonic transducer," in *1998 IEEE International Conference on Robotics and Automation, 1998. Proceedings*, vol. 2, May 1998, pp. 1590–1596 vol.2.

- [83] E. Hong, L. Conroy, and M. Scholand, "U.S. Lighting Market Characterization Volume II Energy Efficient Lighting Technology Options." United States. Office of the Assistant Secretary of Energy Efficiency and Renewable Energy ;, Washington, D.C. :, Tech. Rep., Sept. 2005.
- [84] L. Halonen, E. Tetri, and R. G. Bhusal, Eds., *Guidebook on energy efficient electric lighting for buildings*. Aalto, Finland: Aalto University, school of Science and Technology, 2010.
- [85] Advanced Manufacturing Office, "Industrial Assessment Center Database," 2014.
- [86] M. Labbe and F. Michaud, "Online global loop closure detection for large-scale multi-session graph-based SLAM," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept. 2014, pp. 2661–2666.
- [87] A. P. Gagge, A. P. Fobelets, and L. G. Berglund, "A Standard Predictive Index of Human Response to the Thermal Environment," *ASHRAE Trans.; (United States)*, vol. 92:2B, Jan. 1986. [Online]. Available: <https://www.osti.gov/scitech/biblio/6494216>
- [88] Trimble, "3d modeling for everyone," 2017. [Online]. Available: <https://www.sketchup.com/>
- [89] D. B. Crawley, C. O. Pedersen, L. K. Lawrie, and F. C. Winkelmann, "EnergyPlus: Energy Simulation Program," *ASHRAE Journal*, vol. 42, pp. 49–56, 2000.
- [90] W. Bernal, M. Behl, T. X. Nghiem, and R. Mangharam, "MLE+: A Tool for Integrated Design and Deployment of Energy Efficient Building Controls." *4th ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings, (BuildSys '12), Toronto, Canada.*, Nov. 2012. [Online]. Available: http://repository.upenn.edu/mlab/_papers/51/

- [91] M. Wetter, “Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed,” *Journal of Building Performance Simulation*, vol. 4, no. 3, pp. 185–203, Sept. 2011. [Online]. Available: <http://dx.doi.org/10.1080/19401493.2010.518631>
- [92] R. Chintala, C. J. Bay, and B. P. Rasmussen, “A Black-Box Dynamic Modeling Strategy for Implementing Optimal Predictive Control Algorithm for Building HVAC Systems,” *Buildings and Environment*, vol. In review, 2017.
- [93] R. Chintala and B. Rasmussen, “Automated Modeling of Building HVAC Systems,” *Buildings and Environment*, vol. In review, 2017.
- [94] U.S. Department of Energy, “Advanced Manufacturing Office: www1.eere.energy.gov,” Mar. 2014. [Online]. Available: http://www1.eere.energy.gov/manufacturing/tech_assistance/iacs.html
- [95] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, vol. 3, 2009.
- [96] T. Terrill, C. Bay, and B. Rasmussen, “Autonomous Lighting Audits: Part 2 - Light Identification and Analysis,” in *ASME 2014 Dynamic Systems and Control Conference*, vol. 1. San Antonio, TX, USA: ASME, Oct. 2014.
- [97] U. Ruckert, S. Joaquin, and W. Felix, “Advances in Autonomous Mini Robots,” in *Proceedings of the 6-th AMiRE Symposium*. Bielefeld University, Germany: Springer, Mar. 2012.
- [98] S. Shen, N. Michael, and V. Kumar, “Autonomous multi-floor indoor navigation with a computationally constrained MAV,” in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 20–25.

- [99] K. Palmer, M. Walls, H. Gordon, and T. Gerarden, "Assessing the energy-efficiency information gap: results from a survey of home energy auditors," *Energy Efficiency*, vol. 6, no. 2, pp. 271–292, May 2013. [Online]. Available: <http://link.springer.com/article/10.1007/s12053-012-9178-2>
- [100] M. C. Baechler, "A Guide to Energy Audits," 2011. [Online]. Available: <http://www.pnl.gov/publications/abstracts.asp?report=388600>
- [101] R. Stephen, E. Tennant, C. Freyman, J. Ozawa, J. Chase, and D. Querejazu, "Saving Energy, Building Skills - Industrial Assessment Centers Impact," SRI International," Technical Report, Mar. 2015. [Online]. Available: <https://iac.university/technicalDocs/Industrial%20Assessment%20Centers%20Impacts%20SRI%20International.pdf>
- [102] R. C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," *The International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, Dec. 1986. [Online]. Available: <http://ijr.sagepub.com/content/5/4/56>
- [103] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK ; New York: Cambridge University Press, Apr. 2004.
- [104] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
- [105] N. M. P. a. A. Division, "Euler Angles, Quaternions, and Transformation Matrices," NASA, Tech. Rep., July 1977. [Online]. Available: <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19770024290.pdf>
- [106] D. L. Dilaura, K. W. Houser, R. G. Mistrick, and G. R. Steffy, Eds., *The Lighting Handbook: Reference and Application*, 10th ed. New York City: Illuminating Engineering, July 2011.

- [107] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: <http://www.nature.com/nature/journal/v521/n7553/abs/nature14539.html>
- [108] J. Siroky, F. Oldewurtel, J. Cigler, and S. Privara, “Experimental analysis of model predictive control for an energy efficient building heating system,” *Applied Energy*, vol. 88, no. 9, pp. 3079–3087, Sept. 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0306261911001668>
- [109] S. Privara, J. Siroky, L. Ferkl, and J. Cigler, “Model predictive control of a building heating system: The first experience,” *Energy and Buildings*, vol. 43, no. 2–3, pp. 564–572, Feb. 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378778810003749>
- [110] C. J. Bay, R. Chintala, and B. P. Rasmussen, “Steady-State Optimal Control of Building Energy Systems Using an Economic and Occupant Comfort Focused Objective Function,” *Energies*, vol. To be submitted, 2017.

APPENDIX A

UBO COMPONENT MODELS

Below are various models that were identified from the Utilities Business Office (UBO) and used in the simulations.

A.1 UBO Zone Models

Case 1: Normal Damper Operation

Equation A.1 shows the form of the linear, discrete models that were identified for the zones within the UBO.

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_i u_i + K_i e_i(k) \\ y_i(k) &= C_i x_i(k) + D_i u_i(k) \end{aligned} \quad (\text{A.1})$$

Below are the inputs used and the identified subsystem matrices for each of the zones within the UBO. The model identification procedure developed in [92] was used.

Subsystem Model - Zone 1:

$$\begin{aligned} u_1 &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad Rh_{OA}]^T \\ A_1 &= [0.6163] \quad B_1 = [0.0133 \quad 0.3714 \quad 0.001 \quad 0.001 \quad 0.001] \\ C_1 &= [1] \quad K_1 = [1.0963] \end{aligned} \quad (\text{A.2})$$

Subsystem Model - Zone 2:

$$\begin{aligned} u_2 &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad T_{ZONE,6} \quad T_{ZONE,1}]^T \\ A_2 &= [0.004 \quad 0.321 \quad 0.001 \quad 0.9602] \\ B_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0133 & 0.3714 & 0.001 & 0.001 & 0.001 & 0.0081 \end{bmatrix} \\ C_2 &= [0 \quad 1] \quad K_2 = \begin{bmatrix} -0.5983 \\ 1.6106 \end{bmatrix} \end{aligned} \quad (\text{A.3})$$

Subsystem Model - Zone 3:

$$\begin{aligned}
u_3 &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad T_{ZONE,4}]^T \\
A_3 &= [0.4113] \\
B_3 &= [0.0088 \quad 0.5742 \quad 0.001 \quad 0.001 \quad 0.001] \\
C_3 &= [1] \quad K_3 = [0.7928]
\end{aligned} \tag{A.4}$$

Subsystem Model - Zone 4:

$$\begin{aligned}
u_4 &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad Rh_{OA} \quad T_{ZONE,3}]^T \\
A_4 &= [-0.1052] \\
B_4 &= [0.0049 \quad 0.5437 \quad 0.0005 \quad 0.0005 \quad 0.0005 \quad 0.0028] \\
C_4 &= [2] \quad K_4 = [-0.0526]
\end{aligned} \tag{A.5}$$

Subsystem Model - Zone 5:

$$\begin{aligned}
u_5 &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad Rh_{OA}]^T \\
A_5 &= [0 \quad -0.3521 \quad 0.5 \quad 0.4022] \\
B_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.0094 & 0.7643 & 0.001 & 0.0245 & 0.001 \end{bmatrix} \\
C_5 &= [0 \quad 1] \quad K_5 = \begin{bmatrix} -0.3521 \\ 0.4022 \end{bmatrix}
\end{aligned} \tag{A.6}$$

Subsystem Model - Zone 6:

$$\begin{aligned}
u_6 &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad Rh_{OA} \quad T_{ZONE,5}]^T \\
A_6 &= [0.2596] \\
B_6 &= [0.0048 \quad 0.3635 \quad 0.0005 \quad 0.0005 \quad 0.0005 \quad 0.0005] \\
C_6 &= [2] \quad K_6 = [0.3247]
\end{aligned} \tag{A.7}$$

Subsystem Model - Zone 7:

$$\begin{aligned}
u_7 &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad Rh_{OA}]^T \\
A_7 &= [0 \quad -0.2476 \quad 0.25 \quad 0.2247] \\
B_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.009 & 0.8244 & 0.001 & 0.001 & 0.001 \end{bmatrix} \\
C_7 &= [0 \quad 1] \quad K_7 = \begin{bmatrix} -0.2476 \\ 0.2247 \end{bmatrix}
\end{aligned} \tag{A.8}$$

Subsystem Model - Zone 8:

$$\begin{aligned}
u_8 &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad Rh_{OA} \quad T_{ZONE,2}]^T \\
A_8 &= [0 \quad -0.7675 \quad 0.5 \quad 1.0458] \\
B_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0102 & 0.3279 & 0.001 & 0.001 & 0.001 & 0.001 \end{bmatrix} \\
C_8 &= [0 \quad 1] \quad K_8 = \begin{bmatrix} -0.7675 \\ 1.0458 \end{bmatrix}
\end{aligned} \tag{A.9}$$

Subsystem Model - Zone 9:

$$\begin{aligned}
u_9 &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad T_{ZONE,10}]^T \\
A_9 &= [-0.0128] \\
B_9 &= [0.0042 \quad 0.9943 \quad 0.001 \quad 0.001 \quad 0.0078] \\
C_9 &= [0 \quad 1] \quad K_9 = [-0.0128]
\end{aligned} \tag{A.10}$$

Subsystem Model - Zone 10:

$$\begin{aligned}
u_{10} &= [T_{OA} \quad T_{ZONE,1}^* \quad T_{DA} \quad P_{EDS} \quad T_{ZONE,9}]^T \\
A_{10} &= [0 \quad -0.5570 \quad 0.5 \quad 0.9957] \\
B_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.008 & 0.2724 & 0.001 & 0.0092 & 0.001 \end{bmatrix} \\
C_{10} &= [0 \quad 1] \quad K_{10} = \begin{bmatrix} -0.557 \\ 1.3372 \end{bmatrix}
\end{aligned} \tag{A.11}$$

A.2 UBO VAV Models

Equation A.12 details the form of the fit used for developing the individual zone models. This equation form was adopted from [93]. Data from the UBO was collected and the fit coefficients for each of the zones are listed below.

$$q_{VAV} = (a \cdot u_d^2 + b \cdot u_d) \sqrt{P_{EDS}} \tag{A.12}$$

VAV 1 coefficients:

$$a = -0.07766 \quad b = 16.08 \tag{A.13}$$

VAV 2 coefficients:

$$a = -0.02071 \quad b = 4.288 \tag{A.14}$$

VAV 3 coefficients:

$$a = -0.03849 \quad b = 7.969 \tag{A.15}$$

VAV 4 coefficients:

$$a = -0.1144 \quad b = 23.69 \quad (\text{A.16})$$

VAV 5 coefficients:

$$a = -0.04725 \quad b = 9.783 \quad (\text{A.17})$$

VAV 6 coefficients:

$$a = -0.06338 \quad b = 13.12 \quad (\text{A.18})$$

VAV 7 coefficients:

$$a = -0.07664 \quad b = 15.87 \quad (\text{A.19})$$

VAV 8 coefficients:

$$a = -0.03224 \quad b = 6.675 \quad (\text{A.20})$$

VAV 9 coefficients:

$$a = -0.2522 \quad b = 52.22 \quad (\text{A.21})$$

VAV 10 coefficients:

$$a = -0.08071 \quad b = 16.71 \quad (\text{A.22})$$

A.3 UBO Chilled Water Valve Fit

An approximate fit of the UBO data for the discharge air temperature and chilled water valve was generated to convert a discharge air temperature setpoint to a chilled water flow (assuming a linear relationship between chilled water valve position and the chilled water flow). The fit of the data is shown in Figure A.1. The fit is defined in Equation A.23:

$$\eta = (3.021T_{DA}^{*2} - 109.4T_{DA}^* + 1002) \quad (\text{A.23})$$

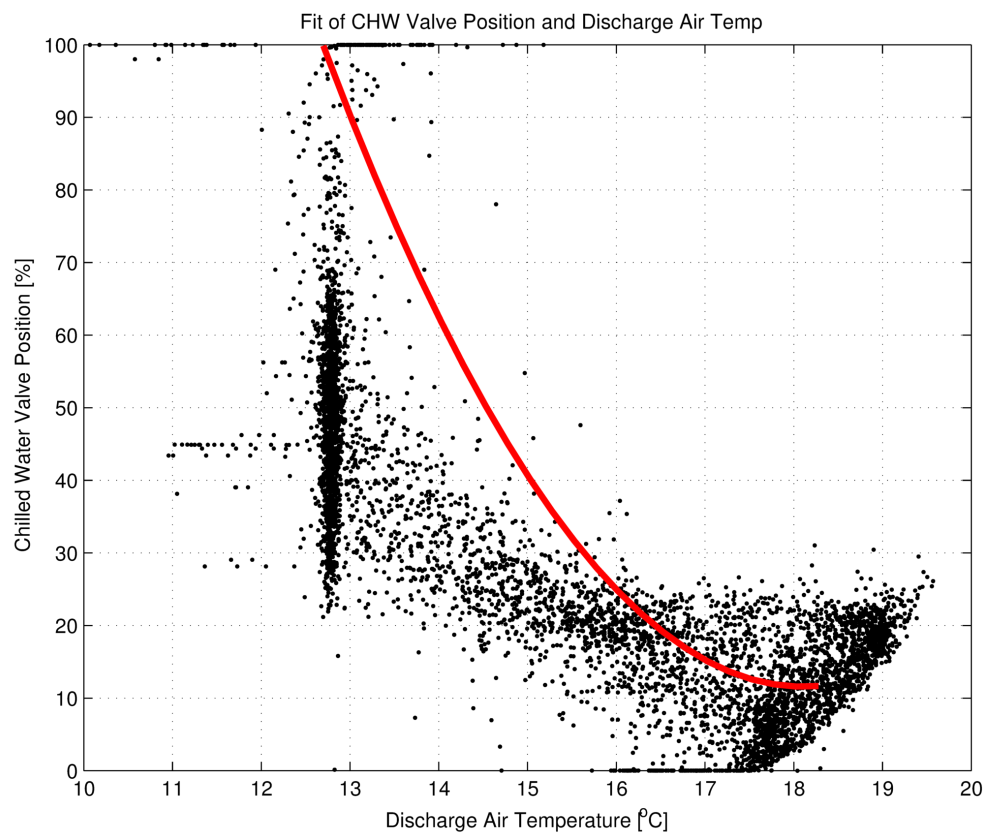


Figure A.1: Fit of the data from the UBO for the discharge air temperature and chilled water valve position.

APPENDIX B

PMV APPROXIMATION

For calculating the Predicted Mean Vote (PMV), the full equation was initially coded. After further inspection, it was determined that a fit of the PMV curve could be made as I was only varying zone air temperature and the zone relative humidity. The main advantage of this was in speeding up the optimization as the full PMV function includes an unconstrained optimization, which when nested within the unconstrained optimization for the minimization of the objective functions, resulted in a significant increase in simulation time. The generated fit is defined as:

$$PMV = 0.5542 \cdot rh + 0.23 \cdot T_{zone} - 5.44 \quad (B.1)$$

Figure B.1 shows the fit of the PMV data.

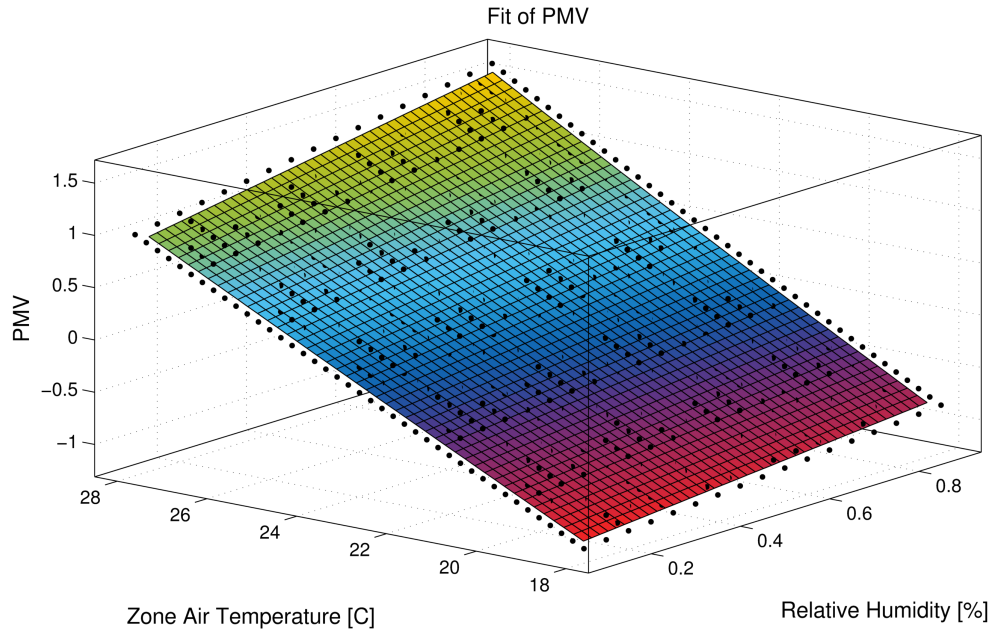


Figure B.1: Fit of the PMV function with varying zone air temperature and relative humidity.

APPENDIX C

ZONE MODEL INPUT MANIPULATION

For the inputs for model identification, the difference between zone temperature and surrounding zone temperatures was used for the identification of interactions between zone temperatures. The purpose of the difference is to reduce the chance of collinearity in the models; however, it would simplify the models and control if the subsystem's zone temperature could be removed from the inputs for its own model, particularly in the dynamic MPC case where the future zone temperatures must be predicted along a horizon. This is possible through manipulating the identified model structure. Recall, the linear, discrete models that were identified have the form of:

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

The input vector u can be written as:

$$u = u' - \Phi y_m \tag{C.2}$$

where u' is the input vector without the subsystem's zone temperature, Φ is a vector of 1's and 0's corresponding to the input terms that did contain the subsystem's zone temperature, and y_m is the measured output of the model. The measured model output can be defined as:

$$y_m = y - e \tag{C.3}$$

where y is the model output and e is the error between the model output and the measured value. With Equations C.2 and C.3, Equation C.1 can be manipulated as follows:

$$\begin{aligned} x(k+1) &= Ax + B(u' - \Phi y_m) + Ke \\ x(k+1) &= Ax + B(u' - \Phi y - \Phi e) + Ke \\ x(k+1) &= Ax + B(u' - \Phi Cx - \Phi e) + Ke \\ x(k+1) &= (A - \Phi BC)x + Bu' + (K - \Phi B)e \end{aligned} \tag{C.4}$$

This derivation is adapted from [92].

APPENDIX D

ENERGYPLUS INPUT FILE

```

!-Generator IDFEditor 1.48
!-Option OriginalOrderTop UseSpecialFormat

!-NOTE: All comments with '!' are ignored by the IDFEditor and are generated
      automatically.
!-      Use '!' comments if they need to be retained when using the IDFEditor.

Schedule:Compact,
  AHU Schedule,           !- Name
  On/Off,                 !- Schedule Type Limits Name
  Through: 12/31,         !- Field 1
  For: Weekdays SummerDesignDay, !- Field 2
  Until: 06:00, 0,        !- Field 4
  Until: 18:00, 1,        !- Field 6
  Until: 24:00, 0,        !- Field 8
  For: Saturday WinterDesignDay, !- Field 9
  Until: 06:00, 0,        !- Field 11
  Until: 18:00, 0,        !- Field 13
  Until: 24:00, 0,        !- Field 15
  For: Sunday Holidays AllOtherDays, !- Field 16
  Until: 24:00, 0;        !- Field 18

Output:Variable,*,System Node Current Density Volume Flow Rate,Timestep;
Output:Variable,*,System Node Temperature,Timestep;
Output:Variable,*,Site Outdoor Air Relative Humidity,Timestep;
Output:Variable,*,Zone Air Relative Humidity,Timestep;
Output:Variable,*,System Node Mass Flow Rate,Timestep;

ExternalInterface:Schedule,
  T_ahu,                  !- Name
  Temperature,            !- Schedule Type Limits Name
  12.8;                   !- Initial Value

ExternalInterface:Schedule,
  People_1,               !- Name
  Fraction,               !- Schedule Type Limits Name
  1;                      !- Initial Value

ExternalInterface:Schedule,
  People_2,               !- Name
  Fraction,               !- Schedule Type Limits Name
  1;                      !- Initial Value

ExternalInterface:Schedule,
  People_3,               !- Name
  Fraction,               !- Schedule Type Limits Name
  1;                      !- Initial Value

ExternalInterface:Schedule,
  People_4,               !- Name
  Fraction,               !- Schedule Type Limits Name
  1;                      !- Initial Value

ExternalInterface:Schedule,
  People_5,               !- Name

```

```

        Fraction ,           !- Schedule Type Limits Name
        1;                   !- Initial Value

    ExternalInterface : Schedule ,
        People_6 ,           !- Name
        Fraction ,           !- Schedule Type Limits Name
        1;                   !- Initial Value

    ExternalInterface : Schedule ,
        People_7 ,           !- Name
        Fraction ,           !- Schedule Type Limits Name
        1;                   !- Initial Value

    ExternalInterface : Schedule ,
        People_8 ,           !- Name
        Fraction ,           !- Schedule Type Limits Name
        1;                   !- Initial Value

    ExternalInterface : Schedule ,
        People_9 ,           !- Name
        Fraction ,           !- Schedule Type Limits Name
        1;                   !- Initial Value

    ExternalInterface : Schedule ,
        People_10 ,          !- Name
        Fraction ,           !- Schedule Type Limits Name
        1;                   !- Initial Value

    ExternalInterface : Schedule ,
        VAV_2 ,              !- Name
        Fraction ,           !- Schedule Type Limits Name
        0.05;                !- Initial Value

    ExternalInterface : Schedule ,
        VAV_3 ,              !- Name
        Fraction ,           !- Schedule Type Limits Name
        0.05;                !- Initial Value

    ExternalInterface : Schedule ,
        VAV_4 ,              !- Name
        Fraction ,           !- Schedule Type Limits Name
        0.05;                !- Initial Value

    ExternalInterface : Schedule ,
        VAV_5 ,              !- Name
        Fraction ,           !- Schedule Type Limits Name
        0.05;                !- Initial Value

    ExternalInterface : Schedule ,
        VAV_6 ,              !- Name
        Fraction ,           !- Schedule Type Limits Name
        0.05;                !- Initial Value

    ExternalInterface : Schedule ,
        VAV_7 ,              !- Name
        Fraction ,           !- Schedule Type Limits Name
        0.05;                !- Initial Value

    ExternalInterface : Schedule ,
        VAV_8 ,              !- Name
        Fraction ,           !- Schedule Type Limits Name
        0.05;                !- Initial Value

    ExternalInterface : Schedule ,

```

```

VAV_9,                                !- Name
Fraction ,                            !- Schedule Type Limits Name
0.05;                                !- Initial Value

ExternalInterface : Schedule ,
VAV_10,                               !- Name
Fraction ,                            !- Schedule Type Limits Name
0.05;                                !- Initial Value

ExternalInterface : Schedule ,
T_hot ,                               !- Name
Temperature ,                         !- Schedule Type Limits Name
18;                                  !- Initial Value

ExternalInterface : Schedule ,
T_cold ,                              !- Name
Temperature ,                         !- Schedule Type Limits Name
20;                                  !- Initial Value

HVACTemplate : Thermostat ,
T_dummy ,                             !- Name
,                                     !- Heating Setpoint Schedule Name
21 ,                                 !- Constant Heating Setpoint {C}
,                                     !- Cooling Setpoint Schedule Name
23;                                 !- Constant Cooling Setpoint {C}

Output: Variable ,*,Zone Air Terminal VAV Damper Position ,Timestep;
Output: Variable ,*,Zone Air Temperature ,Timestep;

ExternalInterface ,
PtolemyServer;                       !- Name of External Interface

ExternalInterface : Schedule ,
VAV_1,                                !- Name
Fraction ,                            !- Schedule Type Limits Name
0.05;                                !- Initial Value

HVACTemplate : Thermostat ,
T1 ,                                  !- Name
T_hot ,                              !- Heating Setpoint Schedule Name
,                                     !- Constant Heating Setpoint {C}
T_cold ,                             !- Cooling Setpoint Schedule Name
;                                     !- Constant Cooling Setpoint {C}

Schedule : Constant , Activity Schedule , Any Number , 80;

Lights ,
Zone 11 Lights ,                     !- Name
Room11 ,                             !- Zone or ZoneList Name
Always On ,                          !- Schedule Name
Watts / Area ,                       !- Design Level Calculation Method
,                                     !- Lighting Level {W}
3 ,                                  !- Watts per Zone Floor Area {W/m2}
,                                     !- Watts per Person {W/person}
,                                     !- Return Air Fraction
,                                     !- Fraction Radiant
,                                     !- Fraction Visible
1 ,                                  !- Fraction Replaceable
General ,                            !- End-Use Subcategory
No;                                  !- Return Air Fraction Calculated from Plenum Temperature

ElectricEquipment ,
Zone 1 Machine ,                     !- Name
Room1 ,                              !- Zone or ZoneList Name

```

Always On,	!- Schedule Name
Watts/Area,	!- Design Level Calculation Method
,	!- Design Level {W}
2,	!- Watts per Zone Floor Area {W/m2}
,	!- Watts per Person {W/person}
0.5,	!- Fraction Latent
0.5,	!- Fraction Radiant
0,	!- Fraction Lost
General;	!- End-Use Subcategory
ElectricEquipment ,	!- Name
Zone 2 Machine ,	!- Zone or ZoneList Name
Room2,	!- Schedule Name
Always On,	!- Design Level Calculation Method
Watts/Area ,	!- Design Level {W}
,	!- Watts per Zone Floor Area {W/m2}
2,	!- Watts per Person {W/person}
,	!- Fraction Latent
0.5,	!- Fraction Radiant
0.5,	!- Fraction Lost
0,	!- End-Use Subcategory
General;	
ElectricEquipment ,	!- Name
Zone 3 Machine ,	!- Zone or ZoneList Name
Room3,	!- Schedule Name
Always On,	!- Design Level Calculation Method
Watts/Area ,	!- Design Level {W}
,	!- Watts per Zone Floor Area {W/m2}
2,	!- Watts per Person {W/person}
,	!- Fraction Latent
0.5,	!- Fraction Radiant
0.5,	!- Fraction Lost
0,	!- End-Use Subcategory
General;	
ElectricEquipment ,	!- Name
Zone 4 Machine ,	!- Zone or ZoneList Name
Room4,	!- Schedule Name
Always On,	!- Design Level Calculation Method
Watts/Area ,	!- Design Level {W}
,	!- Watts per Zone Floor Area {W/m2}
2,	!- Watts per Person {W/person}
,	!- Fraction Latent
0.5,	!- Fraction Radiant
0.5,	!- Fraction Lost
0,	!- End-Use Subcategory
General;	
ElectricEquipment ,	!- Name
Zone 5 Machine ,	!- Zone or ZoneList Name
Room5,	!- Schedule Name
Always On,	!- Design Level Calculation Method
Watts/Area ,	!- Design Level {W}
,	!- Watts per Zone Floor Area {W/m2}
2,	!- Watts per Person {W/person}
,	!- Fraction Latent
0.5,	!- Fraction Radiant
0.5,	!- Fraction Lost
0,	!- End-Use Subcategory
General;	
ElectricEquipment ,	!- Name
Zone 6 Machine ,	

Room6,	!- Zone or ZoneList Name
Always On,	!- Schedule Name
Watts/Area ,	!- Design Level Calculation Method
,	!- Design Level {W}
2,	!- Watts per Zone Floor Area {W/m2}
,	!- Watts per Person {W/person}
0.5,	!- Fraction Latent
0.5,	!- Fraction Radiant
0,	!- Fraction Lost
General;	!- End-Use Subcategory
ElectricEquipment ,	!- Name
Zone 7 Machine ,	!- Zone or ZoneList Name
Room7,	!- Schedule Name
Always On,	!- Design Level Calculation Method
Watts/Area ,	!- Design Level {W}
,	!- Watts per Zone Floor Area {W/m2}
2,	!- Watts per Person {W/person}
,	!- Fraction Latent
0.5,	!- Fraction Radiant
0.5,	!- Fraction Lost
0,	!- End-Use Subcategory
General;	
ElectricEquipment ,	!- Name
Zone 8 Machine ,	!- Zone or ZoneList Name
Room8,	!- Schedule Name
Always On,	!- Design Level Calculation Method
Watts/Area ,	!- Design Level {W}
,	!- Watts per Zone Floor Area {W/m2}
2,	!- Watts per Person {W/person}
,	!- Fraction Latent
0.5,	!- Fraction Radiant
0.5,	!- Fraction Lost
0,	!- End-Use Subcategory
General;	
ElectricEquipment ,	!- Name
Zone 9 Machine ,	!- Zone or ZoneList Name
Room9,	!- Schedule Name
Always On,	!- Design Level Calculation Method
Watts/Area ,	!- Design Level {W}
,	!- Watts per Zone Floor Area {W/m2}
2,	!- Watts per Person {W/person}
,	!- Fraction Latent
0.5,	!- Fraction Radiant
0.5,	!- Fraction Lost
0,	!- End-Use Subcategory
General;	
ElectricEquipment ,	!- Name
Zone 11 Machine ,	!- Zone or ZoneList Name
Room11,	!- Schedule Name
Always On,	!- Design Level Calculation Method
Watts/Area ,	!- Design Level {W}
,	!- Watts per Zone Floor Area {W/m2}
1200,	!- Watts per Person {W/person}
,	!- Fraction Latent
0.5,	!- Fraction Radiant
0.5,	!- Fraction Lost
0,	!- End-Use Subcategory
General;	
ElectricEquipment ,	

```

Zone 10 Machine ,      !- Name
Room10 ,               !- Zone or ZoneList Name
Always On ,            !- Schedule Name
Watts / Area ,         !- Design Level Calculation Method
 ,                     !- Design Level {W}
2 ,                    !- Watts per Zone Floor Area {W/m2}
 ,                     !- Watts per Person {W/person}
0.5 ,                  !- Fraction Latent
0.5 ,                  !- Fraction Radiant
0 ,                    !- Fraction Lost
General ;              !- End-Use Subcategory

Material ,
  G01b 19mm gypsum board , !- Name
  MediumSmooth ,           !- Roughness
  0.019 ,                  !- Thickness {m}
  0.5 ,                    !- Conductivity {W/m-K}
  800 ,                    !- Density {kg/m3}
  1500 ;                   !- Specific Heat {J/kg-K}

People ,
  Room 11 People ,        !- Name
  Room11 ,                !- Zone or ZoneList Name
  Always On ,             !- Number of People Schedule Name
  People ,                !- Number of People Calculation Method
  2 ,                     !- Number of People
  ,                       !- People per Zone Floor Area {person/m2}
  ,                       !- Zone Floor Area per Person {m2/person}
  0.3 ,                   !- Fraction Radiant
  autocalculate ,         !- Sensible Heat Fraction
  Activity Schedule ,      !- Activity Level Schedule Name
  0.0000000382 ,          !- Carbon Dioxide Generation Rate {m3/s-W}
  No ,                    !- Enable ASHRAE 55 Comfort Warnings
  ZoneAveraged ,          !- Mean Radiant Temperature Calculation Type
  ,                       !- Surface Name/Angle Factor List Name
  ,                       !- Work Efficiency Schedule Name
  ClothingInsulationSchedule ; !- Clothing Insulation Calculation Method

HVACTemplate : Zone : VAV ,
  Room11 ,                !- Zone Name
  Main AHU ,              !- Template VAV System Name
  T dummy ,               !- Template Thermostat Name
  0.7 ,                   !- Supply Air Maximum Flow Rate {m3/s}
  ,                       !- Zone Heating Sizing Factor
  ,                       !- Zone Cooling Sizing Factor
  Constant ,              !- Zone Minimum Air Flow Input Method
  0.05 ,                  !- Constant Minimum Air Flow Fraction
  ,                       !- Fixed Minimum Air Flow Rate {m3/s}
  ,                       !- Minimum Air Flow Fraction Schedule Name
  Flow / Person ,         !- Outdoor Air Method
  0.00944 ,               !- Outdoor Air Flow Rate per Person {m3/s}
  0 ,                     !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
  0 ,                     !- Outdoor Air Flow Rate per Zone {m3/s}
  None ,                  !- Reheat Coil Type
  ,                       !- Reheat Coil Availability Schedule Name
  Normal ,                !- Damper Heating Action
  ,                       !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
  ,                       !- Maximum Flow Fraction During Reheat
  ,                       !- Maximum Reheat Air Temperature {C}
  ,                       !- Design Specification Outdoor Air Object Name for Control
  ,                       !- Supply Plenum Name
  ,                       !- Return Plenum Name
  None ,                  !- Baseboard Heating Type
  ,                       !- Baseboard Heating Availability Schedule Name

```

```

autosize ,                !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature , !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7 ,                    !- Zone Cooling Design Supply Air Temperature {C}
11.11 ,                   !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature ,    !- Zone Heating Design Supply Air Temperature Input Method
50 ,                      !- Zone Heating Design Supply Air Temperature {C}
;                          !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate:Plant:Boiler ,
    MainBoiler ,          !- Name
    HotWaterBoiler ,      !- Boiler Type
    100000 ,              !- Capacity {W}
    0.8 ,                 !- Efficiency
    NaturalGas ,          !- Fuel Type
    ,                     !- Priority
    ;                     !- Sizing Factor

HVACTemplate:Plant:HotWaterLoop ,
    HotWaterLoop ,        !- Name
    ,                     !- Pump Schedule Name
    Intermittent ,        !- Pump Control Type
    Default ,             !- Hot Water Plant Operation Scheme Type
    ,                     !- Hot Water Plant Equipment Operation Schemes Name
    ,                     !- Hot Water Setpoint Schedule Name
    82 ,                  !- Hot Water Design Setpoint {C}
    ConstantFlow ,        !- Hot Water Pump Configuration
    179352 ,              !- Hot Water Pump Rated Head {Pa}
    OutdoorAirTemperatureReset , !- Hot Water Setpoint Reset Type
    82.2 ,                !- Hot Water Setpoint at Outdoor Dry-Bulb Low {C}
    -6.7 ,                !- Hot Water Reset Outdoor Dry-Bulb Low {C}
    65.6 ,                !- Hot Water Setpoint at Outdoor Dry-Bulb High {C}
    10 ;                  !- Hot Water Reset Outdoor Dry-Bulb High {C}

SurfaceConvectionAlgorithm:Inside,TARP;
SurfaceConvectionAlgorithm:Outside,DOE-2;
HeatBalanceAlgorithm,ConductionTransferFunction;

ConvergenceLimits ,
    1 ,                   !- Minimum System Timestep {minutes}
    20 ,                  !- Maximum HVAC Iterations
    4 ,                   !- Minimum Plant Iterations
    10 ;                  !- Maximum Plant Iterations

Site:GroundTemperature:BuildingSurface
    ,21.5,21.4,21.5,21.5,22,22.9,23,23.1,22.2,21.7,21.7,21.6;

ZoneInfiltration:DesignFlowRate ,
    Room1 Infill ,        !- Name
    Room1 ,               !- Zone or ZoneList Name
    InfiltrationSchedule , !- Schedule Name
    AirChanges/Hour ,     !- Design Flow Rate Calculation Method
    ,                     !- Design Flow Rate {m3/s}
    ,                     !- Flow per Zone Floor Area {m3/s-m2}
    ,                     !- Flow per Exterior Surface Area {m3/s-m2}
    0.3 ,                 !- Air Changes per Hour {1/hr}
    0 ,                   !- Constant Term Coefficient
    0 ,                   !- Temperature Term Coefficient
    0.2237 ,              !- Velocity Term Coefficient
    0 ;                   !- Velocity Squared Term Coefficient

ZoneInfiltration:DesignFlowRate ,

```

```

Room2 Infil ,           !- Name
Room2,                 !- Zone or ZoneList Name
Infiltration Schedule , !- Schedule Name
AirChanges/Hour ,      !- Design Flow Rate Calculation Method
,                      !- Design Flow Rate {m3/s}
,                      !- Flow per Zone Floor Area {m3/s-m2}
,                      !- Flow per Exterior Surface Area {m3/s-m2}
0.3 ,                 !- Air Changes per Hour {1/hr}
0 ,                   !- Constant Term Coefficient
0 ,                   !- Temperature Term Coefficient
0.2237 ,              !- Velocity Term Coefficient
0 ;                   !- Velocity Squared Term Coefficient

ZoneInfiltration:DesignFlowRate ,
Room3 Infil ,         !- Name
Room3,                !- Zone or ZoneList Name
Infiltration Schedule , !- Schedule Name
AirChanges/Hour ,     !- Design Flow Rate Calculation Method
,                     !- Design Flow Rate {m3/s}
,                     !- Flow per Zone Floor Area {m3/s-m2}
,                     !- Flow per Exterior Surface Area {m3/s-m2}
0.3 ,                 !- Air Changes per Hour {1/hr}
0 ,                   !- Constant Term Coefficient
0 ,                   !- Temperature Term Coefficient
0.2237 ,              !- Velocity Term Coefficient
0 ;                   !- Velocity Squared Term Coefficient

ZoneInfiltration:DesignFlowRate ,
Room4 Infil ,         !- Name
Room4,                !- Zone or ZoneList Name
Infiltration Schedule , !- Schedule Name
AirChanges/Hour ,     !- Design Flow Rate Calculation Method
,                     !- Design Flow Rate {m3/s}
,                     !- Flow per Zone Floor Area {m3/s-m2}
,                     !- Flow per Exterior Surface Area {m3/s-m2}
0.3 ,                 !- Air Changes per Hour {1/hr}
0 ,                   !- Constant Term Coefficient
0 ,                   !- Temperature Term Coefficient
0.2237 ,              !- Velocity Term Coefficient
0 ;                   !- Velocity Squared Term Coefficient

ZoneInfiltration:DesignFlowRate ,
Room5 Infil ,         !- Name
Room5,                !- Zone or ZoneList Name
Infiltration Schedule , !- Schedule Name
AirChanges/Hour ,     !- Design Flow Rate Calculation Method
,                     !- Design Flow Rate {m3/s}
,                     !- Flow per Zone Floor Area {m3/s-m2}
,                     !- Flow per Exterior Surface Area {m3/s-m2}
0.3 ,                 !- Air Changes per Hour {1/hr}
0 ,                   !- Constant Term Coefficient
0 ,                   !- Temperature Term Coefficient
0.2237 ,              !- Velocity Term Coefficient
0 ;                   !- Velocity Squared Term Coefficient

ZoneInfiltration:DesignFlowRate ,
Room6 Infil ,         !- Name
Room6,                !- Zone or ZoneList Name
Infiltration Schedule , !- Schedule Name
AirChanges/Hour ,     !- Design Flow Rate Calculation Method
,                     !- Design Flow Rate {m3/s}
,                     !- Flow per Zone Floor Area {m3/s-m2}
,                     !- Flow per Exterior Surface Area {m3/s-m2}
0.3 ,                 !- Air Changes per Hour {1/hr}

```

```

0,                !- Constant Term Coefficient
0,                !- Temperature Term Coefficient
0.2237,          !- Velocity Term Coefficient
0;               !- Velocity Squared Term Coefficient

ZoneInfiltration:DesignFlowRate,
Room7 Infil,      !- Name
Room7,            !- Zone or ZoneList Name
Infiltration Schedule, !- Schedule Name
AirChanges/Hour,  !- Design Flow Rate Calculation Method
,                !- Design Flow Rate {m3/s}
,                !- Flow per Zone Floor Area {m3/s-m2}
,                !- Flow per Exterior Surface Area {m3/s-m2}
0.3,             !- Air Changes per Hour {1/hr}
0,               !- Constant Term Coefficient
0,               !- Temperature Term Coefficient
0.2237,          !- Velocity Term Coefficient
0;               !- Velocity Squared Term Coefficient

ZoneInfiltration:DesignFlowRate,
Room8 Infil,      !- Name
Room8,            !- Zone or ZoneList Name
Infiltration Schedule, !- Schedule Name
AirChanges/Hour,  !- Design Flow Rate Calculation Method
,                !- Design Flow Rate {m3/s}
,                !- Flow per Zone Floor Area {m3/s-m2}
,                !- Flow per Exterior Surface Area {m3/s-m2}
0.3,             !- Air Changes per Hour {1/hr}
0,               !- Constant Term Coefficient
0,               !- Temperature Term Coefficient
0.2237,          !- Velocity Term Coefficient
0;               !- Velocity Squared Term Coefficient

ZoneInfiltration:DesignFlowRate,
Room9 Infil,      !- Name
Room9,            !- Zone or ZoneList Name
Infiltration Schedule, !- Schedule Name
Flow/ExteriorWallArea, !- Design Flow Rate Calculation Method
,                !- Design Flow Rate {m3/s}
,                !- Flow per Zone Floor Area {m3/s-m2}
,                !- Flow per Exterior Surface Area {m3/s-m2}
0.3,             !- Air Changes per Hour {1/hr}
0,               !- Constant Term Coefficient
0,               !- Temperature Term Coefficient
0.2237,          !- Velocity Term Coefficient
0;               !- Velocity Squared Term Coefficient

ZoneInfiltration:DesignFlowRate,
Room10 Infil,     !- Name
Room10,           !- Zone or ZoneList Name
Infiltration Schedule, !- Schedule Name
AirChanges/Hour,  !- Design Flow Rate Calculation Method
,                !- Design Flow Rate {m3/s}
,                !- Flow per Zone Floor Area {m3/s-m2}
,                !- Flow per Exterior Surface Area {m3/s-m2}
0.3,             !- Air Changes per Hour {1/hr}
0,               !- Constant Term Coefficient
0,               !- Temperature Term Coefficient
0.2237,          !- Velocity Term Coefficient
0;               !- Velocity Squared Term Coefficient

HVACTemplate:Plant:Tower,
Main Tower,       !- Name
SingleSpeed,      !- Tower Type

```

```

autosize ,           !- High Speed Nominal Capacity {W}
autosize ,           !- High Speed Fan Power {W}
autosize ,           !- Low Speed Nominal Capacity {W}
autosize ,           !- Low Speed Fan Power {W}
autosize ,           !- Free Convection Capacity {W}
1 ,                  !- Priority
;                    !- Sizing Factor

SizingPeriod:DesignDay ,
  CHICAGO_IL_USA Annual Heating 99% Design Conditions DB, !- Name
  1 ,                 !- Month
  8 ,                 !- Day of Month
  WinterDesignDay ,  !- Day Type
  -17.3 ,             !- Maximum Dry-Bulb Temperature {C}
  0.0 ,               !- Daily Dry-Bulb Temperature Range {deltaC}
  ,                  !- Dry-Bulb Temperature Range Modifier Type
  ,                  !- Dry-Bulb Temperature Range Modifier Day Schedule Name
  Wetbulb ,          !- Humidity Condition Type
  -17.3 ,             !- Wetbulb or DewPoint at Maximum Dry-Bulb {C}
  ,                  !- Humidity Condition Day Schedule Name
  ,                  !- Humidity Ratio at Maximum Dry-Bulb {kgWater/kgDryAir}
  ,                  !- Enthalpy at Maximum Dry-Bulb {J/kg}
  ,                  !- Daily Wet-Bulb Temperature Range {deltaC}
  99063. ,           !- Barometric Pressure {Pa}
  4.9 ,              !- Wind Speed {m/s}
  270 ,              !- Wind Direction {deg}
  No ,               !- Rain Indicator
  No ,               !- Snow Indicator
  No ,               !- Daylight Saving Time Indicator
  ASHRAEClearSky ,  !- Solar Model Indicator
  ,                  !- Beam Solar Day Schedule Name
  ,                  !- Diffuse Solar Day Schedule Name
  ,                  !- ASHRAE Clear Sky Optical Depth for Beam Irradiance (taub)
    { dimensionless }
  ,                  !- ASHRAE Clear Sky Optical Depth for Diffuse Irradiance (
    taud ) { dimensionless }
  0.0 ;              !- Sky Clearness

SizingPeriod:DesignDay ,
  CHICAGO_IL_USA Annual Cooling 1% Design Conditions DB/MCWB, !- Name
  7 ,                 !- Month
  3 ,                 !- Day of Month
  SummerDesignDay ,  !- Day Type
  70 ,                !- Maximum Dry-Bulb Temperature {C}
  65 ,                !- Daily Dry-Bulb Temperature Range {deltaC}
  ,                  !- Dry-Bulb Temperature Range Modifier Type
  ,                  !- Dry-Bulb Temperature Range Modifier Day Schedule Name
  Wetbulb ,          !- Humidity Condition Type
  23.0 ,              !- Wetbulb or DewPoint at Maximum Dry-Bulb {C}
  ,                  !- Humidity Condition Day Schedule Name
  ,                  !- Humidity Ratio at Maximum Dry-Bulb {kgWater/kgDryAir}
  ,                  !- Enthalpy at Maximum Dry-Bulb {J/kg}
  ,                  !- Daily Wet-Bulb Temperature Range {deltaC}
  99063. ,           !- Barometric Pressure {Pa}
  5.3 ,              !- Wind Speed {m/s}
  230 ,              !- Wind Direction {deg}
  No ,               !- Rain Indicator
  No ,               !- Snow Indicator
  No ,               !- Daylight Saving Time Indicator
  ASHRAEClearSky ,  !- Solar Model Indicator
  ,                  !- Beam Solar Day Schedule Name
  ,                  !- Diffuse Solar Day Schedule Name
  ,                  !- ASHRAE Clear Sky Optical Depth for Beam Irradiance (taub)
    { dimensionless }

```

```

,                                     !- ASHRAE Clear Sky Optical Depth for Diffuse Irradiance (
    tau) {dimensionless}
1.0;                                !- Sky Clearness

HVACTemplate: System: VAV,
Main AHU,                           !- Name
Always On,                          !- System Availability Schedule Name
autosize,                           !- Supply Fan Maximum Flow Rate {m3/s}
0,                                  !- Supply Fan Minimum Flow Rate {m3/s}
0.7,                                !- Supply Fan Total Efficiency
1000,                               !- Supply Fan Delta Pressure {Pa}
0.9,                                !- Supply Fan Motor Efficiency
1,                                  !- Supply Fan Motor in Air Stream Fraction
ChilledWater,                       !- Cooling Coil Type
,                                    !- Cooling Coil Availability Schedule Name
T_ahu,                              !- Cooling Coil Setpoint Schedule Name
10,                                 !- Cooling Coil Design Setpoint {C}
HotWater,                           !- Heating Coil Type
,                                    !- Heating Coil Availability Schedule Name
,                                    !- Heating Coil Setpoint Schedule Name
5,                                  !- Heating Coil Design Setpoint {C}
0.8,                                !- Gas Heating Coil Efficiency
0,                                  !- Gas Heating Coil Parasitic Electric Load {W}
None,                               !- Preheat Coil Type
,                                    !- Preheat Coil Availability Schedule Name
,                                    !- Preheat Coil Setpoint Schedule Name
,                                    !- Preheat Coil Design Setpoint {C}
0.8,                                !- Gas Preheat Coil Efficiency
0,                                  !- Gas Preheat Coil Parasitic Electric Load {W}
autosize,                           !- Maximum Outdoor Air Flow Rate {m3/s}
autosize,                           !- Minimum Outdoor Air Flow Rate {m3/s}
FixedMinimum,                       !- Minimum Outdoor Air Control Type
Always On,                          !- Minimum Outdoor Air Schedule Name
FixedDryBulb,                       !- Economizer Type
NoLockout,                          !- Economizer Lockout
25,                                 !- Economizer Upper Temperature Limit {C}
4,                                  !- Economizer Lower Temperature Limit {C}
,                                    !- Economizer Upper Enthalpy Limit {J/kg}
,                                    !- Economizer Maximum Limit Dewpoint Temperature {C}
,                                    !- Supply Plenum Name
,                                    !- Return Plenum Name
DrawThrough,                        !- Supply Fan Placement
InletVaneDampers,                   !- Supply Fan Part-Load Power Coefficients
CycleOnAny,                         !- Night Cycle Control
,                                    !- Night Cycle Control Zone Name
None,                               !- Heat Recovery Type
0.7,                                !- Sensible Heat Recovery Effectiveness
0.65,                               !- Latent Heat Recovery Effectiveness
None,                               !- Cooling Coil Setpoint Reset Type
None,                               !- Heating Coil Setpoint Reset Type
None,                               !- Dehumidification Control Type
,                                    !- Dehumidification Control Zone Name
60,                                 !- Dehumidification Setpoint {percent}
None,                               !- Humidifier Type
,                                    !- Humidifier Availability Schedule Name
0.000001,                           !- Humidifier Rated Capacity {m3/s}
0,                                  !- Humidifier Rated Electric Power {W}
,                                    !- Humidifier Control Zone Name
0,                                  !- Humidifier Setpoint {percent}
NonCoincident,                     !- Sizing Option
,                                    !- Return Fan
,                                    !- Return Fan Total Efficiency
,                                    !- Return Fan Delta Pressure {Pa}
,                                    !- Return Fan Motor Efficiency

```

```

,                                     !- Return Fan Motor in Air Stream Fraction
;                                     !- Return Fan Part-Load Power Coefficients

Output:Diagnostics ,
  DisplayExtraWarnings;    !- Key 1

People ,
  Room 1 People ,           !- Name
  Room1,                   !- Zone or ZoneList Name
  People_1 ,               !- Number of People Schedule Name
  People ,                 !- Number of People Calculation Method
  2,                       !- Number of People
  ,                       !- People per Zone Floor Area {person/m2}
  ,                       !- Zone Floor Area per Person {m2/person}
  0.3,                    !- Fraction Radiant
  autocalculate ,         !- Sensible Heat Fraction
  Activity Schedule ,      !- Activity Level Schedule Name
  0.0000000382,          !- Carbon Dioxide Generation Rate {m3/s-W}
  No,                     !- Enable ASHRAE 55 Comfort Warnings
  ZoneAveraged ,          !- Mean Radiant Temperature Calculation Type
  ,                       !- Surface Name/Angle Factor List Name
  ,                       !- Work Efficiency Schedule Name
  ClothingInsulationSchedule; !- Clothing Insulation Calculation Method

People ,
  Room 2 People ,           !- Name
  Room2,                   !- Zone or ZoneList Name
  People_2 ,               !- Number of People Schedule Name
  People ,                 !- Number of People Calculation Method
  2,                       !- Number of People
  ,                       !- People per Zone Floor Area {person/m2}
  ,                       !- Zone Floor Area per Person {m2/person}
  0.3,                    !- Fraction Radiant
  autocalculate ,         !- Sensible Heat Fraction
  Activity Schedule ,      !- Activity Level Schedule Name
  0.0000000382,          !- Carbon Dioxide Generation Rate {m3/s-W}
  No,                     !- Enable ASHRAE 55 Comfort Warnings
  ZoneAveraged ,          !- Mean Radiant Temperature Calculation Type
  ,                       !- Surface Name/Angle Factor List Name
  ,                       !- Work Efficiency Schedule Name
  ClothingInsulationSchedule; !- Clothing Insulation Calculation Method

People ,
  Room 3 People ,           !- Name
  Room3,                   !- Zone or ZoneList Name
  People_3 ,               !- Number of People Schedule Name
  People ,                 !- Number of People Calculation Method
  2,                       !- Number of People
  ,                       !- People per Zone Floor Area {person/m2}
  ,                       !- Zone Floor Area per Person {m2/person}
  0.3,                    !- Fraction Radiant
  autocalculate ,         !- Sensible Heat Fraction
  Activity Schedule ,      !- Activity Level Schedule Name
  0.0000000382,          !- Carbon Dioxide Generation Rate {m3/s-W}
  No,                     !- Enable ASHRAE 55 Comfort Warnings
  ZoneAveraged ,          !- Mean Radiant Temperature Calculation Type
  ,                       !- Surface Name/Angle Factor List Name
  ,                       !- Work Efficiency Schedule Name
  ClothingInsulationSchedule; !- Clothing Insulation Calculation Method

People ,
  Room 4 People ,           !- Name
  Room4,                   !- Zone or ZoneList Name
  People_4 ,               !- Number of People Schedule Name

```

```

People ,                                !- Number of People Calculation Method
2,                                     !- Number of People
,                                     !- People per Zone Floor Area {person/m2}
,                                     !- Zone Floor Area per Person {m2/person}
0.3,                                  !- Fraction Radiant
autocalculate ,                       !- Sensible Heat Fraction
Activity Schedule ,                   !- Activity Level Schedule Name
0.0000000382,                        !- Carbon Dioxide Generation Rate {m3/s-W}
No,                                   !- Enable ASHRAE 55 Comfort Warnings
ZoneAveraged ,                       !- Mean Radiant Temperature Calculation Type
,                                     !- Surface Name/Angle Factor List Name
,                                     !- Work Efficiency Schedule Name
ClothingInsulationSchedule;          !- Clothing Insulation Calculation Method

People ,
Room 5 People ,                      !- Name
Room5,                               !- Zone or ZoneList Name
People_5 ,                           !- Number of People Schedule Name
People ,                             !- Number of People Calculation Method
2,                                   !- Number of People
,                                     !- People per Zone Floor Area {person/m2}
,                                     !- Zone Floor Area per Person {m2/person}
0.3,                                  !- Fraction Radiant
autocalculate ,                       !- Sensible Heat Fraction
Activity Schedule ,                   !- Activity Level Schedule Name
0.0000000382,                        !- Carbon Dioxide Generation Rate {m3/s-W}
No,                                   !- Enable ASHRAE 55 Comfort Warnings
ZoneAveraged ,                       !- Mean Radiant Temperature Calculation Type
,                                     !- Surface Name/Angle Factor List Name
,                                     !- Work Efficiency Schedule Name
ClothingInsulationSchedule;          !- Clothing Insulation Calculation Method

People ,
Room 6 People ,                      !- Name
Room6,                               !- Zone or ZoneList Name
People_6 ,                           !- Number of People Schedule Name
People ,                             !- Number of People Calculation Method
2,                                   !- Number of People
,                                     !- People per Zone Floor Area {person/m2}
,                                     !- Zone Floor Area per Person {m2/person}
0.3,                                  !- Fraction Radiant
autocalculate ,                       !- Sensible Heat Fraction
Activity Schedule ,                   !- Activity Level Schedule Name
0.0000000382,                        !- Carbon Dioxide Generation Rate {m3/s-W}
No,                                   !- Enable ASHRAE 55 Comfort Warnings
ZoneAveraged ,                       !- Mean Radiant Temperature Calculation Type
,                                     !- Surface Name/Angle Factor List Name
,                                     !- Work Efficiency Schedule Name
ClothingInsulationSchedule;          !- Clothing Insulation Calculation Method

People ,
Room 7 People ,                      !- Name
Room7,                               !- Zone or ZoneList Name
People_7 ,                           !- Number of People Schedule Name
People ,                             !- Number of People Calculation Method
2,                                   !- Number of People
,                                     !- People per Zone Floor Area {person/m2}
,                                     !- Zone Floor Area per Person {m2/person}
0.3,                                  !- Fraction Radiant
autocalculate ,                       !- Sensible Heat Fraction
Activity Schedule ,                   !- Activity Level Schedule Name
0.0000000382,                        !- Carbon Dioxide Generation Rate {m3/s-W}
No,                                   !- Enable ASHRAE 55 Comfort Warnings
ZoneAveraged ,                       !- Mean Radiant Temperature Calculation Type

```

```

,                                     !- Surface Name/Angle Factor List Name
,                                     !- Work Efficiency Schedule Name
ClothingInsulationSchedule; !- Clothing Insulation Calculation Method

People ,
Room 8 People ,                       !- Name
Room8,                               !- Zone or ZoneList Name
People_8 ,                            !- Number of People Schedule Name
People ,                              !- Number of People Calculation Method
2,                                    !- Number of People
,                                     !- People per Zone Floor Area {person/m2}
,                                     !- Zone Floor Area per Person {m2/person}
0.3,                                  !- Fraction Radiant
autocalculate ,                      !- Sensible Heat Fraction
Activity Schedule ,                  !- Activity Level Schedule Name
0.0000000382,                       !- Carbon Dioxide Generation Rate {m3/s-W}
No,                                  !- Enable ASHRAE 55 Comfort Warnings
ZoneAveraged ,                      !- Mean Radiant Temperature Calculation Type
,                                     !- Surface Name/Angle Factor List Name
,                                     !- Work Efficiency Schedule Name
ClothingInsulationSchedule; !- Clothing Insulation Calculation Method

People ,
Room 9 People ,                       !- Name
Room9,                               !- Zone or ZoneList Name
People_9 ,                            !- Number of People Schedule Name
People ,                              !- Number of People Calculation Method
2,                                    !- Number of People
,                                     !- People per Zone Floor Area {person/m2}
,                                     !- Zone Floor Area per Person {m2/person}
0.3,                                  !- Fraction Radiant
autocalculate ,                      !- Sensible Heat Fraction
Activity Schedule ,                  !- Activity Level Schedule Name
0.0000000382,                       !- Carbon Dioxide Generation Rate {m3/s-W}
No,                                  !- Enable ASHRAE 55 Comfort Warnings
ZoneAveraged ,                      !- Mean Radiant Temperature Calculation Type
,                                     !- Surface Name/Angle Factor List Name
,                                     !- Work Efficiency Schedule Name
ClothingInsulationSchedule; !- Clothing Insulation Calculation Method

People ,
Room 10 People ,                     !- Name
Room10,                              !- Zone or ZoneList Name
People_10 ,                           !- Number of People Schedule Name
People ,                              !- Number of People Calculation Method
2,                                    !- Number of People
,                                     !- People per Zone Floor Area {person/m2}
,                                     !- Zone Floor Area per Person {m2/person}
0.3,                                  !- Fraction Radiant
autocalculate ,                      !- Sensible Heat Fraction
Activity Schedule ,                  !- Activity Level Schedule Name
0.0000000382,                       !- Carbon Dioxide Generation Rate {m3/s-W}
No,                                  !- Enable ASHRAE 55 Comfort Warnings
ZoneAveraged ,                      !- Mean Radiant Temperature Calculation Type
,                                     !- Surface Name/Angle Factor List Name
,                                     !- Work Efficiency Schedule Name
ClothingInsulationSchedule; !- Clothing Insulation Calculation Method

HVACTemplate:Zone:VAV,
Room1,                               !- Zone Name
Main AHU,                            !- Template VAV System Name
T1,                                   !- Template Thermostat Name
0.52642912760326,                   !- Supply Air Maximum Flow Rate {m3/s}
,                                     !- Zone Heating Sizing Factor

```

```

,                                     !- Zone Cooling Sizing Factor
Scheduled ,                          !- Zone Minimum Air Flow Input Method
0.05 ,                               !- Constant Minimum Air Flow Fraction
,                                     !- Fixed Minimum Air Flow Rate {m3/s}
VAV_1,                               !- Minimum Air Flow Fraction Schedule Name
Flow/Person ,                        !- Outdoor Air Method
0.00944 ,                            !- Outdoor Air Flow Rate per Person {m3/s}
0 ,                                  !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0 ,                                  !- Outdoor Air Flow Rate per Zone {m3/s}
None ,                               !- Reheat Coil Type
,                                     !- Reheat Coil Availability Schedule Name
Normal ,                             !- Damper Heating Action
,                                     !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,                                     !- Maximum Flow Fraction During Reheat
,                                     !- Maximum Reheat Air Temperature {C}
,                                     !- Design Specification Outdoor Air Object Name for Control
,                                     !- Supply Plenum Name
,                                     !- Return Plenum Name
None ,                               !- Baseboard Heating Type
,                                     !- Baseboard Heating Availability Schedule Name
autosize ,                           !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature ,         !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7 ,                               !- Zone Cooling Design Supply Air Temperature {C}
11.11 ,                              !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature ,               !- Zone Heating Design Supply Air Temperature Input Method
50 ,                                 !- Zone Heating Design Supply Air Temperature {C}
;                                     !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate:Zone:VAV,
Room2,                               !- Zone Name
Main AHU,                            !- Template VAV System Name
T1,                                  !- Template Thermostat Name
0.140376879467937,                  !- Supply Air Maximum Flow Rate {m3/s}
,                                     !- Zone Heating Sizing Factor
,                                     !- Zone Cooling Sizing Factor
Scheduled ,                          !- Zone Minimum Air Flow Input Method
0.05 ,                               !- Constant Minimum Air Flow Fraction
,                                     !- Fixed Minimum Air Flow Rate {m3/s}
VAV_2,                               !- Minimum Air Flow Fraction Schedule Name
Flow/Person ,                        !- Outdoor Air Method
0.00944 ,                            !- Outdoor Air Flow Rate per Person {m3/s}
0 ,                                  !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0 ,                                  !- Outdoor Air Flow Rate per Zone {m3/s}
None ,                               !- Reheat Coil Type
,                                     !- Reheat Coil Availability Schedule Name
Normal ,                             !- Damper Heating Action
,                                     !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,                                     !- Maximum Flow Fraction During Reheat
,                                     !- Maximum Reheat Air Temperature {C}
,                                     !- Design Specification Outdoor Air Object Name for Control
,                                     !- Supply Plenum Name
,                                     !- Return Plenum Name
None ,                               !- Baseboard Heating Type
,                                     !- Baseboard Heating Availability Schedule Name
autosize ,                           !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature ,         !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7 ,                               !- Zone Cooling Design Supply Air Temperature {C}
11.11 ,                              !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature ,               !- Zone Heating Design Supply Air Temperature Input Method

```

```

50,                !- Zone Heating Design Supply Air Temperature {C}
;                  !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate: Zone: VAV,
Room3,              !- Zone Name
Main AHU,           !- Template VAV System Name
T1,                 !- Template Thermostat Name
0.260871783224132, !- Supply Air Maximum Flow Rate {m3/s}
,                   !- Zone Heating Sizing Factor
,                   !- Zone Cooling Sizing Factor
Scheduled,          !- Zone Minimum Air Flow Input Method
0.05,               !- Constant Minimum Air Flow Fraction
,                   !- Fixed Minimum Air Flow Rate {m3/s}
VAV_3,              !- Minimum Air Flow Fraction Schedule Name
Flow/Person,        !- Outdoor Air Method
0.00944,            !- Outdoor Air Flow Rate per Person {m3/s}
0,                  !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0,                  !- Outdoor Air Flow Rate per Zone {m3/s}
None,               !- Reheat Coil Type
,                   !- Reheat Coil Availability Schedule Name
Normal,             !- Damper Heating Action
,                   !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,                   !- Maximum Flow Fraction During Reheat
,                   !- Maximum Reheat Air Temperature {C}
,                   !- Design Specification Outdoor Air Object Name for Control
,                   !- Supply Plenum Name
,                   !- Return Plenum Name
None,               !- Baseboard Heating Type
,                   !- Baseboard Heating Availability Schedule Name
autosize,           !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature, !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7,               !- Zone Cooling Design Supply Air Temperature {C}
11.11,              !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature, !- Zone Heating Design Supply Air Temperature Input Method
50,                 !- Zone Heating Design Supply Air Temperature {C}
;                   !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate: Zone: VAV,
Room4,              !- Zone Name
Main AHU,           !- Template VAV System Name
T1,                 !- Template Thermostat Name
0.775650326333887, !- Supply Air Maximum Flow Rate {m3/s}
,                   !- Zone Heating Sizing Factor
,                   !- Zone Cooling Sizing Factor
Scheduled,          !- Zone Minimum Air Flow Input Method
0.05,               !- Constant Minimum Air Flow Fraction
,                   !- Fixed Minimum Air Flow Rate {m3/s}
VAV_4,              !- Minimum Air Flow Fraction Schedule Name
Flow/Person,        !- Outdoor Air Method
0.00944,            !- Outdoor Air Flow Rate per Person {m3/s}
0,                  !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0,                  !- Outdoor Air Flow Rate per Zone {m3/s}
None,               !- Reheat Coil Type
,                   !- Reheat Coil Availability Schedule Name
Normal,             !- Damper Heating Action
,                   !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,                   !- Maximum Flow Fraction During Reheat
,                   !- Maximum Reheat Air Temperature {C}
,                   !- Design Specification Outdoor Air Object Name for Control
,                   !- Supply Plenum Name

```

```

,                                     !- Return Plenum Name
None,                                !- Baseboard Heating Type
,                                     !- Baseboard Heating Availability Schedule Name
autosize,                            !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature, !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7,                                !- Zone Cooling Design Supply Air Temperature {C}
11.11,                               !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature,               !- Zone Heating Design Supply Air Temperature Input Method
50,                                 !- Zone Heating Design Supply Air Temperature {C}
;                                   !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate:Zone:VAV,
Room5,                              !- Zone Name
Main AHU,                           !- Template VAV System Name
T1,                                  !- Template Thermostat Name
0.320264436783412,                 !- Supply Air Maximum Flow Rate {m3/s}
,                                    !- Zone Heating Sizing Factor
,                                    !- Zone Cooling Sizing Factor
Scheduled,                          !- Zone Minimum Air Flow Input Method
0.05,                                !- Constant Minimum Air Flow Fraction
,                                    !- Fixed Minimum Air Flow Rate {m3/s}
VAV_5,                              !- Minimum Air Flow Fraction Schedule Name
Flow/Person,                        !- Outdoor Air Method
0.00944,                            !- Outdoor Air Flow Rate per Person {m3/s}
0,                                  !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0,                                  !- Outdoor Air Flow Rate per Zone {m3/s}
None,                                !- Reheat Coil Type
,                                    !- Reheat Coil Availability Schedule Name
Normal,                              !- Damper Heating Action
,                                    !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,                                    !- Maximum Flow Fraction During Reheat
,                                    !- Maximum Reheat Air Temperature {C}
,                                    !- Design Specification Outdoor Air Object Name for Control
,                                    !- Supply Plenum Name
,                                    !- Return Plenum Name
None,                                !- Baseboard Heating Type
,                                    !- Baseboard Heating Availability Schedule Name
autosize,                            !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature, !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7,                                !- Zone Cooling Design Supply Air Temperature {C}
11.11,                               !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature,               !- Zone Heating Design Supply Air Temperature Input Method
50,                                 !- Zone Heating Design Supply Air Temperature {C}
;                                   !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate:Zone:VAV,
Room6,                              !- Zone Name
Main AHU,                           !- Template VAV System Name
T1,                                  !- Template Thermostat Name
0.429425348016034,                 !- Supply Air Maximum Flow Rate {m3/s}
,                                    !- Zone Heating Sizing Factor
,                                    !- Zone Cooling Sizing Factor
Scheduled,                          !- Zone Minimum Air Flow Input Method
0.05,                                !- Constant Minimum Air Flow Fraction
,                                    !- Fixed Minimum Air Flow Rate {m3/s}
VAV_6,                              !- Minimum Air Flow Fraction Schedule Name
Flow/Person,                        !- Outdoor Air Method
0.00944,                            !- Outdoor Air Flow Rate per Person {m3/s}

```

```

0,                !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0,                !- Outdoor Air Flow Rate per Zone {m3/s}
None,             !- Reheat Coil Type
,                !- Reheat Coil Availability Schedule Name
Normal,           !- Damper Heating Action
,                !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,                !- Maximum Flow Fraction During Reheat
,                !- Maximum Reheat Air Temperature {C}
,                !- Design Specification Outdoor Air Object Name for Control
,                !- Supply Plenum Name
,                !- Return Plenum Name
None,             !- Baseboard Heating Type
,                !- Baseboard Heating Availability Schedule Name
autosize,         !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature, !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7,             !- Zone Cooling Design Supply Air Temperature {C}
11.11,           !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature, !- Zone Heating Design Supply Air Temperature Input Method
50,              !- Zone Heating Design Supply Air Temperature {C}
;                !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate:Zone:VAV,
Room7,           !- Zone Name
Main AHU,        !- Template VAV System Name
T1,              !- Template Thermostat Name
0.519590741052724, !- Supply Air Maximum Flow Rate {m3/s}
,                !- Zone Heating Sizing Factor
,                !- Zone Cooling Sizing Factor
Scheduled,       !- Zone Minimum Air Flow Input Method
0.05,           !- Constant Minimum Air Flow Fraction
,                !- Fixed Minimum Air Flow Rate {m3/s}
VAV_7,          !- Minimum Air Flow Fraction Schedule Name
Flow/Person,    !- Outdoor Air Method
0.00944,        !- Outdoor Air Flow Rate per Person {m3/s}
0,              !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0,              !- Outdoor Air Flow Rate per Zone {m3/s}
None,           !- Reheat Coil Type
,               !- Reheat Coil Availability Schedule Name
Normal,         !- Damper Heating Action
,               !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,               !- Maximum Flow Fraction During Reheat
,               !- Maximum Reheat Air Temperature {C}
,               !- Design Specification Outdoor Air Object Name for Control
,               !- Supply Plenum Name
,               !- Return Plenum Name
None,           !- Baseboard Heating Type
,               !- Baseboard Heating Availability Schedule Name
autosize,       !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature, !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7,           !- Zone Cooling Design Supply Air Temperature {C}
11.11,         !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature, !- Zone Heating Design Supply Air Temperature Input Method
50,            !- Zone Heating Design Supply Air Temperature {C}
;              !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate:Zone:VAV,
Room8,          !- Zone Name
Main AHU,       !- Template VAV System Name

```

```

T1,                !- Template Thermostat Name
0.218511777647203, !- Supply Air Maximum Flow Rate {m3/s}
,                 !- Zone Heating Sizing Factor
,                 !- Zone Cooling Sizing Factor
Scheduled,        !- Zone Minimum Air Flow Input Method
0.05,            !- Constant Minimum Air Flow Fraction
,                 !- Fixed Minimum Air Flow Rate {m3/s}
VAV_8,           !- Minimum Air Flow Fraction Schedule Name
Flow/Person,     !- Outdoor Air Method
0.00944,         !- Outdoor Air Flow Rate per Person {m3/s}
0,               !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0,               !- Outdoor Air Flow Rate per Zone {m3/s}
None,            !- Reheat Coil Type
,                !- Reheat Coil Availability Schedule Name
Normal,          !- Damper Heating Action
,                !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,                !- Maximum Flow Fraction During Reheat
,                !- Maximum Reheat Air Temperature {C}
,                !- Design Specification Outdoor Air Object Name for Control
,                !- Supply Plenum Name
,                !- Return Plenum Name
None,            !- Baseboard Heating Type
,                !- Baseboard Heating Availability Schedule Name
autosize,        !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature, !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7,            !- Zone Cooling Design Supply Air Temperature {C}
11.11,           !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature, !- Zone Heating Design Supply Air Temperature Input Method
50,              !- Zone Heating Design Supply Air Temperature {C}
;                !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate:Zone:VAV,
Room9,           !- Zone Name
Main AHU,        !- Template VAV System Name
T1,              !- Template Thermostat Name
1.70959663763387, !- Supply Air Maximum Flow Rate {m3/s}
,                !- Zone Heating Sizing Factor
,                !- Zone Cooling Sizing Factor
Scheduled,       !- Zone Minimum Air Flow Input Method
0.05,           !- Constant Minimum Air Flow Fraction
,                !- Fixed Minimum Air Flow Rate {m3/s}
VAV_9,          !- Minimum Air Flow Fraction Schedule Name
Flow/Person,    !- Outdoor Air Method
0.00944,        !- Outdoor Air Flow Rate per Person {m3/s}
0,              !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0,              !- Outdoor Air Flow Rate per Zone {m3/s}
None,           !- Reheat Coil Type
,               !- Reheat Coil Availability Schedule Name
Normal,         !- Damper Heating Action
,               !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,               !- Maximum Flow Fraction During Reheat
,               !- Maximum Reheat Air Temperature {C}
,               !- Design Specification Outdoor Air Object Name for Control
,               !- Supply Plenum Name
,               !- Return Plenum Name
None,           !- Baseboard Heating Type
,               !- Baseboard Heating Availability Schedule Name
autosize,       !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature, !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7,           !- Zone Cooling Design Supply Air Temperature {C}

```

```

11.11,                !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature, !- Zone Heating Design Supply Air Temperature Input Method
50,                  !- Zone Heating Design Supply Air Temperature {C}
;                    !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate:Zone:VAV,
Room10,              !- Zone Name
Main AHU,            !- Template VAV System Name
T1,                  !- Template Thermostat Name
0.547007605648853,  !- Supply Air Maximum Flow Rate {m3/s}
,                    !- Zone Heating Sizing Factor
,                    !- Zone Cooling Sizing Factor
Scheduled,           !- Zone Minimum Air Flow Input Method
0.05,               !- Constant Minimum Air Flow Fraction
,                    !- Fixed Minimum Air Flow Rate {m3/s}
VAV_10,              !- Minimum Air Flow Fraction Schedule Name
Flow/Person,         !- Outdoor Air Method
0.00944,             !- Outdoor Air Flow Rate per Person {m3/s}
0,                   !- Outdoor Air Flow Rate per Zone Floor Area {m3/s-m2}
0,                   !- Outdoor Air Flow Rate per Zone {m3/s}
None,                !- Reheat Coil Type
,                    !- Reheat Coil Availability Schedule Name
Normal,              !- Damper Heating Action
,                    !- Maximum Flow per Zone Floor Area During Reheat {m3/s-m2}
,                    !- Maximum Flow Fraction During Reheat
,                    !- Maximum Reheat Air Temperature {C}
,                    !- Design Specification Outdoor Air Object Name for Control
,                    !- Supply Plenum Name
,                    !- Return Plenum Name
None,                !- Baseboard Heating Type
,                    !- Baseboard Heating Availability Schedule Name
autosize,            !- Baseboard Heating Capacity {W}
SystemSupplyAirTemperature, !- Zone Cooling Design Supply Air Temperature Input
    Method
12.7,                !- Zone Cooling Design Supply Air Temperature {C}
11.11,               !- Zone Cooling Design Supply Air Temperature Difference {
    deltaC }
SupplyAirTemperature, !- Zone Heating Design Supply Air Temperature Input Method
50,                  !- Zone Heating Design Supply Air Temperature {C}
;                    !- Zone Heating Design Supply Air Temperature Difference {
    deltaC }

HVACTemplate:Plant:ChilledWaterLoop,
Chiled Water Loop,   !- Name
,                    !- Pump Schedule Name
Intermittent,        !- Pump Control Type
Default,             !- Chiller Plant Operation Scheme Type
,                    !- Chiller Plant Equipment Operation Schemes Name
,                    !- Chilled Water Setpoint Schedule Name
7.22,                !- Chilled Water Design Setpoint {C}
ConstantPrimaryNoSecondary, !- Chilled Water Pump Configuration
179352,              !- Primary Chilled Water Pump Rated Head {Pa}
179352,              !- Secondary Chilled Water Pump Rated Head {Pa}
Default,             !- Condenser Plant Operation Scheme Type
,                    !- Condenser Equipment Operation Schemes Name
SpecifiedSetpoint,   !- Condenser Water Temperature Control Type
,                    !- Condenser Water Setpoint Schedule Name
29.4,                !- Condenser Water Design Setpoint {C}
179352,              !- Condenser Water Pump Rated Head {Pa}
None,                !- Chilled Water Setpoint Reset Type
12.2,                !- Chilled Water Setpoint at Outdoor Dry-Bulb Low {C}
15.6,                !- Chilled Water Reset Outdoor Dry-Bulb Low {C}

```

```

5.7,                !- Chilled Water Setpoint at Outdoor Dry-Bulb High {C}
26.7,              !- Chilled Water Reset Outdoor Dry-Bulb High {C}
,                  !- Chilled Water Primary Pump Type
,                  !- Chilled Water Secondary Pump Type
,                  !- Condenser Water Pump Type
,                  !- Chilled Water Supply Side Bypass Pipe
,                  !- Chilled Water Demand Side Bypass Pipe
,                  !- Condenser Water Supply Side Bypass Pipe
,                  !- Condenser Water Demand Side Bypass Pipe
,                  !- Fluid Type
,                  !- Loop Design Delta Temperature {deltaC}
7.22;              !- Minimum Outdoor Dry Bulb Temperature {C}

HVACTemplate:Plant:Chiller,
Main Chiller,      !- Name
ElectricReciprocatingChiller, !- Chiller Type
100000,            !- Capacity {W}
3.2,               !- Nominal COP {W/W}
WaterCooled,       !- Condenser Type
1,                 !- Priority
;                  !- Sizing Factor

Output:Variable,*,Site Outdoor Air Drybulb Temperature,Timestep;

Lights,
Zone 1 Lights,    !- Name
Room1,            !- Zone or ZoneList Name
Always On,        !- Schedule Name
Watts/Area,       !- Design Level Calculation Method
,                 !- Lighting Level {W}
3,                !- Watts per Zone Floor Area {W/m2}
,                 !- Watts per Person {W/person}
,                 !- Return Air Fraction
,                 !- Fraction Radiant
,                 !- Fraction Visible
1,                !- Fraction Replaceable
General,          !- End-Use Subcategory
No;               !- Return Air Fraction Calculated from Plenum Temperature

Lights,
Zone 2 Lights,    !- Name
Room2,            !- Zone or ZoneList Name
Always On,        !- Schedule Name
Watts/Area,       !- Design Level Calculation Method
,                 !- Lighting Level {W}
3,                !- Watts per Zone Floor Area {W/m2}
,                 !- Watts per Person {W/person}
,                 !- Return Air Fraction
,                 !- Fraction Radiant
,                 !- Fraction Visible
1,                !- Fraction Replaceable
General,          !- End-Use Subcategory
No;               !- Return Air Fraction Calculated from Plenum Temperature

Lights,
Zone 3 Lights,    !- Name
Room3,            !- Zone or ZoneList Name
Always On,        !- Schedule Name
Watts/Area,       !- Design Level Calculation Method
,                 !- Lighting Level {W}
3,                !- Watts per Zone Floor Area {W/m2}
,                 !- Watts per Person {W/person}
,                 !- Return Air Fraction
,                 !- Fraction Radiant

```

,	!- Fraction Visible
1,	!- Fraction Replaceable
General,	!- End-Use Subcategory
No;	!- Return Air Fraction Calculated from Plenum Temperature

Lights ,

Zone 4 Lights ,	!- Name
Room4,	!- Zone or ZoneList Name
Always On,	!- Schedule Name
Watts/Area ,	!- Design Level Calculation Method
,	!- Lighting Level {W}
3,	!- Watts per Zone Floor Area {W/m2}
,	!- Watts per Person {W/person}
,	!- Return Air Fraction
,	!- Fraction Radiant
,	!- Fraction Visible
1,	!- Fraction Replaceable
General,	!- End-Use Subcategory
No;	!- Return Air Fraction Calculated from Plenum Temperature

Lights ,

Zone 5 Lights ,	!- Name
Room5,	!- Zone or ZoneList Name
Always On,	!- Schedule Name
Watts/Area ,	!- Design Level Calculation Method
,	!- Lighting Level {W}
3,	!- Watts per Zone Floor Area {W/m2}
,	!- Watts per Person {W/person}
,	!- Return Air Fraction
,	!- Fraction Radiant
,	!- Fraction Visible
1,	!- Fraction Replaceable
General,	!- End-Use Subcategory
No;	!- Return Air Fraction Calculated from Plenum Temperature

Lights ,

Zone 6 Lights ,	!- Name
Room6,	!- Zone or ZoneList Name
Always On,	!- Schedule Name
Watts/Area ,	!- Design Level Calculation Method
,	!- Lighting Level {W}
3,	!- Watts per Zone Floor Area {W/m2}
,	!- Watts per Person {W/person}
,	!- Return Air Fraction
,	!- Fraction Radiant
,	!- Fraction Visible
1,	!- Fraction Replaceable
General,	!- End-Use Subcategory
No;	!- Return Air Fraction Calculated from Plenum Temperature

Lights ,

Zone 7 Lights ,	!- Name
Room7,	!- Zone or ZoneList Name
Always On,	!- Schedule Name
Watts/Area ,	!- Design Level Calculation Method
,	!- Lighting Level {W}
3,	!- Watts per Zone Floor Area {W/m2}
,	!- Watts per Person {W/person}
,	!- Return Air Fraction
,	!- Fraction Radiant
,	!- Fraction Visible
1,	!- Fraction Replaceable
General,	!- End-Use Subcategory
No;	!- Return Air Fraction Calculated from Plenum Temperature

```

Lights ,
  Zone 8 Lights ,
  Room8,
  Always On,
  Watts/Area ,
  ,
  3,
  ,
  ,
  ,
  1,
  General ,
  No;
  !- Name
  !- Zone or ZoneList Name
  !- Schedule Name
  !- Design Level Calculation Method
  !- Lighting Level {W}
  !- Watts per Zone Floor Area {W/m2}
  !- Watts per Person {W/person}
  !- Return Air Fraction
  !- Fraction Radiant
  !- Fraction Visible
  !- Fraction Replaceable
  !- End-Use Subcategory
  !- Return Air Fraction Calculated from Plenum Temperature

Lights ,
  Zone 9 Lights ,
  Room9,
  Always On,
  Watts/Area ,
  ,
  3,
  ,
  ,
  ,
  ,
  1,
  General ,
  No;
  !- Name
  !- Zone or ZoneList Name
  !- Schedule Name
  !- Design Level Calculation Method
  !- Lighting Level {W}
  !- Watts per Zone Floor Area {W/m2}
  !- Watts per Person {W/person}
  !- Return Air Fraction
  !- Fraction Radiant
  !- Fraction Visible
  !- Fraction Replaceable
  !- End-Use Subcategory
  !- Return Air Fraction Calculated from Plenum Temperature

Lights ,
  Zone 10 Lights ,
  Room10,
  Always On,
  Watts/Area ,
  ,
  3,
  ,
  ,
  ,
  ,
  1,
  General ,
  No;
  !- Name
  !- Zone or ZoneList Name
  !- Schedule Name
  !- Design Level Calculation Method
  !- Lighting Level {W}
  !- Watts per Zone Floor Area {W/m2}
  !- Watts per Person {W/person}
  !- Return Air Fraction
  !- Fraction Radiant
  !- Fraction Visible
  !- Fraction Replaceable
  !- End-Use Subcategory
  !- Return Air Fraction Calculated from Plenum Temperature

RunPeriod ,
  ,
  8,
  1,
  10,
  31,
  Tuesday ,
  Yes,
  Yes,
  No,
  Yes,
  Yes;
  !- Name
  !- Begin Month
  !- Begin Day of Month
  !- End Month
  !- End Day of Month
  !- Day of Week for Start Day
  !- Use Weather File Holidays and Special Days
  !- Use Weather File Daylight Saving Period
  !- Apply Weekend Holiday Rule
  !- Use Weather File Rain Indicators
  !- Use Weather File Snow Indicators

WindowMaterial:Glazing ,
  CLEAR 6MM,
  SpectralAverage ,
  ,
  0.006,
  !- Name
  !- Optical Data Type
  !- Window Glass Spectral Data Set Name
  !- Thickness {m}

```

```

0.775,      !- Solar Transmittance at Normal Incidence
0.071,      !- Front Side Solar Reflectance at Normal Incidence
0.071,      !- Back Side Solar Reflectance at Normal Incidence
0.881,      !- Visible Transmittance at Normal Incidence
0.080,      !- Front Side Visible Reflectance at Normal Incidence
0.080,      !- Back Side Visible Reflectance at Normal Incidence
0.0,        !- Infrared Transmittance at Normal Incidence
0.84,      !- Front Side Infrared Hemispherical Emissivity
0.84,      !- Back Side Infrared Hemispherical Emissivity
0.9;        !- Conductivity {W/m-K}

WindowMaterial:Gas,
  AIR 3MM,   !- Name
  Air,       !- Gas Type
  0.0032;    !- Thickness {m}

Version,8.1;

SimulationControl,
  Yes,       !- Do Zone Sizing Calculation
  Yes,       !- Do System Sizing Calculation
  Yes,       !- Do Plant Sizing Calculation
  No,        !- Run Simulation for Sizing Periods
  Yes;       !- Run Simulation for Weather File Run Periods

Building,
  Untitled,  !- Name
  30,        !- North Axis {deg}
  City,      !- Terrain
  0.04,      !- Loads Convergence Tolerance Value
  0.4,       !- Temperature Convergence Tolerance Value {deltaC}
  FullExterior, !- Solar Distribution
  25,        !- Maximum Number of Warmup Days
  6;         !- Minimum Number of Warmup Days

Timestep,60;

Site:Location,
  College Station Easterwood Fl TX USA TMY3 WMO#=722445, !- Name
  30.62,      !- Latitude {deg}
  -96.34,     !- Longitude {deg}
  -6.0,       !- Time Zone {hr}
  103;        !- Elevation {m}

ScheduleTypeLimits,
  Any Number; !- Name

ScheduleTypeLimits,
  Fraction,   !- Name
  0.0,        !- Lower Limit Value
  1.0,        !- Upper Limit Value
  CONTINUOUS; !- Numeric Type

ScheduleTypeLimits,
  Temperature, !- Name
  -60,         !- Lower Limit Value
  200,         !- Upper Limit Value
  CONTINUOUS,  !- Numeric Type
  Temperature; !- Unit Type

ScheduleTypeLimits,
  On/Off,     !- Name
  0,          !- Lower Limit Value
  1,          !- Upper Limit Value

```

```

DISCRETE;                                !- Numeric Type

ScheduleTypeLimits ,
  Control Type ,                          !- Name
  0,                                       !- Lower Limit Value
  4,                                       !- Upper Limit Value
  DISCRETE;                               !- Numeric Type

ScheduleTypeLimits ,
  Humidity ,                             !- Name
  10,                                     !- Lower Limit Value
  90,                                     !- Upper Limit Value
  CONTINUOUS;                             !- Numeric Type

ScheduleTypeLimits ,
  FlowRate ,                             !- Name
  0,                                       !- Lower Limit Value
  10,                                     !- Upper Limit Value
  Continuous;                             !- Numeric Type

Schedule:Compact,
  Infiltration Schedule ,               !- Name
  Fraction ,                            !- Schedule Type Limits Name
  Through: 12/31,                        !- Field 1
  For: Weekdays SummerDesignDay,       !- Field 2
  Until: 06:00, 1.0,                     !- Field 4
  Until: 22:00, 1,                       !- Field 6
  Until: 24:00, 1.0,                     !- Field 8
  For: Saturday WinterDesignDay,         !- Field 9
  Until: 06:00, 1.0,                     !- Field 11
  Until: 18:00, 1,                       !- Field 13
  Until: 24:00, 1.0,                     !- Field 15
  For: Sunday Holidays AllOtherDays,     !- Field 16
  Until: 24:00, 1.0;                     !- Field 18

Schedule:Compact,
  Always On,                             !- Name
  Fraction ,                            !- Schedule Type Limits Name
  Through: 12/31,                        !- Field 1
  For: AllDays,                          !- Field 2
  Until: 24:00, 1.0;                     !- Field 4

Schedule:Compact,
  Always Off,                            !- Name
  Fraction ,                            !- Schedule Type Limits Name
  Through: 12/31,                        !- Field 1
  For: AllDays,                          !- Field 2
  Until: 24:00, 0.0;                     !- Field 4

Material ,
  F08 Metal surface ,                   !- Name
  Smooth,                               !- Roughness
  0.0008,                               !- Thickness {m}
  45.28,                                !- Conductivity {W/m-K}
  7824,                                  !- Density {kg/m3}
  500;                                   !- Specific Heat {J/kg-K}

Material ,
  I01 25mm insulation board ,           !- Name
  MediumRough,                          !- Roughness
  0.0254,                                !- Thickness {m}
  0.03,                                  !- Conductivity {W/m-K}
  43,                                    !- Density {kg/m3}
  1210;                                  !- Specific Heat {J/kg-K}

```

Material ,
 I02 50mm insulation board , !- Name
 MediumRough, !- Roughness
 0.0508, !- Thickness {m}
 0.03, !- Conductivity {W/m-K}
 43, !- Density {kg/m3}
 1210; !- Specific Heat {J/kg-K}

Material ,
 G01a 19mm gypsum board , !- Name
 MediumSmooth, !- Roughness
 0.019, !- Thickness {m}
 0.16, !- Conductivity {W/m-K}
 800, !- Density {kg/m3}
 1500; !- Specific Heat {J/kg-K}

Material ,
 M11 100mm lightweight concrete , !- Name
 MediumRough, !- Roughness
 0.1016, !- Thickness {m}
 0.53, !- Conductivity {W/m-K}
 1280, !- Density {kg/m3}
 840; !- Specific Heat {J/kg-K}

Material ,
 F16 Acoustic tile , !- Name
 MediumSmooth, !- Roughness
 0.0191, !- Thickness {m}
 0.06, !- Conductivity {W/m-K}
 368, !- Density {kg/m3}
 590; !- Specific Heat {J/kg-K}

Material ,
 M01 100mm brick , !- Name
 MediumRough, !- Roughness
 0.1016, !- Thickness {m}
 0.89, !- Conductivity {W/m-K}
 1920, !- Density {kg/m3}
 790; !- Specific Heat {J/kg-K}

Material ,
 M15 200mm heavyweight concrete , !- Name
 MediumRough, !- Roughness
 0.2032, !- Thickness {m}
 1.95, !- Conductivity {W/m-K}
 2240, !- Density {kg/m3}
 900; !- Specific Heat {J/kg-K}

Material ,
 M05 200mm concrete block, !- Name
 MediumRough, !- Roughness
 0.2032, !- Thickness {m}
 1.11, !- Conductivity {W/m-K}
 800, !- Density {kg/m3}
 920; !- Specific Heat {J/kg-K}

Material ,
 G05 25mm wood, !- Name
 MediumSmooth, !- Roughness
 0.0254, !- Thickness {m}
 0.15, !- Conductivity {W/m-K}
 608, !- Density {kg/m3}
 1630; !- Specific Heat {J/kg-K}

```

Material:AirGap,
    F04 Wall air space resistance, !- Name
    0.15;                          !- Thermal Resistance {m2-K/W}

Material:AirGap,
    F05 Ceiling air space resistance, !- Name
    0.18;                          !- Thermal Resistance {m2-K/W}

Construction,
    Exterior Floor,                !- Name
    I02 50mm insulation board, !- Outside Layer
    M15 200mm heavyweight concrete; !- Layer 2

Construction,
    Interior Floor,                !- Name
    F16 Acoustic tile,            !- Outside Layer
    F05 Ceiling air space resistance, !- Layer 2
    M11 100mm lightweight concrete; !- Layer 3

Construction,
    Exterior Wall,                !- Name
    M01 100mm brick,              !- Outside Layer
    M15 200mm heavyweight concrete, !- Layer 2
    I02 50mm insulation board, !- Layer 3
    F04 Wall air space resistance, !- Layer 4
    G01a 19mm gypsum board; !- Layer 5

Construction,
    Interior Wall,                !- Name
    G01b 19mm gypsum board; !- Outside Layer

Construction,
    Exterior Roof,                !- Name
    M11 100mm lightweight concrete, !- Outside Layer
    F05 Ceiling air space resistance, !- Layer 2
    F16 Acoustic tile;            !- Layer 3

Construction,
    Interior Ceiling,              !- Name
    M11 100mm lightweight concrete, !- Outside Layer
    F05 Ceiling air space resistance, !- Layer 2
    F16 Acoustic tile;            !- Layer 3

Construction,
    Exterior Window,              !- Name
    CLEAR 6MM,                    !- Outside Layer
    AIR 3MM,                      !- Layer 2
    CLEAR 6MM;                    !- Layer 3

Construction,
    Interior Window,              !- Name
    CLEAR 6MM;                    !- Outside Layer

Construction,
    Exterior Door,                !- Name
    F08 Metal surface,            !- Outside Layer
    I01 25mm insulation board; !- Layer 2

Construction,
    Interior Door,                !- Name
    G05 25mm wood;                !- Outside Layer

GlobalGeometryRules,

```

```

UpperLeftCorner ,      !- Starting Vertex Position
Counterclockwise ,    !- Vertex Entry Direction
Relative ;             !- Coordinate System

Zone ,
Room4 ,                !- Name
0.0 ,                  !- Direction of Relative North {deg}
0.0, 0.0, 0.0 ,        !- X,Y,Z   {m}
1 ,                    !- Type
1 ,                    !- Multiplier
2.81 ;                 !- Ceiling Height {m}

Zone ,
Room5 ,                !- Name
0.0 ,                  !- Direction of Relative North {deg}
5.225281, 0.163972, 0.0 ,    !- X,Y,Z   {m}
1 ,                    !- Type
1 ,                    !- Multiplier
2.81 ;                 !- Ceiling Height {m}

Zone ,
Room6 ,                !- Name
0.0 ,                  !- Direction of Relative North {deg}
8.725281, 0.163972, 0.0 ,    !- X,Y,Z   {m}
1 ,                    !- Type
1 ,                    !- Multiplier
2.81 ;                 !- Ceiling Height {m}

Zone ,
Room7 ,                !- Name
0.0 ,                  !- Direction of Relative North {deg}
14.085281, 0.163972, 0.0 ,    !- X,Y,Z   {m}
1 ,                    !- Type
1 ,                    !- Multiplier
2.81 ;                 !- Ceiling Height {m}

Zone ,
Room10 ,               !- Name
0.0 ,                  !- Direction of Relative North {deg}
19.915281, 0.163972, 0.0 ,    !- X,Y,Z   {m}
1 ,                    !- Type
1 ,                    !- Multiplier
2.81 ;                 !- Ceiling Height {m}

Zone ,
Room9 ,                !- Name
0.0 ,                  !- Direction of Relative North {deg}
14.301345, 4.973972, 0.0 ,    !- X,Y,Z   {m}
1 ,                    !- Type
1 ,                    !- Multiplier
2.81 ;                 !- Ceiling Height {m}

Zone ,
Room8 ,                !- Name
0.0 ,                  !- Direction of Relative North {deg}
10.189113, 8.632842, 0.0 ,    !- X,Y,Z   {m}
1 ,                    !- Type
1 ,                    !- Multiplier
2.81 ;                 !- Ceiling Height {m}

Zone ,
Room8A ,               !- Name
0.0 ,                  !- Direction of Relative North {deg}
12.20384, 8.641922, 0.0 ,    !- X,Y,Z   {m}

```

```

1,                !- Type
1,                !- Multiplier
2.81;             !- Ceiling Height {m}

Zone,
Room8B,           !- Name
0.0,              !- Direction of Relative North {deg}
10.146699, 7.209293, 0.0,      !- X,Y,Z {m}
1,                !- Type
1,                !- Multiplier
2.81;             !- Ceiling Height {m}

Zone,
Room2,            !- Name
0.0,              !- Direction of Relative North {deg}
5.447727, 14.528834, 0.0,     !- X,Y,Z {m}
1,                !- Type
1,                !- Multiplier
2.81;             !- Ceiling Height {m}

Zone,
Room3,            !- Name
0.0,              !- Direction of Relative North {deg}
0.165281, 4.973972, 0.0,     !- X,Y,Z {m}
1,                !- Type
1,                !- Multiplier
2.81;             !- Ceiling Height {m}

Zone,
Room1,            !- Name
0.0,              !- Direction of Relative North {deg}
0.0, 17.163348, 0.0,         !- X,Y,Z {m}
1,                !- Type
1,                !- Multiplier
2.81;             !- Ceiling Height {m}

Zone,
Room11,           !- Name
0.0,              !- Direction of Relative North {deg}
18.245281, 0.163972, 0.0,    !- X,Y,Z {m}
1,                !- Type
1,                !- Multiplier
2.81;             !- Ceiling Height {m}

BuildingSurface:Detailed,
6A1B56,           !- Name
Floor,            !- Surface Type
Exterior Floor,   !- Construction Name
Room4,            !- Zone Name
Ground,           !- Outside Boundary Condition
,                 !- Outside Boundary Condition Object
NoSun,            !- Sun Exposure
NoWind,           !- Wind Exposure
,                 !- View Factor to Ground
4,                !- Number of Vertices
5.225280670593, 4.973972273697, 0.000000000000,
                                     !- X,Y,Z 1 {m}
5.225280670593, 0.163972273697, 0.000000000000,
                                     !- X,Y,Z 2 {m}
0.165280670593, 0.163972273697, 0.000000000000,
                                     !- X,Y,Z 3 {m}
0.165280670593, 4.973972273697, 0.000000000000;
                                     !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
446665,                !- Name
Wall,                  !- Surface Type
Exterior Wall,         !- Construction Name
Room4,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
0.165280670593, 4.973972273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
0.165280670593, 4.973972273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
0.165280670593, 0.163972273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
0.165280670593, 0.163972273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
7991EC,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room4,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room1,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
5.225280670593, 4.973972273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
5.225280670593, 4.973972273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
3.745280670593, 4.973972273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
3.745280670593, 4.973972273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
4933D6,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room4,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room5,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
5.225280670593, 0.163972273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
5.225280670593, 0.163972273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
5.225280670593, 4.973972273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
5.225280670593, 4.973972273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
9293BE,                !- Name
Roof,                  !- Surface Type
Exterior Roof,         !- Construction Name

```

```

Room4,                !- Zone Name
Outdoors ,            !- Outside Boundary Condition
,                    !- Outside Boundary Condition Object
SunExposed ,          !- Sun Exposure
WindExposed ,         !- Wind Exposure
,                    !- View Factor to Ground
4,                    !- Number of Vertices
0.165280670593, 4.973972273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
0.165280670593, 0.163972273697, 2.810000000000,
                        !- X,Y,Z 2 {m}
5.225280670593, 0.163972273697, 2.810000000000,
                        !- X,Y,Z 3 {m}
5.225280670593, 4.973972273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed ,
FD5687,                !- Name
Wall,                  !- Surface Type
Exterior Wall ,        !- Construction Name
Room4,                 !- Zone Name
Outdoors ,             !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed ,           !- Sun Exposure
WindExposed ,          !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
0.165280670593, 0.163972273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
0.165280670593, 0.163972273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
5.225280670593, 0.163972273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
5.225280670593, 0.163972273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed ,
AC66D5,                !- Name
Floor ,               !- Surface Type
Exterior Floor ,      !- Construction Name
Room5,                 !- Zone Name
Ground ,              !- Outside Boundary Condition
,                     !- Outside Boundary Condition Object
NoSun ,               !- Sun Exposure
NoWind ,              !- Wind Exposure
,                     !- View Factor to Ground
4,                     !- Number of Vertices
3.499999670593, 4.810000273697, 0.000000000000,
                        !- X,Y,Z 1 {m}
3.499999670593, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.000000329407, 4.810000273697, 0.000000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed ,
864EDD,                !- Name
Wall,                  !- Surface Type
Interior Wall ,        !- Construction Name
Room5,                 !- Zone Name
Zone ,                !- Outside Boundary Condition
Room1,                 !- Outside Boundary Condition Object
NoSun ,               !- Sun Exposure

```

```

NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
3.499999670593, 4.810000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
3.499999670593, 4.810000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.000000329407, 4.810000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
567236,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room5,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room4,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
-0.000000329407, 4.810000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.000000329407, 0.000000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
ECC2C5,                !- Name
Wall,                  !- Surface Type
Exterior Wall,         !- Construction Name
Room5,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
-0.000000329407, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
3.499999670593, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
3.499999670593, 0.000000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
1329FD,                !- Name
Roof,                  !- Surface Type
Exterior Roof,         !- Construction Name
Room5,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
-0.000000329407, 4.810000273697, 2.810000000000,

```

```

!- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 2.810000000000,
!- X,Y,Z 2 {m}
3.499999670593, 0.000000273697, 2.810000000000,
!- X,Y,Z 3 {m}
3.499999670593, 4.810000273697, 2.810000000000;
!- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

8A2D5A,           !- Name
Wall,             !- Surface Type
Interior Wall,    !- Construction Name
Room5,            !- Zone Name
Zone,             !- Outside Boundary Condition
Room6,            !- Outside Boundary Condition Object
NoSun,            !- Sun Exposure
NoWind,           !- Wind Exposure
0.0,              !- View Factor to Ground
4,                !- Number of Vertices
3.499999670593, 0.000000273697, 2.810000000000,
!- X,Y,Z 1 {m}
3.499999670593, 0.000000273697, 0.000000000000,
!- X,Y,Z 2 {m}
3.499999670593, 4.810000273697, 0.000000000000,
!- X,Y,Z 3 {m}
3.499999670593, 4.810000273697, 2.810000000000;
!- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

5CAB05,           !- Name
Floor,            !- Surface Type
Exterior Floor,   !- Construction Name
Room6,            !- Zone Name
Ground,           !- Outside Boundary Condition
,                 !- Outside Boundary Condition Object
NoSun,            !- Sun Exposure
NoWind,           !- Wind Exposure
,                 !- View Factor to Ground
4,                !- Number of Vertices
5.359999670593, 4.810000273697, 0.000000000000,
!- X,Y,Z 1 {m}
5.359999670593, 0.000000273697, 0.000000000000,
!- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
!- X,Y,Z 3 {m}
-0.000000329407, 4.810000273697, 0.000000000000;
!- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

8DEB6D,           !- Name
Roof,             !- Surface Type
Exterior Roof,    !- Construction Name
Room6,            !- Zone Name
Outdoors,         !- Outside Boundary Condition
,                 !- Outside Boundary Condition Object
SunExposed,       !- Sun Exposure
WindExposed,      !- Wind Exposure
,                 !- View Factor to Ground
4,                !- Number of Vertices
-0.000000329407, 4.810000273697, 2.810000000000,
!- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 2.810000000000,
!- X,Y,Z 2 {m}
5.359999670593, 0.000000273697, 2.810000000000,

```

```

                                !- X,Y,Z 3 {m}
5.359999670593, 4.810000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
4238E0,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room6,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room7,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                              !- Number of Vertices
5.359999670593, 0.000000273697, 2.810000000000,
                                !- X,Y,Z 1 {m}
5.359999670593, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
5.359999670593, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
5.359999670593, 4.810000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
BF9A90,                        !- Name
Wall,                          !- Surface Type
Exterior Wall,                !- Construction Name
Room6,                         !- Zone Name
Outdoors,                     !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                   !- Sun Exposure
WindExposed,                  !- Wind Exposure
,                               !- View Factor to Ground
4,                              !- Number of Vertices
-0.000000329407, 0.000000273697, 2.810000000000,
                                !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
5.359999670593, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
5.359999670593, 0.000000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
4AFBB9,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room6,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room1,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                              !- Number of Vertices
5.359999670593, 4.810000273697, 2.810000000000,
                                !- X,Y,Z 1 {m}
5.359999670593, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
-0.000000329407, 4.810000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
4B0702,           !- Name
Wall,            !- Surface Type
Interior Wall,   !- Construction Name
Room6,          !- Zone Name
Zone,           !- Outside Boundary Condition
Room5,          !- Outside Boundary Condition Object
NoSun,          !- Sun Exposure
NoWind,         !- Wind Exposure
0.0,            !- View Factor to Ground
4,              !- Number of Vertices
-0.000000329407, 4.810000273697, 2.810000000000,
                !- X,Y,Z 1 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                !- X,Y,Z 3 {m}
-0.000000329407, 0.000000273697, 2.810000000000;
                !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
E5194D,          !- Name
Floor,           !- Surface Type
Exterior Floor,  !- Construction Name
Room7,          !- Zone Name
Ground,         !- Outside Boundary Condition
,               !- Outside Boundary Condition Object
NoSun,          !- Sun Exposure
NoWind,         !- Wind Exposure
,               !- View Factor to Ground
4,              !- Number of Vertices
4.159999670593, 4.810000273697, 0.000000000000,
                !- X,Y,Z 1 {m}
4.159999670593, 0.000000273697, 0.000000000000,
                !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                !- X,Y,Z 3 {m}
-0.000000329407, 4.810000273697, 0.000000000000;
                !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
ED158F,          !- Name
Wall,           !- Surface Type
Interior Wall,   !- Construction Name
Room7,          !- Zone Name
Zone,           !- Outside Boundary Condition
Room11,         !- Outside Boundary Condition Object
NoSun,          !- Sun Exposure
NoWind,         !- Wind Exposure
0.0,            !- View Factor to Ground
4,              !- Number of Vertices
4.159999670593, 0.000000273697, 2.810000000000,
                !- X,Y,Z 1 {m}
4.159999670593, 0.000000273697, 0.000000000000,
                !- X,Y,Z 2 {m}
4.159999670593, 4.810000273697, 0.000000000000,
                !- X,Y,Z 3 {m}
4.159999670593, 4.810000273697, 2.810000000000;
                !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
9B7370,          !- Name
Wall,           !- Surface Type
Interior Wall,   !- Construction Name

```

```

Room7,                !- Zone Name
Zone,                 !- Outside Boundary Condition
Room6,                !- Outside Boundary Condition Object
NoSun,                !- Sun Exposure
NoWind,               !- Wind Exposure
0.0,                  !- View Factor to Ground
4,                    !- Number of Vertices
-0.000000329407, 4.810000273697, 2.810000000000,
                    !- X,Y,Z 1 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                    !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                    !- X,Y,Z 3 {m}
-0.000000329407, 0.000000273697, 2.810000000000;
                    !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
F85C1E,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room7,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room9,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
4.159999670593, 4.810000273697, 2.810000000000,
                    !- X,Y,Z 1 {m}
4.159999670593, 4.810000273697, 0.000000000000,
                    !- X,Y,Z 2 {m}
0.189999670593, 4.810000273697, 0.000000000000,
                    !- X,Y,Z 3 {m}
0.189999670593, 4.810000273697, 2.810000000000;
                    !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
653AE9,                !- Name
Wall,                  !- Surface Type
Exterior Wall,         !- Construction Name
Room7,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
-0.000000329407, 0.000000273697, 2.810000000000,
                    !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                    !- X,Y,Z 2 {m}
4.159999670593, 0.000000273697, 0.000000000000,
                    !- X,Y,Z 3 {m}
4.159999670593, 0.000000273697, 2.810000000000;
                    !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
86661E,                !- Name
Roof,                  !- Surface Type
Exterior Roof,         !- Construction Name
Room7,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure

```

```

WindExposed,           !- Wind Exposure
,                       !- View Factor to Ground
4,                       !- Number of Vertices
-0.000000329407, 4.810000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 2 {m}
4.159999670593, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 3 {m}
4.159999670593, 4.810000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
98A6BC,                 !- Name
Floor,                 !- Surface Type
Exterior Floor,        !- Construction Name
Room10,                !- Zone Name
Ground,                !- Outside Boundary Condition
,                       !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
,                       !- View Factor to Ground
4,                       !- Number of Vertices
6.234719000000, 16.822887731968, 0.000000000000,
                        !- X,Y,Z 1 {m}
6.234719000000, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.000000329407, 16.822887731968, 0.000000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
B8827C,                 !- Name
Wall,                 !- Surface Type
Interior Wall,         !- Construction Name
Room10,                !- Zone Name
Zone,                 !- Outside Boundary Condition
Room9,                !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                       !- Number of Vertices
-0.000000329407, 16.822887731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.000000329407, 16.822887731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.000000329407, 4.810000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
E908B6,                 !- Name
Wall,                 !- Surface Type
Exterior Wall,         !- Construction Name
Room10,                !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                       !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                       !- View Factor to Ground
4,                       !- Number of Vertices
-0.000000329407, 0.000000273697, 2.810000000000,

```

```

                                !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
6.234719000000, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
6.234719000000, 0.000000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
8A40EE,                        !- Name
Wall,                          !- Surface Type
Exterior Wall,                 !- Construction Name
Room10,                        !- Zone Name
Outdoors,                      !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                    !- Sun Exposure
WindExposed,                   !- Wind Exposure
,                               !- View Factor to Ground
4,                              !- Number of Vertices
6.234719000000, 16.822887731968, 2.810000000000,
                                !- X,Y,Z 1 {m}
6.234719000000, 16.822887731968, 0.000000000000,
                                !- X,Y,Z 2 {m}
-0.000000329407, 16.822887731968, 0.000000000000,
                                !- X,Y,Z 3 {m}
-0.000000329407, 16.822887731968, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
2FACA9,                        !- Name
Wall,                          !- Surface Type
Exterior Wall,                 !- Construction Name
Room10,                        !- Zone Name
Outdoors,                      !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                    !- Sun Exposure
WindExposed,                   !- Wind Exposure
,                               !- View Factor to Ground
4,                              !- Number of Vertices
6.234719000000, 0.000000273697, 2.810000000000,
                                !- X,Y,Z 1 {m}
6.234719000000, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
6.234719000000, 16.822887731968, 0.000000000000,
                                !- X,Y,Z 3 {m}
6.234719000000, 16.822887731968, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
3C8082,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room10,                        !- Zone Name
Zone,                          !- Outside Boundary Condition
Room11,                        !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                              !- Number of Vertices
-0.000000329407, 4.810000273697, 2.810000000000,
                                !- X,Y,Z 1 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,

```

```

                                !- X,Y,Z 3 {m}
-0.000000329407, 0.000000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
9EE35A,                        !- Name
Roof,                          !- Surface Type
Exterior Roof,                 !- Construction Name
Room10,                        !- Zone Name
Outdoors,                      !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                    !- Sun Exposure
WindExposed,                   !- Wind Exposure
,                               !- View Factor to Ground
4,                             !- Number of Vertices
-0.000000329407, 16.822887731968, 2.810000000000,
                                !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 2.810000000000,
                                !- X,Y,Z 2 {m}
6.234719000000, 0.000000273697, 2.810000000000,
                                !- X,Y,Z 3 {m}
6.234719000000, 16.822887731968, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
8BDA76,                        !- Name
Floor,                         !- Surface Type
Exterior Floor,                !- Construction Name
Room9,                         !- Zone Name
Ground,                        !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
,                               !- View Factor to Ground
4,                             !- Number of Vertices
5.613935670593, 12.012887731968, 0.000000000000,
                                !- X,Y,Z 1 {m}
5.613935670593, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
-0.026064329407, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
-0.026064329407, 12.012887731968, 0.000000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
DD801D,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room9,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room8,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                             !- Number of Vertices
-0.026064329407, 12.012887731968, 2.810000000000,
                                !- X,Y,Z 1 {m}
-0.026064329407, 12.012887731968, 0.000000000000,
                                !- X,Y,Z 2 {m}
-0.026064329407, 3.692887731968, 0.000000000000,
                                !- X,Y,Z 3 {m}
-0.026064329407, 3.692887731968, 2.810000000000;
                                !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
3AD07D,           !- Name
Wall,             !- Surface Type
Interior Wall,    !- Construction Name
Room9,            !- Zone Name
Zone,             !- Outside Boundary Condition
Room7,            !- Outside Boundary Condition Object
NoSun,            !- Sun Exposure
NoWind,           !- Wind Exposure
0.0,              !- View Factor to Ground
4,                !- Number of Vertices
-0.026064329407, 0.000000273697, 2.810000000000,
                  !- X,Y,Z 1 {m}
-0.026064329407, 0.000000273697, 0.000000000000,
                  !- X,Y,Z 2 {m}
3.943935670593, 0.000000273697, 0.000000000000,
                  !- X,Y,Z 3 {m}
3.943935670593, 0.000000273697, 2.810000000000;
                  !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
9F916F,           !- Name
Wall,             !- Surface Type
Exterior Wall,    !- Construction Name
Room9,            !- Zone Name
Outdoors,         !- Outside Boundary Condition
,                 !- Outside Boundary Condition Object
SunExposed,       !- Sun Exposure
WindExposed,      !- Wind Exposure
,                 !- View Factor to Ground
4,                !- Number of Vertices
5.613935670593, 12.012887731968, 2.810000000000,
                  !- X,Y,Z 1 {m}
5.613935670593, 12.012887731968, 0.000000000000,
                  !- X,Y,Z 2 {m}
-0.026064329407, 12.012887731968, 0.000000000000,
                  !- X,Y,Z 3 {m}
-0.026064329407, 12.012887731968, 2.810000000000;
                  !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
2A5B72,           !- Name
Wall,             !- Surface Type
Exterior Wall,    !- Construction Name
Room9,            !- Zone Name
Outdoors,         !- Outside Boundary Condition
,                 !- Outside Boundary Condition Object
SunExposed,       !- Sun Exposure
WindExposed,      !- Wind Exposure
,                 !- View Factor to Ground
4,                !- Number of Vertices
5.613935670593, 0.000000273697, 2.810000000000,
                  !- X,Y,Z 1 {m}
5.613935670593, 0.000000273697, 0.000000000000,
                  !- X,Y,Z 2 {m}
5.613935670593, 12.012887731968, 0.000000000000,
                  !- X,Y,Z 3 {m}
5.613935670593, 12.012887731968, 2.810000000000;
                  !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
7EF784,           !- Name
Roof,             !- Surface Type
Exterior Roof,    !- Construction Name

```

```

Room9,                !- Zone Name
Outdoors,             !- Outside Boundary Condition
,                     !- Outside Boundary Condition Object
SunExposed,           !- Sun Exposure
WindExposed,          !- Wind Exposure
,                     !- View Factor to Ground
4,                    !- Number of Vertices
-0.026064329407, 12.012887731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.026064329407, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 2 {m}
5.613935670593, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 3 {m}
5.613935670593, 12.012887731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
8B96E4,               !- Name
Floor,               !- Surface Type
Exterior Floor,      !- Construction Name
Room8,               !- Zone Name
Ground,              !- Outside Boundary Condition
,                    !- Outside Boundary Condition Object
NoSun,               !- Sun Exposure
NoWind,              !- Wind Exposure
,                    !- View Factor to Ground
4,                   !- Number of Vertices
4.086167670593, 8.354017731968, 0.000000000000,
                        !- X,Y,Z 1 {m}
4.086167670593, 0.034017731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.043832329407, 0.034017731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.043832329407, 8.354017731968, 0.000000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
8F0729,              !- Name
Wall,                !- Surface Type
Interior Wall,       !- Construction Name
Room8,               !- Zone Name
Zone,                !- Outside Boundary Condition
Room9,               !- Outside Boundary Condition Object
NoSun,               !- Sun Exposure
NoWind,              !- Wind Exposure
0.0,                 !- View Factor to Ground
4,                   !- Number of Vertices
4.086167670593, 0.034017731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
4.086167670593, 0.034017731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
4.086167670593, 8.354017731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
4.086167670593, 8.354017731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
C063C1,              !- Name
Wall,                !- Surface Type
Interior Wall,       !- Construction Name
Room8,               !- Zone Name
Zone,                !- Outside Boundary Condition
Room8B,              !- Outside Boundary Condition Object
NoSun,               !- Sun Exposure

```

```

NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
-0.043832329407, 0.034017731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.043832329407, 0.034017731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
2.021167670593, 0.034017731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
2.021167670593, 0.034017731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
65ECE3,                !- Name
Wall,                  !- Surface Type
Exterior Wall,         !- Construction Name
Room8,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
4.086167670593, 8.354017731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
4.086167670593, 8.354017731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.043832329407, 8.354017731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.043832329407, 8.354017731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
11516B,                !- Name
Roof,                  !- Surface Type
Exterior Roof,         !- Construction Name
Room8,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
-0.043832329407, 8.354017731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.043832329407, 0.034017731968, 2.810000000000,
                        !- X,Y,Z 2 {m}
4.086167670593, 0.034017731968, 2.810000000000,
                        !- X,Y,Z 3 {m}
4.086167670593, 8.354017731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
317104,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room8,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room1,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
-0.043832329407, 8.354017731968, 2.810000000000,

```

```

!- X,Y,Z 1 {m}
-0.043832329407, 8.354017731968, 0.000000000000,
!- X,Y,Z 2 {m}
-0.043832329407, 0.034017731968, 0.000000000000,
!- X,Y,Z 3 {m}
-0.043832329407, 0.034017731968, 2.810000000000;
!- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

BEA260,           !- Name
Floor,           !- Surface Type
Exterior Floor,  !- Construction Name
Room8A,          !- Zone Name
Ground,          !- Outside Boundary Condition
,               !- Outside Boundary Condition Object
NoSun,           !- Sun Exposure
NoWind,          !- Wind Exposure
,               !- View Factor to Ground
4,              !- Number of Vertices
2.071440670593, 0.024937731968, 0.000000000000,
!- X,Y,Z 1 {m}
2.071440670593, -1.425062268032, 0.000000000000,
!- X,Y,Z 2 {m}
0.006440670593, -1.425062268032, 0.000000000000,
!- X,Y,Z 3 {m}
0.006440670593, 0.024937731968, 0.000000000000;
!- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

ADA4D1,          !- Name
Wall,            !- Surface Type
Interior Wall,   !- Construction Name
Room8A,          !- Zone Name
Zone,            !- Outside Boundary Condition
Room1,           !- Outside Boundary Condition Object
NoSun,           !- Sun Exposure
NoWind,          !- Wind Exposure
0.0,             !- View Factor to Ground
4,              !- Number of Vertices
0.006440670593, -1.425062268032, 2.810000000000,
!- X,Y,Z 1 {m}
0.006440670593, -1.425062268032, 0.000000000000,
!- X,Y,Z 2 {m}
2.071440670593, -1.425062268032, 0.000000000000,
!- X,Y,Z 3 {m}
2.071440670593, -1.425062268032, 2.810000000000;
!- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

B975A8,          !- Name
Wall,            !- Surface Type
Interior Wall,   !- Construction Name
Room8A,          !- Zone Name
Zone,            !- Outside Boundary Condition
Room8B,          !- Outside Boundary Condition Object
NoSun,           !- Sun Exposure
NoWind,          !- Wind Exposure
0.0,             !- View Factor to Ground
4,              !- Number of Vertices
0.006440670593, 0.024937731968, 2.810000000000,
!- X,Y,Z 1 {m}
0.006440670593, 0.024937731968, 0.000000000000,
!- X,Y,Z 2 {m}
0.006440670593, -1.425062268032, 0.000000000000,

```

```

                                !- X,Y,Z 3 {m}
0.006440670593, -1.425062268032, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
3AAB48,                        !- Name
Roof,                          !- Surface Type
Exterior Roof,                 !- Construction Name
Room8A,                        !- Zone Name
Outdoors,                      !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                    !- Sun Exposure
WindExposed,                   !- Wind Exposure
,                               !- View Factor to Ground
4,                              !- Number of Vertices
0.006440670593, 0.024937731968, 2.810000000000,
                                !- X,Y,Z 1 {m}
0.006440670593, -1.425062268032, 2.810000000000,
                                !- X,Y,Z 2 {m}
2.071440670593, -1.425062268032, 2.810000000000,
                                !- X,Y,Z 3 {m}
2.071440670593, 0.024937731968, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
7D5B04,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room8A,                        !- Zone Name
Zone,                          !- Outside Boundary Condition
Room9,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                              !- Number of Vertices
2.071440670593, -1.425062268032, 2.810000000000,
                                !- X,Y,Z 1 {m}
2.071440670593, -1.425062268032, 0.000000000000,
                                !- X,Y,Z 2 {m}
2.071440670593, 0.024937731968, 0.000000000000,
                                !- X,Y,Z 3 {m}
2.071440670593, 0.024937731968, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
4D607C,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room8A,                        !- Zone Name
Zone,                          !- Outside Boundary Condition
Room8,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                              !- Number of Vertices
2.071440670593, 0.024937731968, 2.810000000000,
                                !- X,Y,Z 1 {m}
2.071440670593, 0.024937731968, 0.000000000000,
                                !- X,Y,Z 2 {m}
0.006440670593, 0.024937731968, 0.000000000000,
                                !- X,Y,Z 3 {m}
0.006440670593, 0.024937731968, 2.810000000000;
                                !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
544CE2,                !- Name
Floor,                 !- Surface Type
Exterior Floor,        !- Construction Name
Room8B,                !- Zone Name
Ground,                !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
2.063581670593, 1.457566731968, 0.000000000000,
                        !- X,Y,Z 1 {m}
2.063581670593, 0.007566731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.001418329407, 0.007566731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.001418329407, 1.457566731968, 0.000000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
3FF307,                !- Name
Roof,                  !- Surface Type
Exterior Roof,         !- Construction Name
Room8B,                !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
-0.001418329407, 1.457566731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.001418329407, 0.007566731968, 2.810000000000,
                        !- X,Y,Z 2 {m}
2.063581670593, 0.007566731968, 2.810000000000,
                        !- X,Y,Z 3 {m}
2.063581670593, 1.457566731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
A7F575,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room8B,                !- Zone Name
Zone,                  !- Outside Boundary Condition
Room8A,                !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
2.063581670593, 0.007566731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
2.063581670593, 0.007566731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
2.063581670593, 1.457566731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
2.063581670593, 1.457566731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
6E354B,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name

```

```

Room8B,                !- Zone Name
Zone,                  !- Outside Boundary Condition
Room1,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
-0.001418329407, 0.007566731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.001418329407, 0.007566731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
2.063581670593, 0.007566731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
2.063581670593, 0.007566731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
856E50,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room8B,                !- Zone Name
Zone,                  !- Outside Boundary Condition
Room1,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
-0.001418329407, 1.457566731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.001418329407, 1.457566731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.001418329407, 0.007566731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.001418329407, 0.007566731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
2449BC,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room8B,                !- Zone Name
Zone,                  !- Outside Boundary Condition
Room8,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
2.063581670593, 1.457566731968, 2.810000000000,
                        !- X,Y,Z 1 {m}
2.063581670593, 1.457566731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.001418329407, 1.457566731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.001418329407, 1.457566731968, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
C1625D,                !- Name
Floor,                 !- Surface Type
Exterior Floor,        !- Construction Name
Room2,                 !- Zone Name
Ground,                !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure

```

```

NoWind,                !- Wind Exposure
,                      !- View Factor to Ground
4,                      !- Number of Vertices
3.053838787653, -0.025946327433, 0.000000000000,
                        !- X,Y,Z 1 {m}
3.053838787653, -7.315946327433, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.066161212347, -7.315946327433, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.066161212347, -0.025946327433, 0.000000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed ,
EA393B,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room2,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room1,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                      !- Number of Vertices
3.053838787653, -7.315946327433, 2.810000000000,
                        !- X,Y,Z 1 {m}
3.053838787653, -7.315946327433, 0.000000000000,
                        !- X,Y,Z 2 {m}
3.053838787653, -0.025946327433, 0.000000000000,
                        !- X,Y,Z 3 {m}
3.053838787653, -0.025946327433, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed ,
E4C736,                !- Name
Roof,                  !- Surface Type
Exterior Roof,         !- Construction Name
Room2,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                      !- Number of Vertices
-0.066161212347, -0.025946327433, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.066161212347, -7.315946327433, 2.810000000000,
                        !- X,Y,Z 2 {m}
3.053838787653, -7.315946327433, 2.810000000000,
                        !- X,Y,Z 3 {m}
3.053838787653, -0.025946327433, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed ,
85EB4A,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room2,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room1,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                      !- Number of Vertices
-0.066161212347, -7.315946327433, 2.810000000000,

```

```

!- X,Y,Z 1 {m}
-0.066161212347, -7.315946327433, 0.000000000000,
!- X,Y,Z 2 {m}
3.053838787653, -7.315946327433, 0.000000000000,
!- X,Y,Z 3 {m}
3.053838787653, -7.315946327433, 2.810000000000;
!- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

CAFEF1, !- Name
Wall, !- Surface Type
Interior Wall, !- Construction Name
Room2, !- Zone Name
Zone, !- Outside Boundary Condition
Room1, !- Outside Boundary Condition Object
NoSun, !- Sun Exposure
NoWind, !- Wind Exposure
0.0, !- View Factor to Ground
4, !- Number of Vertices
-0.066161212347, -0.025946327433, 2.810000000000,
!- X,Y,Z 1 {m}
-0.066161212347, -0.025946327433, 0.000000000000,
!- X,Y,Z 2 {m}
-0.066161212347, -7.315946327433, 0.000000000000,
!- X,Y,Z 3 {m}
-0.066161212347, -7.315946327433, 2.810000000000;
!- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

EDAC29, !- Name
Wall, !- Surface Type
Interior Wall, !- Construction Name
Room2, !- Zone Name
Zone, !- Outside Boundary Condition
Room1, !- Outside Boundary Condition Object
NoSun, !- Sun Exposure
NoWind, !- Wind Exposure
0.0, !- View Factor to Ground
4, !- Number of Vertices
3.053838787653, -0.025946327433, 2.810000000000,
!- X,Y,Z 1 {m}
3.053838787653, -0.025946327433, 0.000000000000,
!- X,Y,Z 2 {m}
-0.066161212347, -0.025946327433, 0.000000000000,
!- X,Y,Z 3 {m}
-0.066161212347, -0.025946327433, 2.810000000000;
!- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

43DEAB, !- Name
Floor, !- Surface Type
Exterior Floor, !- Construction Name
Room3, !- Zone Name
Ground, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
NoSun, !- Sun Exposure
NoWind, !- Wind Exposure
, !- View Factor to Ground
4, !- Number of Vertices
3.579999670593, 6.160000273697, 0.000000000000,
!- X,Y,Z 1 {m}
3.579999670593, 0.000000273697, 0.000000000000,
!- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,

```

```

                                !- X,Y,Z 3 {m}
-0.000000329407, 6.160000273697, 0.000000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
CCFIDB,                        !- Name
Roof,                          !- Surface Type
Exterior Roof,                 !- Construction Name
Room3,                         !- Zone Name
Outdoors,                      !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                    !- Sun Exposure
WindExposed,                   !- Wind Exposure
,                               !- View Factor to Ground
4,                              !- Number of Vertices
-0.000000329407, 6.160000273697, 2.810000000000,
                                !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 2.810000000000,
                                !- X,Y,Z 2 {m}
3.579999670593, 0.000000273697, 2.810000000000,
                                !- X,Y,Z 3 {m}
3.579999670593, 6.160000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
12CSED,                        !- Name
Wall,                          !- Surface Type
Exterior Wall,                 !- Construction Name
Room3,                         !- Zone Name
Outdoors,                      !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                    !- Sun Exposure
WindExposed,                   !- Wind Exposure
,                               !- View Factor to Ground
4,                              !- Number of Vertices
-0.000000329407, 6.160000273697, 2.810000000000,
                                !- X,Y,Z 1 {m}
-0.000000329407, 6.160000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
-0.000000329407, 0.000000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
B9E604,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room3,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room1,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                              !- Number of Vertices
3.579999670593, 0.000000273697, 2.810000000000,
                                !- X,Y,Z 1 {m}
3.579999670593, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
3.579999670593, 6.160000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
3.579999670593, 6.160000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
7581C8,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room3,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room1,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
3.579999670593, 6.160000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
3.579999670593, 6.160000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.000000329407, 6.160000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.000000329407, 6.160000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
146CEB,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room3,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room4,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
-0.000000329407, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
3.579999670593, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
3.579999670593, 0.000000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
583722,                !- Name
Floor,                 !- Surface Type
Exterior Floor,        !- Construction Name
Room1,                 !- Zone Name
Ground,                !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
,                      !- View Factor to Ground
8,                     !- Number of Vertices
10.145280670593, -0.176488268032, 0.000000000000,
                        !- X,Y,Z 1 {m}
10.146699000000, -9.954055000000, 0.000000000000,
                        !- X,Y,Z 2 {m}
14.275280670593, -9.946488268032, 0.000000000000,
                        !- X,Y,Z 3 {m}
14.275280670593, -12.189375726303, 0.000000000000,
                        !- X,Y,Z 4 {m}
3.745280670593, -12.189375726303, 0.000000000000,
                        !- X,Y,Z 5 {m}
3.745280670593, -6.029375726303, 0.000000000000,
                        !- X,Y,Z 6 {m}
0.165280670593, -6.029375726303, 0.000000000000,

```

```

                                !- X,Y,Z 7 {m}
0.165280670593, -0.176488268032, 0.000000000000;
                                !- X,Y,Z 8 {m}

BuildingSurface:Detailed,
7D9BA0,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room1,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room4,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                              !- Number of Vertices
3.745280670593, -12.189375726303, 2.810000000000,
                                !- X,Y,Z 1 {m}
3.745280670593, -12.189375726303, 0.000000000000,
                                !- X,Y,Z 2 {m}
5.225280670593, -12.189375726303, 0.000000000000,
                                !- X,Y,Z 3 {m}
5.225280670593, -12.189375726303, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
188910,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room1,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room3,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                              !- Number of Vertices
3.745280670593, -6.029375726303, 2.810000000000,
                                !- X,Y,Z 1 {m}
3.745280670593, -6.029375726303, 0.000000000000,
                                !- X,Y,Z 2 {m}
3.745280670593, -12.189375726303, 0.000000000000,
                                !- X,Y,Z 3 {m}
3.745280670593, -12.189375726303, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
344A74,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room1,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room3,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                              !- Number of Vertices
0.165280670593, -6.029375726303, 2.810000000000,
                                !- X,Y,Z 1 {m}
0.165280670593, -6.029375726303, 0.000000000000,
                                !- X,Y,Z 2 {m}
3.745280670593, -6.029375726303, 0.000000000000,
                                !- X,Y,Z 3 {m}
3.745280670593, -6.029375726303, 2.810000000000;
                                !- X,Y,Z 4 {m}

```

```

BuildingSurface : Detailed ,
CF54A3,                !- Name
Wall,                  !- Surface Type
Exterior Wall,         !- Construction Name
Room1,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
10.146699000000, -9.954055000000, 2.810000000000,
                        !- X,Y,Z 1 {m}
10.146699000000, -9.954055000000, 0.000000000000,
                        !- X,Y,Z 2 {m}
10.145280670593, -0.176488268032, 0.000000000000,
                        !- X,Y,Z 3 {m}
10.145280670593, -0.176488268032, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface : Detailed ,
A28E69,                !- Name
Wall,                  !- Surface Type
Exterior Wall,         !- Construction Name
Room1,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
0.165280670593, -0.176488268032, 2.810000000000,
                        !- X,Y,Z 1 {m}
0.165280670593, -0.176488268032, 0.000000000000,
                        !- X,Y,Z 2 {m}
0.165280670593, -6.029375726303, 0.000000000000,
                        !- X,Y,Z 3 {m}
0.165280670593, -6.029375726303, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface : Detailed ,
58E2C5,                !- Name
Roof,                  !- Surface Type
Exterior Roof,         !- Construction Name
Room1,                 !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
8,                     !- Number of Vertices
0.165280670593, -0.176488268032, 2.810000000000,
                        !- X,Y,Z 1 {m}
0.165280670593, -6.029375726303, 2.810000000000,
                        !- X,Y,Z 2 {m}
3.745280670593, -6.029375726303, 2.810000000000,
                        !- X,Y,Z 3 {m}
3.745280670593, -12.189375726303, 2.810000000000,
                        !- X,Y,Z 4 {m}
14.275280670593, -12.189375726303, 2.810000000000,
                        !- X,Y,Z 5 {m}
14.275280670593, -9.946488268032, 2.810000000000,
                        !- X,Y,Z 6 {m}
10.146699000000, -9.954055000000, 2.810000000000,

```

```

                                !- X,Y,Z 7 {m}
10.145280670593, -0.176488268032, 2.810000000000;
                                !- X,Y,Z 8 {m}

BuildingSurface:Detailed,
C80A69,                        !- Name
Wall,                          !- Surface Type
Exterior Wall,                 !- Construction Name
Room1,                         !- Zone Name
Outdoors,                      !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                    !- Sun Exposure
WindExposed,                   !- Wind Exposure
,                               !- View Factor to Ground
4,                              !- Number of Vertices
10.145280670593, -0.176488268032, 2.810000000000,
                                !- X,Y,Z 1 {m}
10.145280670593, -0.176488268032, 0.000000000000,
                                !- X,Y,Z 2 {m}
0.165280670593, -0.176488268032, 0.000000000000,
                                !- X,Y,Z 3 {m}
0.165280670593, -0.176488268032, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
998797,                        !- Name
Wall,                          !- Surface Type
Exterior Wall,                 !- Construction Name
Room1,                         !- Zone Name
Outdoors,                      !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                    !- Sun Exposure
WindExposed,                   !- Wind Exposure
,                               !- View Factor to Ground
4,                              !- Number of Vertices
14.275280670593, -9.946488268032, 2.810000000000,
                                !- X,Y,Z 1 {m}
14.275280670593, -9.946488268032, 0.000000000000,
                                !- X,Y,Z 2 {m}
10.146699000000, -9.954055000000, 0.000000000000,
                                !- X,Y,Z 3 {m}
10.146699000000, -9.954055000000, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
C3851F,                        !- Name
Wall,                          !- Surface Type
Exterior Wall,                 !- Construction Name
Room1,                         !- Zone Name
Outdoors,                      !- Outside Boundary Condition
,                               !- Outside Boundary Condition Object
SunExposed,                    !- Sun Exposure
WindExposed,                   !- Wind Exposure
,                               !- View Factor to Ground
4,                              !- Number of Vertices
14.275280670593, -12.189375726303, 2.810000000000,
                                !- X,Y,Z 1 {m}
14.275280670593, -12.189375726303, 0.000000000000,
                                !- X,Y,Z 2 {m}
14.275280670593, -9.946488268032, 0.000000000000,
                                !- X,Y,Z 3 {m}
14.275280670593, -9.946488268032, 2.810000000000;
                                !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
672EFE,                !- Name
Floor,                 !- Surface Type
Exterior Floor,        !- Construction Name
Room11,                !- Zone Name
Ground,                !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
1.669999670593, 4.810000273697, 0.000000000000,
                        !- X,Y,Z 1 {m}
1.669999670593, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.000000329407, 4.810000273697, 0.000000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
836244,                !- Name
Roof,                  !- Surface Type
Exterior Roof,         !- Construction Name
Room11,                !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
-0.000000329407, 4.810000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 2 {m}
1.669999670593, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 3 {m}
1.669999670593, 4.810000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
B75849,                !- Name
Wall,                  !- Surface Type
Exterior Wall,         !- Construction Name
Room11,                !- Zone Name
Outdoors,              !- Outside Boundary Condition
,                      !- Outside Boundary Condition Object
SunExposed,            !- Sun Exposure
WindExposed,           !- Wind Exposure
,                      !- View Factor to Ground
4,                     !- Number of Vertices
-0.000000329407, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
1.669999670593, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
1.669999670593, 0.000000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

```

```

BuildingSurface:Detailed,
F46CCD,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name

```

```

Room11,           !- Zone Name
Zone,             !- Outside Boundary Condition
Room9,            !- Outside Boundary Condition Object
NoSun,            !- Sun Exposure
NoWind,           !- Wind Exposure
0.0,              !- View Factor to Ground
4,                !- Number of Vertices
1.669999670593, 4.810000273697, 2.810000000000,
                  !- X,Y,Z 1 {m}
1.669999670593, 4.810000273697, 0.000000000000,
                  !- X,Y,Z 2 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                  !- X,Y,Z 3 {m}
-0.000000329407, 4.810000273697, 2.810000000000;
                  !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
85AF29,           !- Name
Wall,             !- Surface Type
Interior Wall,    !- Construction Name
Room11,           !- Zone Name
Zone,             !- Outside Boundary Condition
Room7,            !- Outside Boundary Condition Object
NoSun,            !- Sun Exposure
NoWind,           !- Wind Exposure
0.0,              !- View Factor to Ground
4,                !- Number of Vertices
-0.000000329407, 4.810000273697, 2.810000000000,
                  !- X,Y,Z 1 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                  !- X,Y,Z 2 {m}
-0.000000329407, 0.000000273697, 0.000000000000,
                  !- X,Y,Z 3 {m}
-0.000000329407, 0.000000273697, 2.810000000000;
                  !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
D9CDCA,           !- Name
Wall,             !- Surface Type
Exterior Wall,    !- Construction Name
Room11,           !- Zone Name
Zone,             !- Outside Boundary Condition
Room10,           !- Outside Boundary Condition Object
NoSun,            !- Sun Exposure
NoWind,           !- Wind Exposure
0.0,              !- View Factor to Ground
4,                !- Number of Vertices
1.669999670593, 0.000000273697, 2.810000000000,
                  !- X,Y,Z 1 {m}
1.669999670593, 0.000000273697, 0.000000000000,
                  !- X,Y,Z 2 {m}
1.669999670593, 4.810000273697, 0.000000000000,
                  !- X,Y,Z 3 {m}
1.669999670593, 4.810000273697, 2.810000000000;
                  !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
122039,           !- Name
Wall,             !- Surface Type
Interior Wall,    !- Construction Name
Room4,            !- Zone Name
Zone,             !- Outside Boundary Condition
Room3,            !- Outside Boundary Condition Object
NoSun,            !- Sun Exposure

```

```

NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
3.745280670593, 4.973972273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
3.745280670593, 4.973972273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
0.165280670593, 4.973972273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
0.165280670593, 4.973972273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed ,
5FE841,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room7,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room1,                 !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
0.189999670593, 4.810000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
0.189999670593, 4.810000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.000000329407, 4.810000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.000000329407, 4.810000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed ,
84806A,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room9,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room11,                !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
3.943935670593, 0.000000273697, 2.810000000000,
                        !- X,Y,Z 1 {m}
3.943935670593, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 2 {m}
5.613935670593, 0.000000273697, 0.000000000000,
                        !- X,Y,Z 3 {m}
5.613935670593, 0.000000273697, 2.810000000000;
                        !- X,Y,Z 4 {m}

BuildingSurface:Detailed ,
DB6846,                !- Name
Wall,                  !- Surface Type
Interior Wall,         !- Construction Name
Room9,                 !- Zone Name
Zone,                  !- Outside Boundary Condition
Room8A,                !- Outside Boundary Condition Object
NoSun,                 !- Sun Exposure
NoWind,                !- Wind Exposure
0.0,                   !- View Factor to Ground
4,                     !- Number of Vertices
-0.026064329407, 3.692887731968, 2.810000000000,

```

```

                                !- X,Y,Z 1 {m}
-0.026064329407, 3.692887731968, 0.000000000000,
                                !- X,Y,Z 2 {m}
-0.026064329407, 2.242887731968, 0.000000000000,
                                !- X,Y,Z 3 {m}
-0.026064329407, 2.242887731968, 2.810000000000;
                                !- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

247A41,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room9,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room1,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                             !- Number of Vertices
-0.026064329407, 2.242887731968, 2.810000000000,
                                !- X,Y,Z 1 {m}
-0.026064329407, 2.242887731968, 0.000000000000,
                                !- X,Y,Z 2 {m}
-0.026064329407, 0.000000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
-0.026064329407, 0.000000273697, 2.810000000000;
                                !- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

56C07A,                       !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room8,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room8A,                       !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                             !- Number of Vertices
2.021167670593, 0.034017731968, 2.810000000000,
                                !- X,Y,Z 1 {m}
2.021167670593, 0.034017731968, 0.000000000000,
                                !- X,Y,Z 2 {m}
4.086167670593, 0.034017731968, 0.000000000000,
                                !- X,Y,Z 3 {m}
4.086167670593, 0.034017731968, 2.810000000000;
                                !- X,Y,Z 4 {m}

```

BuildingSurface:Detailed ,

```

988527,                       !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room1,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room5,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                             !- Number of Vertices
5.225280670593, -12.189375726303, 2.810000000000,
                                !- X,Y,Z 1 {m}
5.225280670593, -12.189375726303, 0.000000000000,
                                !- X,Y,Z 2 {m}
8.725280670593, -12.189375726303, 0.000000000000,

```

```

                                !- X,Y,Z 3 {m}
8.725280670593, -12.189375726303, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
B54F20,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room1,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room6,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                             !- Number of Vertices
8.725280670593, -12.189375726303, 2.810000000000,
                                !- X,Y,Z 1 {m}
8.725280670593, -12.189375726303, 0.000000000000,
                                !- X,Y,Z 2 {m}
14.085280670593, -12.189375726303, 0.000000000000,
                                !- X,Y,Z 3 {m}
14.085280670593, -12.189375726303, 2.810000000000;
                                !- X,Y,Z 4 {m}

BuildingSurface:Detailed,
4E3184,                        !- Name
Wall,                          !- Surface Type
Interior Wall,                 !- Construction Name
Room1,                         !- Zone Name
Zone,                          !- Outside Boundary Condition
Room7,                         !- Outside Boundary Condition Object
NoSun,                         !- Sun Exposure
NoWind,                        !- Wind Exposure
0.0,                           !- View Factor to Ground
4,                             !- Number of Vertices
14.085280670593, -12.189375726303, 2.810000000000,
                                !- X,Y,Z 1 {m}
14.085280670593, -12.189375726303, 0.000000000000,
                                !- X,Y,Z 2 {m}
14.275280670593, -12.189375726303, 0.000000000000,
                                !- X,Y,Z 3 {m}
14.275280670593, -12.189375726303, 2.810000000000;
                                !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
FE6C4A,                        !- Name
Window,                       !- Surface Type
Exterior Window,              !- Construction Name
12C5ED,                       !- Building Surface Name
,                              !- Outside Boundary Condition Object
,                              !- View Factor to Ground
,                              !- Shading Control Name
,                              !- Frame and Divider Name
,                              !- Multiplier
4,                             !- Number of Vertices
-0.000000329407, 2.515281024135, 1.977000000000,
                                !- X,Y,Z 1 {m}
-0.000000329407, 2.515281024135, 0.910000000000,
                                !- X,Y,Z 2 {m}
-0.000000329407, 0.488100490714, 0.910000000000,
                                !- X,Y,Z 3 {m}
-0.000000329407, 0.488100490714, 1.977000000000;
                                !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
6EE8E6,                !- Name
Window,                !- Surface Type
Exterior Window,       !- Construction Name
446665,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name
,                      !- Frame and Divider Name
,                      !- Multiplier
4,                     !- Number of Vertices
0.165280670593, 3.805082273192, 1.977000000000,
                                !- X,Y,Z 1 {m}
0.165280670593, 3.805082273192, 0.910000000000,
                                !- X,Y,Z 2 {m}
0.165280670593, 1.033646947971, 0.910000000000,
                                !- X,Y,Z 3 {m}
0.165280670593, 1.033646947971, 1.977000000000;
                                !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
C60837,                !- Name
Window,                !- Surface Type
Exterior Window,       !- Construction Name
ECC2C5,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name
,                      !- Frame and Divider Name
,                      !- Multiplier
4,                     !- Number of Vertices
0.445898518552, 0.000000273697, 1.977000000000,
                                !- X,Y,Z 1 {m}
0.445898518552, 0.000000273697, 0.910000000000,
                                !- X,Y,Z 2 {m}
2.776165584850, 0.000000273697, 0.910000000000,
                                !- X,Y,Z 3 {m}
2.776165584850, 0.000000273697, 1.977000000000;
                                !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
7EF545,                !- Name
Window,                !- Surface Type
Exterior Window,       !- Construction Name
BF9A90,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name
,                      !- Frame and Divider Name
,                      !- Multiplier
4,                     !- Number of Vertices
1.927322788081, 0.000000273697, 1.977000000000,
                                !- X,Y,Z 1 {m}
1.927322788081, 0.000000273697, 0.910000000000,
                                !- X,Y,Z 2 {m}
3.305923299506, 0.000000273697, 0.910000000000,
                                !- X,Y,Z 3 {m}
3.305923299506, 0.000000273697, 1.977000000000;
                                !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
BFAF7F,                !- Name
Window,                !- Surface Type
Exterior Window,       !- Construction Name

```

```

653AE9,          !- Building Surface Name
,                !- Outside Boundary Condition Object
,                !- View Factor to Ground
,                !- Shading Control Name
,                !- Frame and Divider Name
,                !- Multiplier
4,               !- Number of Vertices
0.652472374161, 0.000000273697, 1.977000000000,
                  !- X,Y,Z 1 {m}
0.652472374161, 0.000000273697, 0.910000000000,
                  !- X,Y,Z 2 {m}
2.991046466010, 0.000000273697, 0.910000000000,
                  !- X,Y,Z 3 {m}
2.991046466010, 0.000000273697, 1.977000000000;
                  !- X,Y,Z 4 {m}

FenestrationSurface:Detailed ,
774BE9,          !- Name
Window,          !- Surface Type
Exterior Window, !- Construction Name
A28E69,          !- Building Surface Name
,                !- Outside Boundary Condition Object
,                !- View Factor to Ground
,                !- Shading Control Name
,                !- Frame and Divider Name
,                !- Multiplier
4,               !- Number of Vertices
0.165280670593, -1.154103183117, 1.977000000000,
                  !- X,Y,Z 1 {m}
0.165280670593, -1.154103183117, 0.910000000000,
                  !- X,Y,Z 2 {m}
0.165280670593, -3.932234766505, 0.910000000000,
                  !- X,Y,Z 3 {m}
0.165280670593, -3.932234766505, 1.977000000000;
                  !- X,Y,Z 4 {m}

FenestrationSurface:Detailed ,
E7B1C7,          !- Name
Window,          !- Surface Type
Exterior Window, !- Construction Name
C80A69,          !- Building Surface Name
,                !- Outside Boundary Condition Object
,                !- View Factor to Ground
,                !- Shading Control Name
,                !- Frame and Divider Name
,                !- Multiplier
4,               !- Number of Vertices
5.155280670593, -0.176488268032, 1.977000000000,
                  !- X,Y,Z 1 {m}
5.155280670593, -0.176488268032, 0.910000000000,
                  !- X,Y,Z 2 {m}
2.821104296833, -0.176488268032, 0.910000000000,
                  !- X,Y,Z 3 {m}
2.821104296833, -0.176488268032, 1.977000000000;
                  !- X,Y,Z 4 {m}

FenestrationSurface:Detailed ,
435590,          !- Name
Window,          !- Surface Type
Exterior Window, !- Construction Name
C80A69,          !- Building Surface Name
,                !- Outside Boundary Condition Object
,                !- View Factor to Ground
,                !- Shading Control Name

```

```

,                                     !- Frame and Divider Name
,                                     !- Multiplier
4,                                    !- Number of Vertices
8.017095043553, -0.176488268032, 1.977000000000,
                                     !- X,Y,Z 1 {m}
8.017095043553, -0.176488268032, 0.910000000000,
                                     !- X,Y,Z 2 {m}
6.873299879269, -0.176488268032, 0.910000000000,
                                     !- X,Y,Z 3 {m}
6.873299879269, -0.176488268032, 1.977000000000;
                                     !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
4BEB8F,                             !- Name
Window,                             !- Surface Type
Exterior Window,                    !- Construction Name
9F916F,                             !- Building Surface Name
,                                     !- Outside Boundary Condition Object
,                                     !- View Factor to Ground
,                                     !- Shading Control Name
,                                     !- Frame and Divider Name
,                                     !- Multiplier
4,                                    !- Number of Vertices
3.050216896649, 12.012887731968, 1.977000000000,
                                     !- X,Y,Z 1 {m}
3.050216896649, 12.012887731968, 0.910000000000,
                                     !- X,Y,Z 2 {m}
0.646765570748, 12.012887731968, 0.910000000000,
                                     !- X,Y,Z 3 {m}
0.646765570748, 12.012887731968, 1.977000000000;
                                     !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
67DF64,                             !- Name
Door,                               !- Surface Type
Exterior Door,                      !- Construction Name
E908B6,                             !- Building Surface Name
,                                     !- Outside Boundary Condition Object
,                                     !- View Factor to Ground
,                                     !- Shading Control Name
,                                     !- Frame and Divider Name
,                                     !- Multiplier
4,                                    !- Number of Vertices
4.039998877550, 0.000000273697, 2.080000000000,
                                     !- X,Y,Z 1 {m}
4.039998877550, 0.000000273697, 0.000000000000,
                                     !- X,Y,Z 2 {m}
4.943783008511, 0.000000273697, 0.000000000000,
                                     !- X,Y,Z 3 {m}
4.943783008511, 0.000000273697, 2.080000000000;
                                     !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
DC2F63,                             !- Name
Door,                               !- Surface Type
Exterior Door,                      !- Construction Name
8A40EE,                             !- Building Surface Name
,                                     !- Outside Boundary Condition Object
,                                     !- View Factor to Ground
,                                     !- Shading Control Name
,                                     !- Frame and Divider Name
,                                     !- Multiplier
4,                                    !- Number of Vertices
3.285809571876, 16.822887731968, 2.080000000000,

```

```

                                !- X,Y,Z 1 {m}
3.285809571876, 16.822887731968, 0.000000000000,
                                !- X,Y,Z 2 {m}
2.358599134438, 16.822887731968, 0.000000000000,
                                !- X,Y,Z 3 {m}
2.358599134438, 16.822887731968, 2.080000000000;
                                !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
4080DF,                        !- Name
Door,                          !- Surface Type
Exterior Door,                 !- Construction Name
9F916F,                        !- Building Surface Name
,                               !- Outside Boundary Condition Object
,                               !- View Factor to Ground
,                               !- Shading Control Name
,                               !- Frame and Divider Name
,                               !- Multiplier
4,                              !- Number of Vertices
5.363575569057, 12.012887731968, 2.080000000000,
                                !- X,Y,Z 1 {m}
5.363575569057, 12.012887731968, 0.000000000000,
                                !- X,Y,Z 2 {m}
4.502633033478, 12.012887731968, 0.000000000000,
                                !- X,Y,Z 3 {m}
4.502633033478, 12.012887731968, 2.080000000000;
                                !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
910D52,                        !- Name
Door,                          !- Surface Type
Exterior Door,                 !- Construction Name
C80A69,                        !- Building Surface Name
,                               !- Outside Boundary Condition Object
,                               !- View Factor to Ground
,                               !- Shading Control Name
,                               !- Frame and Divider Name
,                               !- Multiplier
4,                              !- Number of Vertices
6.785879283529, -0.176488268032, 2.080000000000,
                                !- X,Y,Z 1 {m}
6.785879283529, -0.176488268032, 0.000000000000,
                                !- X,Y,Z 2 {m}
5.882069597124, -0.176488268032, 0.000000000000,
                                !- X,Y,Z 3 {m}
5.882069597124, -0.176488268032, 2.080000000000;
                                !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
58B2A1,                        !- Name
Door,                          !- Surface Type
Interior Door,                 !- Construction Name
7991EC,                        !- Building Surface Name
,                               !- Outside Boundary Condition Object
,                               !- View Factor to Ground
,                               !- Shading Control Name
,                               !- Frame and Divider Name
,                               !- Multiplier
4,                              !- Number of Vertices
5.081880408718, 4.973972273697, 2.080000000000,
                                !- X,Y,Z 1 {m}
5.081880408718, 4.973972273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
4.217807204085, 4.973972273697, 0.000000000000,

```

```

                                !- X,Y,Z 3 {m}
4.217807204085, 4.973972273697, 2.080000000000;
                                !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
92F431,                        !- Name
Door,                          !- Surface Type
Interior Door,                 !- Construction Name
864EDD,                        !- Building Surface Name
,                               !- Outside Boundary Condition Object
,                               !- View Factor to Ground
,                               !- Shading Control Name
,                               !- Frame and Divider Name
,                               !- Multiplier
4,                              !- Number of Vertices
3.309458656307, 4.810000273697, 2.080000000000,
                                !- X,Y,Z 1 {m}
3.309458656307, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
2.381570468688, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
2.381570468688, 4.810000273697, 2.080000000000;
                                !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
D34249,                        !- Name
Door,                          !- Surface Type
Interior Door,                 !- Construction Name
4AFBB9,                        !- Building Surface Name
,                               !- Outside Boundary Condition Object
,                               !- View Factor to Ground
,                               !- Shading Control Name
,                               !- Frame and Divider Name
,                               !- Multiplier
4,                              !- Number of Vertices
1.048764678823, 4.810000273697, 2.080000000000,
                                !- X,Y,Z 1 {m}
1.048764678823, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
0.174151263794, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
0.174151263794, 4.810000273697, 2.080000000000;
                                !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
F1C641,                        !- Name
Door,                          !- Surface Type
Interior Door,                 !- Construction Name
F85C1E,                        !- Building Surface Name
,                               !- Outside Boundary Condition Object
,                               !- View Factor to Ground
,                               !- Shading Control Name
,                               !- Frame and Divider Name
,                               !- Multiplier
4,                              !- Number of Vertices
3.942610414785, 4.810000273697, 2.080000000000,
                                !- X,Y,Z 1 {m}
3.942610414785, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 2 {m}
3.090948946859, 4.810000273697, 0.000000000000,
                                !- X,Y,Z 3 {m}
3.090948946859, 4.810000273697, 2.080000000000;
                                !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
BE7414,                !- Name
Door,                  !- Surface Type
Interior Door,         !- Construction Name
247A41,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name
,                      !- Frame and Divider Name
,                      !- Multiplier
4,                     !- Number of Vertices
-0.026064329407, 1.764375655680, 2.080000000000,
                        !- X,Y,Z 1 {m}
-0.026064329407, 1.764375655680, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.026064329407, 0.495022383975, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.026064329407, 0.495022383975, 2.080000000000;
                        !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
BEC835,                !- Name
Door,                  !- Surface Type
Interior Door,         !- Construction Name
B8827C,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name
,                      !- Frame and Divider Name
,                      !- Multiplier
4,                     !- Number of Vertices
-0.000000329407, 15.629470792036, 2.080000000000,
                        !- X,Y,Z 1 {m}
-0.000000329407, 15.629470792036, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.000000329407, 14.355674410609, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.000000329407, 14.355674410609, 2.080000000000;
                        !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
FF15C2,                !- Name
Door,                  !- Surface Type
Interior Door,         !- Construction Name
8F0729,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name
,                      !- Frame and Divider Name
,                      !- Multiplier
4,                     !- Number of Vertices
4.086167670593, 6.181905239417, 2.080000000000,
                        !- X,Y,Z 1 {m}
4.086167670593, 6.181905239417, 0.000000000000,
                        !- X,Y,Z 2 {m}
4.086167670593, 7.378145381956, 0.000000000000,
                        !- X,Y,Z 3 {m}
4.086167670593, 7.378145381956, 2.080000000000;
                        !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
39FD64,                !- Name
Door,                  !- Surface Type
Interior Door,         !- Construction Name

```

```

56C07A,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name
,                      !- Frame and Divider Name
,                      !- Multiplier
4,                      !- Number of Vertices
2.674606096689, 0.034017731968, 2.080000000000,
                        !- X,Y,Z 1 {m}
2.674606096689, 0.034017731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
3.417608090485, 0.034017731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
3.417608090485, 0.034017731968, 2.080000000000;
                        !- X,Y,Z 4 {m}

FenestrationSurface:Detailed ,
635A6C,                !- Name
Door,                  !- Surface Type
Interior Door,         !- Construction Name
317104,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name
,                      !- Frame and Divider Name
,                      !- Multiplier
4,                      !- Number of Vertices
-0.043832329407, 7.508731364224, 2.080000000000,
                        !- X,Y,Z 1 {m}
-0.043832329407, 7.508731364224, 0.000000000000,
                        !- X,Y,Z 2 {m}
-0.043832329407, 6.279703275805, 0.000000000000,
                        !- X,Y,Z 3 {m}
-0.043832329407, 6.279703275805, 2.080000000000;
                        !- X,Y,Z 4 {m}

FenestrationSurface:Detailed ,
D0584D,                !- Name
Door,                  !- Surface Type
Interior Door,         !- Construction Name
6E354B,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name
,                      !- Frame and Divider Name
,                      !- Multiplier
4,                      !- Number of Vertices
0.743446340122, 0.007566731968, 2.080000000000,
                        !- X,Y,Z 1 {m}
0.743446340122, 0.007566731968, 0.000000000000,
                        !- X,Y,Z 2 {m}
1.499029057860, 0.007566731968, 0.000000000000,
                        !- X,Y,Z 3 {m}
1.499029057860, 0.007566731968, 2.080000000000;
                        !- X,Y,Z 4 {m}

FenestrationSurface:Detailed ,
EA00F3,                !- Name
Door,                  !- Surface Type
Interior Door,         !- Construction Name
EA393B,                !- Building Surface Name
,                      !- Outside Boundary Condition Object
,                      !- View Factor to Ground
,                      !- Shading Control Name

```

```

,                                     !- Frame and Divider Name
,                                     !- Multiplier
4,                                   !- Number of Vertices
3.053838787653, -6.907468622761, 2.080000000000,
                                     !- X,Y,Z 1 {m}
3.053838787653, -6.907468622761, 0.000000000000,
                                     !- X,Y,Z 2 {m}
3.053838787653, -5.641874296056, 0.000000000000,
                                     !- X,Y,Z 3 {m}
3.053838787653, -5.641874296056, 2.080000000000;
                                     !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
94659B,                             !- Name
Door,                               !- Surface Type
Interior Door,                      !- Construction Name
EA393B,                             !- Building Surface Name
,                                   !- Outside Boundary Condition Object
,                                   !- View Factor to Ground
,                                   !- Shading Control Name
,                                   !- Frame and Divider Name
,                                   !- Multiplier
4,                                   !- Number of Vertices
3.053838787653, -1.443736726734, 2.080000000000,
                                     !- X,Y,Z 1 {m}
3.053838787653, -1.443736726734, 0.000000000000,
                                     !- X,Y,Z 2 {m}
3.053838787653, -0.217527289851, 0.000000000000,
                                     !- X,Y,Z 3 {m}
3.053838787653, -0.217527289851, 2.080000000000;
                                     !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
EOE5A7,                             !- Name
Door,                               !- Surface Type
Interior Door,                      !- Construction Name
EA393B,                             !- Building Surface Name
,                                   !- Outside Boundary Condition Object
,                                   !- View Factor to Ground
,                                   !- Shading Control Name
,                                   !- Frame and Divider Name
,                                   !- Multiplier
4,                                   !- Number of Vertices
3.053838787653, -4.633388538197, 2.080000000000,
                                     !- X,Y,Z 1 {m}
3.053838787653, -4.633388538197, 0.000000000000,
                                     !- X,Y,Z 2 {m}
3.053838787653, -2.569908044049, 0.000000000000,
                                     !- X,Y,Z 3 {m}
3.053838787653, -2.569908044049, 2.080000000000;
                                     !- X,Y,Z 4 {m}

FenestrationSurface:Detailed,
22080A,                             !- Name
Door,                               !- Surface Type
Interior Door,                      !- Construction Name
CAFEF1,                             !- Building Surface Name
,                                   !- Outside Boundary Condition Object
,                                   !- View Factor to Ground
,                                   !- Shading Control Name
,                                   !- Frame and Divider Name
,                                   !- Multiplier
4,                                   !- Number of Vertices
-0.066161212347, -3.135650394632, 2.080000000000,

```

```

                                !- X,Y,Z 1 {m}
-0.066161212347, -3.135650394632, 0.000000000000,
                                !- X,Y,Z 2 {m}
-0.066161212347, -4.141903368308, 0.000000000000,
                                !- X,Y,Z 3 {m}
-0.066161212347, -4.141903368308, 2.080000000000;
                                !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
7FCDA3,                        !- Name
Door,                          !- Surface Type
Interior Door,                 !- Construction Name
D9CDCA,                        !- Building Surface Name
,                               !- Outside Boundary Condition Object
,                               !- View Factor to Ground
,                               !- Shading Control Name
,                               !- Frame and Divider Name
,                               !- Multiplier
4,                              !- Number of Vertices
1.669999670593, 0.987600219606, 2.080000000000,
                                !- X,Y,Z 1 {m}
1.669999670593, 0.987600219606, 0.000000000000,
                                !- X,Y,Z 2 {m}
1.669999670593, 3.428429060647, 0.000000000000,
                                !- X,Y,Z 3 {m}
1.669999670593, 3.428429060647, 2.080000000000;
                                !- X,Y,Z 4 {m}

```

```

FenestrationSurface:Detailed,
694D5C,                        !- Name
Door,                          !- Surface Type
Interior Door,                 !- Construction Name
B9E604,                        !- Building Surface Name
,                               !- Outside Boundary Condition Object
,                               !- View Factor to Ground
,                               !- Shading Control Name
,                               !- Frame and Divider Name
,                               !- Multiplier
4,                              !- Number of Vertices
3.579999670593, 0.349522297304, 2.080000000000,
                                !- X,Y,Z 1 {m}
3.579999670593, 0.349522297304, 0.000000000000,
                                !- X,Y,Z 2 {m}
3.579999670593, 1.548278662931, 0.000000000000,
                                !- X,Y,Z 3 {m}
3.579999670593, 1.548278662931, 2.080000000000;
                                !- X,Y,Z 4 {m}

```

APPENDIX E

MATLAB SIMULATION CODE

In this appendix the MATLAB code for both simulations is detailed. The specific files regarding the steady-state optimal control method are detailed first, followed by the dynamic MPC specific files. The final appendix section includes the files used by both methods. The code is thoroughly commented.

E.1 Steady-State Method Specific Code

Main simulation file for steady-state method.

```

1 clear all; close all;
2 % This is the main simulation file for the steady-state optimal control
3 % method.
4
5 % initialize and load necessary variables/data
6 case1 = [];
7 case2 = [];
8 T_room31_data = [];
9 T_room21_data = [];
10 T_room32_data = [];
11 T_room22_data = [];
12 T_room3_data = [];
13 T_room2_data = [];
14 var_data = [];
15 repeat_itr = 0;
16
17 % load('zone_people_1minute_data.mat'); % load in random occupancy data (5-minute
    intervals)
18 % zone_people = zone_people(2:end,:); % shift the occupancy data to match desired
    occupancy timeline
19 load('zone_people_1minute_10day_full_occ.mat'); % load in full occupancy data
20
21 model_SS1 = load('model_steady_state_orig10_no_flow_limit.mat'); % load in steady-
    state model gains
22 model_input_means1 = load('model_input_means_orig10_no_flow_limit.mat'); % load in
    the required mean data for the model
23 model_SS2 = load('model_steady_state_orig14_no_flow_limit_singles3.mat'); % load in
    steady-state model gains
24 model_input_means2 = load('model_input_means_orig14_no_flow_limit_singles3.mat'); %
    load in the required mean data for the model
25
26 % case 1 data means
27 T_ua_mean1 = model_input_means1.model_input_means(1);
28 T_room_diff_mean1 = model_input_means1.model_input_means(2:11);
29 T_AHU_mean1 = model_input_means1.model_input_means(12);
30 P_EDS_mean1 = model_input_means1.model_input_means(13);
31 T_room11_mean1 = model_input_means1.model_input_means(14);
32 Rh_ua_mean1 = model_input_means1.model_input_means(15);
33 T_room_mean1 = model_input_means1.model_input_means(16:26);
34 T_room_stpt_mean1 = model_input_means1.model_input_means(27:36);
35
36 % case 2 data means
37 T_ua_mean2 = model_input_means2.model_input_means(1);
38 T_AHU_mean2 = model_input_means2.model_input_means(2);
39 P_EDS_mean2 = model_input_means2.model_input_means(3);

```

```

40 T_room11_mean2 = model_input_means2.model_input_means(4);
41 Rh_oa_mean2 = model_input_means2.model_input_means(5);
42 T_room_mean2 = model_input_means2.model_input_means(6:16);
43
44 n_rooms = 10; % number of rooms
45 n_pred = 1; % number of predictions; 1 for steady-state
46 T_AHU_ref = 12.7; % initial discharge air temp
47 PEDS = 1.2; % initial end static pressure
48
49 % deadband temperatures for EnergyPlus thermostat
50 T_rooms_hot_ref = 10;
51 T_rooms_cold_ref = 32;
52
53 V_rooms_ref = 0.2*ones(1,10); % initial zone air flows
54 Occ_rooms_ref = ones(1,10); % initialize zone occupancy
55
56 % maximum vav flows for each zone; data taken from fits of the UBO data
57 max_vav_flow = [1.115440149895995e+03, 2.974417623670221e+02, 5.527560040379480e+02,
58               1.643509963462346e+03, 6.786019098116361e+02,...
59               9.099007814042144e+02, 1.100950429401797e+03, 4.630002354211065e+02,
60               3.622430123549660e+03, 1.159043475457241e+03]./2118.88;
61
62 pyear = [50000, 50000, 50000, 50000, 50000, 50000, 50000, 50000, 50000, 50000]; %
63         annual salary for each zone
64 hours = 24*2; % number of hours to simulate
65
66 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
67 % Controller Parameters
68
69 % Cooling loop parameters
70 kp_clg = [20,20,20,20,20,20,20,20,20,20];
71 ki_clg = [1,1,1,1,1,1,1,1,1,1];
72 kd_clg = [10,10,10,10,10,10,10,10,10,10];
73 lterm_clg = zeros(1,10);
74 outMax_clg = 100;
75 outMin_clg = 0;
76 bias_clg = 0;
77 control_act_clg = 0; % reverse acting
78 lastInput_clg = zeros(1,10); % initial value
79 setpoint_clg = 23*ones(1,10); % initial value
80
81 % Flow loop parameters
82 kp_flow = [0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5];
83 ki_flow = [0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15];
84 kd_flow = [0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1];
85 lterm_flow = zeros(1,10);
86 outMax_flow = 1;
87 outMin_flow = 0;
88 bias_flow = 0;
89 control_act_flow = 1; % normal acting
90 lastInput_flow = zeros(1,10); % initial value
91 ud = zeros(10,1); % initial damper positions
92
93 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94 % Create an mlepProcess instance and configure it
95 ep = mlepProcess;
96 ep.arguments = {'UBO_final_edition', ...
97               'USA_TX_College.Station-Easterwood.Field.722445_TMY3'};
98 ep.acceptTimeout = 30000;
99 VERNUMBER = 2;
100
101                                     % version
102         number of communication protocol (2 for E+ 7.2.0)
103 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

99 % Start EnergyPlus cosimulation
100 [status, msg] = ep.start;
101 if status ~= 0
102     error('Could not start EnergyPlus: %s.', msg);
103 end
104
105 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
106 % The main simulation loop
107 kStep = 1; % initiliaze simulation step
108 deltaT = 60; % seconds in a time step
109 steps_in_hour = 60; % number of time steps in an hour
110 timestep_OPT = 15; % time step in minutes for the supervisory controllers
111 timestep_PID = 1; % time step in minutes for the local control
112 MAXSTEPS = steps_in_hour*hours; % total time steps (12 / hr, 24 hrs, 5 days)
113
114 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
115 % Output Variables
116 EP_T_oa = zeros(MAXSTEPS,1); % outdoor air temperature
117 EP_Rh_oa = zeros(MAXSTEPS,1); % outdoor relative humidity
118 EP_T_rooms = zeros(MAXSTEPS,11); % zone temperatures
119 EP_dmpr_rooms = zeros(MAXSTEPS,11); % zone damper positions
120 EP_Rh_rooms = zeros(MAXSTEPS,11); % zone relative humidities
121 EP_mdot_rooms = zeros(MAXSTEPS,11); % zone air mass flows
122 EP_T_AHU_out = zeros(MAXSTEPS,1); % AHU discharge air temperature
123 EP_V_AHU_out = zeros(MAXSTEPS,1); % AHU air volume flow
124 EP_T_AHU_CW_flow = zeros(MAXSTEPS,1); % AHU chw mass flow
125 EP_T_AHU_CW_supp = zeros(MAXSTEPS,1); % AHU chw supply temperature
126 EP_T_AHU_CW_ret = zeros(MAXSTEPS,1); % AHU chw return temperature
127 EP_V_rooms = zeros(MAXSTEPS,11); % zone air volume flows
128 error_troom = zeros(10,1); % error between zone predictions and measures
129
130 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
131 % Optimization Variables
132 options = optimset('Display','iter','DiffMinChange',0,'Algorithm','interior-point',
133     ...
134     'GradObj','off'); % options for fmincon 'DiffMinChange',0.05,
135
136 % initial values for the optimization
137 T_room_stpt_0 = 29.44*ones(1,10);
138 T_AHU_stpt_0 = 12.7;
139 P_EDS_stpt_0 = 1.2;
140 setpoints_0 = [T_room_stpt_0, T_AHU_stpt_0, P_EDS_stpt_0];
141 setpoints = [T_room_stpt_0, T_AHU_stpt_0, P_EDS_stpt_0];
142 flows_maxed = 0;
143
144 % optimization variable limits
145 % 1:10 = zone temperatures
146 % 11 = AHU discharge air temp
147 % 12 = AHU end static pressure
148 lb = [18,18,18,18,18,18,18,18,18,18,10,0.2];
149 ub = [28,28,28,28,28,28,28,28,28,28,20,1.8];
150
151 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
152 % Run the EnergyPlus simulation
153 while kStep <= MAXSTEPS
154     % Display progress
155     percent = sprintf('%2f', kStep/MAXSTEPS*100);    progress = strcat(percent
156         , '%');    disp(progress);
157     % Read a data packet from E+
158     packet = ep.read;
159     if isempty(packet)
160         error('Could not read outputs from E+.');
161     end

```

```

161 % Parse it to obtain building outputs
162 [flag, eptime, outputs] = mlepDecodePacket(packet);
163 if flag ~= 0, break; end
164 disp(kStep)
165 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
166 % Parse data from Energy+
167
168 EP_T_ua(kStep,:) = outputs(1);
169 EP_Rh_ua(kStep,:) = outputs(2);
170 EP_T_rooms(kStep,:) = outputs(3:13);
171 EP_dmpr_rooms(kStep,:) = outputs(14:24);
172 EP_Rh_rooms(kStep,:) = outputs(25:35);
173 EP_mdot_rooms(kStep,:) = outputs(36:46);
174 EP_T_AHU_out(kStep,:) = outputs(47);
175 EP_V_AHU_out(kStep,:) = outputs(48);
176 EP_T_AHU_CW_flow(kStep,:) = outputs(49); % mass flow rate
177 EP_T_AHU_CW_supp(kStep,:) = outputs(50);
178 EP_T_AHU_CW_ret(kStep,:) = outputs(51);
179 EP_V_rooms(kStep,:) = outputs(52:62);
180
181 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
182 % Execute controllers
183
184 Occ_rooms_ref = zone_people(kStep,1:10); % set occupancy for this simulation
      step
185
186 % build case 1 and case 2 input arrays
187 model_input1 = [EP_T_ua(kStep,1), T_room_stpt_0, EP_T_rooms(kStep,11), EP_T_AHU_out
      (kStep,1), -P_EDS_stpt_0, EP_Rh_ua(kStep,1), ...
188      T_ua_mean1, T_room_stpt_mean1, T_room11_mean1, T_AHU_mean1,
      P_EDS_mean1, Rh_ua_mean1]';
189 model_input2 = [EP_T_ua(kStep,1), T_AHU_stpt_0, -P_EDS_stpt_0, EP_T_rooms(kStep,11)
      , EP_Rh_ua(kStep,1), ...
190      T_ua_mean2, T_AHU_mean2, P_EDS_mean2, T_room11_mean2, Rh_ua_mean2]';
191
192 % if either the first timestep or if the timestep is a multiple of the
193 % optimization step, perform optimization
194 if mod(kStep*timestep_PID, timestep_OPT) == 0 || kStep == 1
195
196     for z=1:n_rooms
197         % if room is occupied, determine PMV optimal room setpoint
198         if Occ_rooms_ref(1,z) == 1
199             x0 = [18,26];
200             f3 = @(ta) find_Ta_for_zero_PMV(ta, EP_Rh_rooms(kStep,z)/100);
201             [x, fval] = fminbnd(f3, x0(1), x0(2));
202
203             T_room_stpt(1,z) = x;
204
205             x0 = [18,26];
206             f4 = @(ta) find_Ta_for_zero_PMV_neg(ta, EP_Rh_rooms(kStep,z)/100);
207             [x1, fval] = fminbnd(f4, x0(1), x0(2));
208             T_room_stpt_neg(1,z) = x1;
209             % if room is unoccupied, set room to setback temperature
210             else
211                 T_room_stpt(1,z) = 29.44;
212                 T_room_stpt_neg(1,z) = 29.44;
213             end
214         end
215
216         % initialize zone prediction temperatures if first timestep or
217         % beginning of occupied time period
218         if kStep == 1 || rem(kStep,1440) == 420
219             error_troom = zeros(10,1);
220             error_troom1 = zeros(10,1);

```

```

221         error_troom_pred(kStep,:) = error_troom;
222     % else, determine zone prediction errors
223     else
224         error_troom = [EP_T_rooms(kStep,1:10)]' - T_room22;
225         error_troom1 = [EP_T_rooms(kStep,1:10)]' - T_room21;
226         error_troom_pred(kStep/timestep_OPT*timestep_PID+1,:) = error_troom;
227     end
228
229     % if beginning of occupied time period, reset last optimization
230     % variable values
231     if rem(kStep,1440) == 420
232         last_PEDS = 0.8;
233         last_TAHU = 12;
234         last_Tstpts = 22.5*ones(1,10);
235     end
236
237     % Function to be minimized by optimization
238     f1 = @(x) determine_total_cost(x, model_SS1.model_SS, model_SS2.model_SS,
        model_input1, model_input2, EP_Rh_rooms(kStep,1:10)/100, n_rooms, pyear,
        ...
        EP_T_AHU_CW_flow(kStep,1), EP_T_AHU_CW_supp(kStep,1), EP_T_AHU_CW_ret(
        kStep,1), ud, T_room_mean1(1:10),...
        T_room_mean2(1:10), T_room_stpt, Occ_rooms_ref, timestep_OPT,
        error_troom, error_troom1, transition2, trans_error2, EP_T_rooms(
        kStep,1:11));
241
242     % Nonlinear constraint function
243     f2 = @(x) find_opt_constraints(x,ud,n_rooms,n_pred,last_TAHU,last_PEDS,
        last_Tstpts);
244
245     % if the beginning of an occupied time period during a day, reset
246     % the zone temperature setpoints
247     if rem(kStep,1440) == 420
248         setpoints_0 = [22.5*ones(1,10),12,0.8]; % 22.5*ones(1,10),12,0.8
249     end
250
251     % if the current timestep is between the occupied hours (7 AM to 6
252     % PM), then perform optimization
253     if rem((kStep)/steps_in_hour,24) >= 7 && rem((kStep)/steps_in_hour,24) < 18
254         [setpoints,fval] = fmincon(f1,setpoints_0,[],[],[],[],lb,ub,f2,options);
255     else
256         setpoints(1:10) = 29.44;
257         setpoints(11) = 18;
258         setpoints(12) = 0;
259         fval = 0;
260     end
261
262     % record values for next optimization
263     last_Tstpts = setpoints(1:10);
264     last_TAHU = setpoints(11);
265     last_PEDS = setpoints(12);
266
267     T_room_stpt_0 = setpoints(1:10);
268
269     for i=1:n_rooms % overwrite room setpoint if unoccupied
270         if Occ_rooms_ref(1,i) ~= 1
271             T_room_stpt_0(1,i) = 29.44;
272         end
273     end
274
275     % assign optimized variables to coder friendly variables
276     T_AHU_stpt_0 = setpoints(11);
277     P_EDS_stpt_0 = setpoints(12);
278

```

```

279 % calculate airflow fractions to send to E+
280 V_rooms_ref_pred = vavs_pred./(2118.88);
281
282 % update model inputs with current variables to determine values
283 % from the last step of the optimization
284 model_input1 = [EP_T_oa(kStep,1), T_room_stpt_0, EP_T_rooms(kStep,11),
                T_AHU_stpt_0, -P_EDS_stpt_0, EP_Rh_oa(kStep,1), ...
285                T_oa_mean1, T_room_stpt_mean1, T_room11_mean1, T_AHU_mean1, P_EDS_mean1,
                Rh_oa_mean1]';
286 model_input2 = [EP_T_oa(kStep,1), T_AHU_stpt_0, -P_EDS_stpt_0, EP_T_rooms(kStep
                ,11), EP_Rh_oa(kStep,1), ...
287                T_oa_mean2, T_AHU_mean2, P_EDS_mean2, T_room11_mean2, Rh_oa_mean2]';
288
289 % loop over zones determine zone air temp predictions
290 i = 1;
291 while i < n_rooms+1
292
293     % compute room temperatures with steady-state model for case 1
294     T_room31(i,1) = model_SS1.model_SS(i,:) * model_input1 + T_room_mean1(1,i)
                ';
295     T_room21(i,1) = model_SS1.model_SS(i,:) * model_input1 + T_room_mean1(1,i)
                ';
296
297     % determine the case 2 zone temperature predictions
298     switch i
299     case 1
300         T_room32(i,1) = case2_room1(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, model_input2(5), EP_T_rooms(6), EP_T_rooms(2))
                + error_troom(i,1);
301         T_room22(i,1) = case2_room1(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, model_input2(5), EP_T_rooms(6), EP_T_rooms(2))
                ;
302     case 2
303         T_room32(i,1) = case2_room2(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, model_input2(5), EP_T_rooms(8), EP_T_rooms(4))
                + error_troom(i,1);
304         T_room22(i,1) = case2_room2(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, model_input2(5), EP_T_rooms(8), EP_T_rooms(4))
                ;
305     case 3
306         T_room32(i,1) = case2_room3(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, EP_T_rooms(5)) + error_troom(i,1);
307         T_room22(i,1) = case2_room3(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, EP_T_rooms(5)) ;
308     case 4
309         T_room32(i,1) = case2_room4(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, model_input2(5), EP_T_rooms(5), EP_T_rooms(7))
                + error_troom(i,1);
310         T_room22(i,1) = case2_room4(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, model_input2(5), EP_T_rooms(5), EP_T_rooms(7))
                ;
311     case 5
312         T_room32(i,1) = case2_room5(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, model_input2(5)) + error_troom(i,1);
313         T_room22(i,1) = case2_room5(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, model_input2(5)) ;
314     case 6
315         T_room32(i,1) = case2_room6(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, EP_T_rooms(5)) + error_troom(i,1);
316         T_room22(i,1) = case2_room6(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, EP_T_rooms(5)) ;
317     case 7
318         T_room32(i,1) = case2_room7(model_input2(1), T_AHU_stpt_0, -
                P_EDS_stpt_0, model_input2(5)) + error_troom(i,1);

```

```

319         T_room22(i,1) = case2_room7(model_input2(1), T_AHU_stpt_0, -
320             P_EDS_stpt_0, model_input2(5)) ;
321     case 8
322         T_room32(i,1) = case2_room8(model_input2(1), T_AHU_stpt_0, -
323             P_EDS_stpt_0, model_input2(5), EP_T_rooms(10), EP_T_rooms(2)
324             ) + error_troom(i,1);
325         T_room22(i,1) = case2_room8(model_input2(1), T_AHU_stpt_0, -
326             P_EDS_stpt_0, model_input2(5), EP_T_rooms(10), EP_T_rooms(2)
327             ) ;
328     case 9
329         T_room32(i,1) = case2_room9(model_input2(1), T_AHU_stpt_0, -
330             P_EDS_stpt_0, model_input2(5), EP_T_rooms(7)) + error_troom(
331             i,1);
332         T_room22(i,1) = case2_room9(model_input2(1), T_AHU_stpt_0, -
333             P_EDS_stpt_0, model_input2(5), EP_T_rooms(7)) ;
334     case 10
335         T_room32(i,1) = case2_room10(model_input2(1), T_AHU_stpt_0, -
336             P_EDS_stpt_0, model_input2(5), EP_T_rooms(8), EP_T_rooms(11)
337             ) + error_troom(i,1);
338         T_room22(i,1) = case2_room10(model_input2(1), T_AHU_stpt_0, -
339             P_EDS_stpt_0, model_input2(5), EP_T_rooms(8), EP_T_rooms(11)
340             ) ;
341     end
342     % logic for determining if case 1 or case 2
343     if T_room31(i,1) > T_room22(i,1) && ud(i) < 95% % case 1
344         T_room3(i,1) = T_room31(i,1) + error_troom1(i,1);
345         T_room2(i,1) = T_room21(i,1);
346     else % case 2
347         T_room3(i,1) = T_room32(i,1);
348         T_room2(i,1) = T_room22(i,1);
349     end
350     i = i + 1;
351 end
352
353 % assign setpoints for control loops and next optimization
354 setpoints_0 = [T_room_stpt_0, T_AHU_stpt_0, P_EDS_stpt_0];
355
356 for i=1:1:n_rooms % determine the comfort cost for each room
357     if Occ_rooms_ref(1,i) == 1
358         [J_room_all3(1,i), PMV_all3(1,i), error_all3(1,i), dLOP_dpmv_all3(1,i),
359             R_all3(1,i)] = ...
360             room_cost(timestep_OPT, pyear(i), T_room3(i,1), EP_Rh_rooms(kStep,i)
361                 /100, T_room_stpt(1,i), T_room3(i,1));
362     else
363         J_room_all3(1,i) = 0;
364         PMV_all3(1,i) = NaN;
365         error_all3(1,i) = NaN;
366         dLOP_dpmv_all3(1,i) = NaN;
367         R_all3(1,i) = NaN;
368     end
369 end
370
371 % determine chw and fan costs
372 chw_flow = 3.949579252219686; % maximum mass flow rate EP_T_AHU_CW_flow(kStep,1)
373 [J_chw_pred] = chw_cost(T_AHU_stpt_0, chw_flow, EP_T_AHU_CW_supp(kStep,1),
374     EP_T_AHU_CW_ret(kStep,1), timestep_OPT); % determine the cost of chilled
375     water
376 [J_fan_pred] = fan_cost (ud, P_EDS_stpt_0, timestep_OPT); % determine the cost
377     of fan power
378 end
379
380 % simulate zone vav controllers to determine air flow fractions to send to
381 % EnergyPlus

```

```

366 for i=1:1:10 %%%%% room vav controllers %%%%%
367 % reset I terms of cooling PID controller is at beginning or end of
368 % occupied time period
369 if rem((kStep)/steps_in_hour,24) == 7
370     Iterm_clg(1,i) = 0;
371 elseif rem((kStep)/steps_in_hour,24) == 18 || kStep == 1
372     Iterm_clg(1,i) = 0;
373 end
374
375 % Cooling loop controller
376 setpoint_clg = T_room_stpt_0;
377 input_clg = EP_T_rooms(kStep,i);
378 [output_clg, lastInput_clg(1,i), Iterm_clg(1,i)] = pid_control(setpoint_clg(1,i)
    , input_clg, ...
379     lastInput_clg(1,i), Iterm_clg(1,i), kp_clg(i),...
380     ki_clg(i), kd_clg(i), outMax_clg, outMin_clg, bias_clg, control_act_clg,
        timestep_PID);
381
382 % reset I terms of cooling PID controller is at beginning or end of
383 % occupied time period
384 if rem((kStep)/steps_in_hour,24) == 7
385     Iterm_flow(1,i) = 0;
386 elseif rem((kStep)/steps_in_hour,24) == 18 || kStep == 1
387     Iterm_flow(1,i) = 0;
388 end
389
390 % Flow loop controller
391 setpoint_flow = output_clg/100;
392 input_flow = EP_V_rooms(kStep,i)/max_vav_flow(1,i);
393
394 [output_flow, lastInput_flow(1,i), Iterm_flow(1,i)] = pid_control(setpoint_flow,
    input_flow, ...
395     lastInput_flow(1,i), Iterm_flow(1,i), kp_flow(i),...
396     ki_flow(i), kd_flow(i), outMax_flow, outMin_flow, bias_flow, control_act_flow
        , timestep_PID);
397 end
398
399 % convert controller outputs to damper percents
400 ud = [output_flow_data(kStep,:) * 100]';
401
402 % determine individual VAV flows
403 vav1_flow = vav1(ud(1,1), P_EDS_stpt_0, []);
404 vav2_flow = vav2(ud(2,1), P_EDS_stpt_0, []);
405 vav3_flow = vav3(ud(3,1), P_EDS_stpt_0, []);
406 vav4_flow = vav4(ud(4,1), P_EDS_stpt_0, []);
407 vav5_flow = vav5(ud(5,1), P_EDS_stpt_0, []);
408 vav6_flow = vav6(ud(6,1), P_EDS_stpt_0, []);
409 vav7_flow = vav7(ud(7,1), P_EDS_stpt_0, []);
410 vav8_flow = vav8(ud(8,1), P_EDS_stpt_0, []);
411 vav9_flow = vav9(ud(9,1), P_EDS_stpt_0, []);
412 vav11_flow = vav11(ud(10,1), P_EDS_stpt_0, []);
413
414 vavs = [vav1_flow, vav2_flow, vav3_flow, vav4_flow, vav5_flow, vav6_flow, ...
415     vav7_flow, vav8_flow, vav9_flow, vav11_flow];
416 vavs_flow = sum(vavs);
417
418 % convert VAV flows to VAV air flow fractions
419 V_rooms_ref = vavs./(max_vav_flow*2118.88);
420
421 for i=1:1:n_rooms % determine the comfort cost for each room
422     if Occ_rooms_ref(1,i) == 1
423         [J_room_all(1,i), PMV_all(1,i), error_all(1,i), dLOP_dpmv_all(1,i),
            R_all(1,i)] = ...
424             room_cost(timestep_OPT, pyear(i), EP_T_rooms(kStep,i), EP_Rh_rooms(

```

```

425         kStep,i)/100, T_room_stpt(1,i), EP_T_rooms(kStep,i));
426     else
427         J_room_all(1,i) = 0;
428     end
429 end
430
431 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
432 % Set Points
433
434 SP = [T_AHU_stpt_0, T_rooms_hot_ref, T_rooms_cold_ref, V_rooms_ref,
435       Occ_rooms_ref]; % setpoints passed to energy plus
436
437 ep.write(mlepEncodeRealData(VERNUMBER, 0, (kStep-1)*deltaT, SP));
438
439 kStep = kStep + 1;
440 end
441
442 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
443 % Stop EnergyPlus
444 ep.stop;
445 disp(['Stopped with flag ' num2str(flag)]);
446
447 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
448 %
449 % =====FLAGS=====
450 % Flag Description
451 % +1 Simulation reached end time.
452 % 0 Normal operation.
453 % -1 Simulation terminated due to an unspecified error.
454 % -10 Simulation terminated due to an error during the initialization.
455 % -20 Simulation terminated due to an error during the time integration.
456 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Objective function file for steady-state method.

```

1 function [J_total] = determine_total_cost(x,model_SS1,model_SS2,model_input1,
2     model_input2,Rh_rooms,n_rooms,pyear,chw_flow,...
3     T_supp,T_ret,ud,T_room_mean1,T_room_mean2,T_room_stpt,Occ_rooms_ref,
4     timestep_OPT,error_troom,error_troom1,transition2,...
5     trans_error2,EP_T_rooms)
6
7 % This function is used in the steady-state method's optimization as the
8 % cost function to be minimized by fmincon.
9
10
11 % Assign the optimization variables to coder friendly variables
12 T_DA_stpt = x(11);
13 P_EDS_stpt = x(12);
14
15
16 % Assign the proper variables to the case 1 model input
17 model_input1(2:11) = x(1:10);
18 model_input1(13) = T_DA_stpt;
19 model_input1(14) = -P_EDS_stpt;
20
21 % Assign the proper variables to the case 2 model input
22 model_input2(2) = T_DA_stpt;
23 model_input2(3) = -P_EDS_stpt;
24
25
26 % Loop over the zones making zone temperature predictions
27 for i=1:1:n_rooms
28
29     % compute zone temperatures with steady-state case 1 model
30     T_room1(i,1) = model_SS1(i,:)*model_input1 + T_room_mean1(1,i)';
31
32     % calculate the case 2 zone predictions with and without the error from
33     % the previous prediction

```

```

28     switch i
29     case 1
30         T_room3(i,1) = case2_room1(model_input2(1), T_DA_stpt, -P_EDS_stpt,
31                                     model_input2(5), EP_T_rooms(6), EP_T_rooms(2)) + error_troom(i,1);
32         T_room2(i,1) = case2_room1(model_input2(1), T_DA_stpt, -P_EDS_stpt,
33                                     model_input2(5), EP_T_rooms(6), EP_T_rooms(2));
34     case 2
35         T_room3(i,1) = case2_room2(model_input2(1), T_DA_stpt, -P_EDS_stpt,
36                                     model_input2(5), EP_T_rooms(8), EP_T_rooms(4)) + error_troom(i,1);
37         T_room2(i,1) = case2_room2(model_input2(1), T_DA_stpt, -P_EDS_stpt,
38                                     model_input2(5), EP_T_rooms(8), EP_T_rooms(4));
39     case 3
40         T_room3(i,1) = case2_room3(model_input2(1), T_DA_stpt, -P_EDS_stpt,
41                                     EP_T_rooms(5)) + error_troom(i,1);
42         T_room2(i,1) = case2_room3(model_input2(1), T_DA_stpt, -P_EDS_stpt,
43                                     EP_T_rooms(5));
44     case 4
45         T_room3(i,1) = case2_room4(model_input2(1), T_DA_stpt, -P_EDS_stpt,
46                                     model_input2(5), EP_T_rooms(5), EP_T_rooms(7)) + error_troom(i,1);
47         T_room2(i,1) = case2_room4(model_input2(1), T_DA_stpt, -P_EDS_stpt,
48                                     model_input2(5), EP_T_rooms(5), EP_T_rooms(7));
49     case 5
50         T_room3(i,1) = case2_room5(model_input2(1), T_DA_stpt, -P_EDS_stpt,
51                                     model_input2(5)) + error_troom(i,1);
52         T_room2(i,1) = case2_room5(model_input2(1), T_DA_stpt, -P_EDS_stpt,
53                                     model_input2(5));
54     case 6
55         T_room3(i,1) = case2_room6(model_input2(1), T_DA_stpt, -P_EDS_stpt,
56                                     EP_T_rooms(5)) + error_troom(i,1);
57         T_room2(i,1) = case2_room6(model_input2(1), T_DA_stpt, -P_EDS_stpt,
58                                     EP_T_rooms(5));
59     case 7
60         T_room3(i,1) = case2_room7(model_input2(1), T_DA_stpt, -P_EDS_stpt,
61                                     model_input2(5)) + error_troom(i,1);
62         T_room2(i,1) = case2_room7(model_input2(1), T_DA_stpt, -P_EDS_stpt,
63                                     model_input2(5));
64     case 8
65         T_room3(i,1) = case2_room8(model_input2(1), T_DA_stpt, -P_EDS_stpt,
66                                     model_input2(5), EP_T_rooms(10), EP_T_rooms(2)) + error_troom(i,1);
67         T_room2(i,1) = case2_room8(model_input2(1), T_DA_stpt, -P_EDS_stpt,
68                                     model_input2(5), EP_T_rooms(10), EP_T_rooms(2));
69     case 9
70         T_room3(i,1) = case2_room9(model_input2(1), T_DA_stpt, -P_EDS_stpt,
71                                     model_input2(5), EP_T_rooms(7)) + error_troom(i,1);
72         T_room2(i,1) = case2_room9(model_input2(1), T_DA_stpt, -P_EDS_stpt,
73                                     model_input2(5), EP_T_rooms(7));
74     case 10
75         T_room3(i,1) = case2_room10(model_input2(1), T_DA_stpt, -P_EDS_stpt,
76                                     model_input2(5), EP_T_rooms(8), EP_T_rooms(11)) + error_troom(i,1);
77         T_room2(i,1) = case2_room10(model_input2(1), T_DA_stpt, -P_EDS_stpt,
78                                     model_input2(5), EP_T_rooms(8), EP_T_rooms(11));
79     end
80
81     % logic to determine if a zone should use the case 1 prediction or case
82     % 2 prediction
83     if T_room1(i,1) > T_room2(i,1) && ud(i) < 95
84         T_room(i,1) = T_room1(i,1) + error_troom1(i,1);
85         case_history(i,1) = 0;
86     else
87         T_room(i,1) = T_room3(i,1);
88         case_history(i,1) = 1;
89     end
90
91     %calculate the discomfort cost for each zone

```

```

72 % if a zone is unoccupied, make the discomfort cost zero
73 if Occ_rooms_ref(1,i) == 1
74     [J_room_all(1,i), PMV_all(1,i), error_all(1,i), dLOP_dpmv_all(1,i), Rcalc(1,
75         i)] = ...
76         room_cost(timestep_OPT, pyear(i), T_room(i,1), Rh_rooms(1,i),
77             T_room_stpt(1,i), T_room(i,1));
78 else
79     J_room_all(1,i) = 0;
80 end
81 % calculate the chilled water and fan costs
82 chw_flow = 3.949579252219686; % maximum mass flow rate
83 [J_chw] = chw_cost(T_DA_stpt, chw_flow, T_supp, T_ret, timestep_OPT); % determine
84     the cost of chilled water
85 [J_fan] = fan_cost(ud, P_EDS_stpt, timestep_OPT); % determine the cost of fan power
86 J_total = sum(J_room_all) + J_chw + 0*J_fan; % determine the total cost, which is to
87     be minimized
end

```

Case 2, zone 1 file for steady-state method.

```

1 function [t_pred] = case2_room1(Toa, TAHU, PEDS, Rhoa, Troom6, Troom2)
2 % Inputs:
3 % Toa TAHU PEDS Rhoa Troom6 Troom2
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2700277777784];
8 PEDS_mean = 0;
9 Rhoa_mean = [70.1008597883598];
10 Troom6_mean = [26.6773629298941];
11 Troom2_mean = [27.2311737764551];
12 Troom1_mean = [27.4242157738095];
13
14 % subtract means from data
15 model_input = [Toa - Toa_mean;
16     TAHU - TAHU_mean;
17     PEDS - PEDS_mean;
18     Rhoa - Rhoa_mean;
19     Troom6 - Troom6_mean;
20     Troom2 - Troom2_mean];
21
22 % modelSS = [0.3129 0.0137 5.3496 0.0137 0.3562 0.1634];
23 % modified B21 matrix from identified model that gives better response;
24 modelSS = [0.888, 1.5028, 5.8745, 0.0137, 0.1366, 0.1366];
25
26 % prediction temperature
27 t_pred = modelSS*model_input + Troom1_mean;
28 end

```

Case 2, zone 2 file for steady-state method.

```

1 function [t_pred] = case2_room2(Toa, TAHU, PEDS, Rhoa, Troom8, Troom4)
2 % Inputs:
3 % Toa TAHU PEDS Rhoa Troom8 Troom4
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2700927579371];
8 PEDS_mean = 0;
9 Rhoa_mean = [70.1008597883598];
10 Troom8_mean = [27.1901698082009];

```

```

11 Troom4_mean = [25.6683030753968];
12 Troom2_mean = [27.0163521825397];
13
14 % subtract means from data
15 model_input = [Toa - Toa_mean;
16               TAHU - TAHU_mean;
17               PEDS - PEDS_mean;
18               Rhoa - Rhoa_mean;
19               Troom8 - Troom8_mean;
20               Troom4 - Troom4_mean];
21
22 % identified steady-state gains from model
23 modelSS = [0.2174    0.0269    4.9684    0.0102    0.0105    0.5252];
24
25 % prediction temperature
26 t_pred = modelSS*model_input + Troom2_mean;
27 end

```

Case 2, zone 3 file for steady-state method.

```

1 function [t_pred] = case2_room3(Toa, TAHU, PEDS, Troom5)
2 % Inputs:
3 % Toa TAHU PEDS Troom5
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2700502645509];
8 PEDS_mean = 0;
9 Troom5_mean = [26.3665218253968];
10 Troom3_mean = [26.2664608134921];
11
12 % subtract means from data
13 model_input = [Toa - Toa_mean;
14               TAHU - TAHU_mean;
15               PEDS - PEDS_mean;
16               Troom5 - Troom5_mean];
17
18 % identified steady-state gains from model
19 modelSS = [0.0018    2.3318    5.6130    0.0740];
20
21 % prediction temperature
22 t_pred = modelSS*model_input + Troom3_mean;
23 end

```

Case 2, zone 4 file for steady-state method.

```

1 function [t_pred] = case2_room4(Toa, TAHU, PEDS, Rhoa, Troom5, Troom7)
2 % Inputs:
3 % Toa TAHU PEDS Rhoa Troom6 Troom2
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2727337962969];
8 PEDS_mean = 0;
9 Rhoa_mean = [70.1008597883598];
10 Troom5_mean = [26.1843720238096];
11 Troom7_mean = [27.7279527116402];
12 Troom4_mean = [23.7602959656084];
13
14 % subtract means from data
15 model_input = [Toa - Toa_mean;
16               TAHU - TAHU_mean;
17               PEDS - PEDS_mean;
18               Rhoa - Rhoa_mean;

```

```

19         Troom5 = Troom5_mean;
20         Troom7 = Troom7_mean];
21
22 % identified steady-state gains from model
23 modelSS = [0.0200    0.4905    4.9555    0.0200    0.5453    0.1393];
24
25 % prediction temperature
26 t_pred = modelSS*model_input + Troom4_mean;
27 end

```

Case 2, zone 5 file for steady-state method.

```

1 function [t_pred] = case2_room5(Toa, TAHU, PEDS, Rhoa)
2 % Inputs:
3 % Toa TAHU PEDS Rhoa Troom6 Troom2
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2700042989424];
8 PEDS_mean = 0;
9 Rhoa_mean = [70.1008597883598];
10 Troom5_mean = [25.3901630291006];
11
12 % subtract means from data
13 model_input = [Toa - Toa_mean;
14               TAHU - TAHU_mean;
15               PEDS - PEDS_mean;
16               Rhoa - Rhoa_mean];
17
18 % identified steady-state gains from model
19 modelSS = [0.0325    0.8651    2.8874    0.0146];
20
21 % prediction temperature
22 t_pred = modelSS*model_input + Troom5_mean;
23 end

```

Case 2, zone 6 file for steady-state method.

```

1 function [t_pred] = case2_room6(Toa, TAHU, PEDS, Troom5)
2 % Inputs:
3 % Toa TAHU PEDS Rhoa Troom6 Troom2
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2699890873022];
8 PEDS_mean = 0;
9 Troom5_mean = [26.2868361441801];
10 Troom6_mean = [25.6869740410053];
11
12 % subtract means from data
13 model_input = [Toa - Toa_mean;
14               TAHU - TAHU_mean;
15               PEDS - PEDS_mean;
16               Troom5 - Troom5_mean];
17
18 % identified steady-state gains from model
19 modelSS = [0.0088    0.4657    3.4026    0.0920];
20
21 % prediction temperature
22 t_pred = modelSS*model_input + Troom6_mean;
23 end

```

Case 2, zone 7 file for steady-state method.

```

1 function [t_pred] = case2_room7(Toa, TAHU, PEDS, Rhoa)
2 % Inputs:
3 % Toa TAHU PEDS Rhoa Troom6 Troom2
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2705529100535];
8 PEDS_mean = 0;
9 Rhoa_mean = [70.1008597883598];
10 Troom7_mean = [26.8028381283069];
11
12 % subtract means from data
13 model_input = [Toa - Toa_mean;
14                TAHU - TAHU_mean;
15                PEDS - PEDS_mean;
16                Rhoa - Rhoa_mean];
17
18 % identified steady-state gains from model
19 modelSS = [0.0023    0.0344    12.7866    0.0046];
20
21 % prediction temperature
22 t_pred = modelSS*model_input + Troom7_mean;
23 end

```

Case 2, zone 8 file for steady-state method.

```

1 function [t_pred] = case2_room8(Toa, TAHU, PEDS, Rhoa, Troom10, Troom2)
2 % Inputs:
3 % Toa TAHU PEDS Rhoa Troom6 Troom2
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2700833333339];
8 PEDS_mean = 0;
9 Rhoa_mean = [70.1008597883598];
10 Troom10_mean = [28.3942471891534];
11 Troom2_mean = [27.3701263227513];
12 Troom8_mean = [26.8765838293651];
13
14 % subtract means from data
15 model_input = [Toa - Toa_mean;
16                TAHU - TAHU_mean;
17                PEDS - PEDS_mean;
18                Rhoa - Rhoa_mean;
19                Troom10 - Troom10_mean;
20                Troom2 - Troom2_mean];
21
22 % identified steady-state gains from model
23 modelSS = [0.0035    0.1089    5.7648    0.0057    0.6800    0.0045];
24
25 % prediction temperature
26 t_pred = modelSS*model_input + Troom8_mean;
27 end

```

Case 2, zone 9 file for steady-state method.

```

1 function [t_pred] = case2_room9(Toa, TAHU, PEDS, Rhoa, Troom7)
2 % Inputs:
3 % Toa TAHU PEDS Rhoa Troom6 Troom2
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2761008597890];

```

```

8 PEDS_mean = 0;
9 Rhoa_mean = [70.1008597883598];
10 Troom7_mean = [28.3942471891534];
11 Troom9_mean = [24.3699181547618];
12
13 % subtract means from data
14 model_input = [Toa - Toa_mean;
15                TAHU - TAHU_mean;
16                PEDS - PEDS_mean;
17                Rhoa - Rhoa_mean;
18                Troom7 - Troom7_mean];
19
20 % identified steady-state gains from model
21 modelSS = [0.2638    0.7468    2.8482    0.0588    0.0371];
22
23 % prediction temperature
24 t_pred = modelSS*model_input + Troom9_mean;
25 end

```

Case 2, zone 10 file for steady-state method.

```

1 function [t_pred] = case2_room10(Toa, TAHU, PEDS, Rhoa, Troom8, Troom11)
2 % Inputs:
3 % Toa TAHU PEDS Rhoa Troom6 Troom2
4
5 % means from identified data
6 Toa_mean = [28.5247023809524];
7 TAHU_mean = [17.2700249669319];
8 PEDS_mean = 0;
9 Rhoa_mean = [70.1008597883598];
10 Troom8_mean = [27.1852526455027];
11 Troom11_mean = [23.0571990740739];
12 Troom10_mean = [28.0829950396826];
13
14 % subtract means from data
15 model_input = [Toa - Toa_mean;
16                TAHU - TAHU_mean;
17                PEDS - PEDS_mean;
18                Rhoa - Rhoa_mean;
19                Troom8 - Troom8_mean;
20                Troom11 - Troom11_mean];
21
22 % identified steady-state gains from model
23 modelSS = [0.0098    0.6663    1.0752    0.0081    0.0016    0.0005];
24
25 % prediction temperature
26 t_pred = modelSS*model_input + Troom10_mean;
27 end

```

E.2 Dynamic MPC Method Specific Code

Main simulation file for the dynamic MPC method.

```

1 clear all; close all;
2 % This is the main simulation file for the dynamic MPC control
3 % method.
4
5 n_pred = 6; % number of predictions to use in optimization minimization
6
7 % initialize variables
8 case_status_history = [];
9 T_room_stpt_data = [];
10 T_room_stpt_neg_data = [];

```

```

11 J_total_data = [];
12 Tpred_array_data = [];
13 J_roompred_all_array = [];
14 J_chwpred_array = [];
15 J_fanpred_array = [];
16 ud_array = [];
17 case_all_history = [];
18 Troom1_array_data = [];
19 Troom2_array_data = [];
20 grad_calc_total_data = [];
21 setpoints_raw_data = [];
22 grad_act_data = [];
23
24 % load necessary data
25 load('zone_people_1minute_data.mat'); % load in random occupancy data
26 % load('zone_people_1minute_10day_full_occ.mat'); % load in full occupancy data
27 % zone_people(:, :) = 1;
28 EP_T_ua_master = load('T_ua_master.mat'); % outdoor air temp for predictions
29 EP_Rh_ua_master = load('Rh_ua_master.mat'); % outdoor relative humidity for
    predictions
30
31 % load in the required mean data for the case 1 and case 2 models
32 model_input_means1 = load('model_input_means_orig10_no_flow_limit.mat');
33
34 room_model21 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room1.mat');
35 room_model22 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room2.mat');
36 room_model23 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room3.mat');
37 room_model24 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room4.mat');
38 room_model25 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room5.mat');
39 room_model26 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room6.mat');
40 room_model27 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room7.mat');
41 room_model28 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room8.mat');
42 room_model29 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room9.mat');
43 room_model210 = load('
    room_models_pem_figures_orig14_no_flow_limit_rand_occ3_5min_room10.mat');
44
45 model_input_means21 = room_model21.room_models{1}.means;
46 model_input_means22 = room_model22.room_models{2}.means;
47 model_input_means23 = room_model23.room_models{3}.means;
48 model_input_means24 = room_model24.room_models{4}.means;
49 model_input_means25 = room_model25.room_models{5}.means;
50 model_input_means26 = room_model26.room_models{6}.means;
51 model_input_means27 = room_model27.room_models{7}.means;
52 model_input_means28 = room_model28.room_models{8}.means;
53 model_input_means29 = room_model29.room_models{9}.means;
54 model_input_means210 = room_model210.room_models{10}.means;
55
56 model_input_means2 = {model_input_means21, model_input_means22, model_input_means23,
    model_input_means24, ...
57     model_input_means25, model_input_means26, model_input_means27,
    model_input_means28, ...
58     model_input_means29, model_input_means210};
59
60 T_ua_mean1 = model_input_means1.model_input_means(1);
61 T_room_diff_mean1 = model_input_means1.model_input_means(2:11);

```

```

62 T_AHU_mean1 = model_input_means1.model_input_means(12);
63 P_EDS_mean1 = model_input_means1.model_input_means(13);
64 T_room11_mean1 = model_input_means1.model_input_means(14);
65 Rh_oa_mean1 = model_input_means1.model_input_means(15);
66 T_room_mean1 = model_input_means1.model_input_means(16:26);
67 T_room_stpt_mean1 = model_input_means1.model_input_means(27:36);
68
69 case1_means = [T_oa_mean1,T_room_stpt_mean1,T_room11_mean1,T_AHU_mean1,P_EDS_mean1,
    Rh_oa_mean1];
70 case2_room1_means = [model_input_means21(1), model_input_means21(2),
    model_input_means21(3), model_input_means21(15), model_input_means21(3+6),
    model_input_means21(3+2)];
71 case2_room2_means = [model_input_means22(1), model_input_means22(2),
    model_input_means22(3), model_input_means22(15), model_input_means22(3+8),
    model_input_means22(3+4)];
72 case2_room3_means = [model_input_means23(1), model_input_means23(2),
    model_input_means23(3), model_input_means23(3+5)];
73 case2_room4_means = [model_input_means24(1), model_input_means24(2),
    model_input_means24(3), model_input_means24(15), model_input_means24(3+5),
    model_input_means24(3+7)];
74 case2_room5_means = [model_input_means25(1), model_input_means25(2),
    model_input_means25(3), model_input_means25(15)];
75 case2_room6_means = [model_input_means26(1), model_input_means26(2),
    model_input_means26(3), model_input_means26(3+5)];
76 case2_room7_means = [model_input_means27(1), model_input_means27(2),
    model_input_means27(3), model_input_means27(15)];
77 case2_room8_means = [model_input_means28(1), model_input_means28(2),
    model_input_means28(3), model_input_means28(15), model_input_means28(3+10),
    model_input_means28(3+2)];
78 case2_room9_means = [model_input_means29(1), model_input_means29(2),
    model_input_means29(3), model_input_means29(15), model_input_means29(3+7)];
79 case2_room10_means = [model_input_means210(1), model_input_means210(2),
    model_input_means210(3), model_input_means210(15), model_input_means210(3+8),
    model_input_means210(3+11)];
80 case2_means = {case2_room1_means,case2_room2_means,case2_room3_means,
    case2_room4_means,case2_room5_means,...
81     case2_room6_means,case2_room7_means,case2_room8_means,
    case2_room9_means,case2_room10_means};
82 T_room_mean2 = [model_input_means21(3+1), model_input_means22(3+2),
    model_input_means23(3+3), model_input_means24(3+4),...
83     model_input_means25(3+5), model_input_means26(3+6),
    model_input_means27(3+7), model_input_means28(3+8),...
84     model_input_means29(3+9), model_input_means210(3+10)];
85
86 n_rooms = 10; % number of zones
87
88 % deadband temperatures for EnergyPlus thermostat
89 T_rooms_hot_ref = 10;
90 T_rooms_cold_ref = 32;
91
92 % maximum vav flows for each zone; data taken from fits of the UBO data
93 max_vav_flow = [1.115440149895995e+03, 2.974417623670221e+02, 5.527560040379480e+02,
    1.643509963462346e+03, 6.786019098116361e+02,...
94     9.099007814042144e+02, 1.100950429401797e+03, 4.630002354211065e+02,
    3.622430123549660e+03, 1.159043475457241e+03]./2118.88;
95
96 hours = 24*2; % number of hours to simulate
97 pyear = [50000, 50000, 50000, 50000, 50000, 50000, 50000, 50000, 50000, 50000]; %
    annual salary for each zone
98
99 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
100 % Controller Parameters
101
102 % Cooling loop parameters

```

```

103 kp_clg = 20;
104 ki_clg = 1;
105 kd_clg = 10;
106 lterm_clg = zeros(1,10);
107 outMax_clg = 100;
108 outMin_clg = 0;
109 bias_clg = 0;
110 control_act_clg = 0; % reverse acting
111 lastInput_clg = zeros(1,10); % initial value
112 setpoint_clg = 23*ones(1,10); % initial value
113
114 % Flow loop parameters
115 kp_flow = 0.5;
116 ki_flow = 0.15;
117 kd_flow = 0.1;
118 lterm_flow = zeros(1,10);
119 outMax_flow = 1;
120 outMin_flow = 0;
121 bias_flow = 0;
122 control_act_flow = 1; % normal acting
123 lastInput_flow = zeros(1,10); % initial value
124 ud = zeros(10,1); % initial damper positions
125
126 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
127 % Create an mlepProcess instance and configure it
128 ep = mlepProcess;
129 ep.arguments = {'UBO_final_edition', ...
130 'USA_TX_College.Station-Easterwood.Field.722445_TMY3'};
131 ep.acceptTimeout = 20000;
132 VERNUMBER = 2;
133
134                                     % version
135                                     number of communication protocol (2 for E+ 7.2.0)
136 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
137 % Start EnergyPlus cosimulation
138 [status, msg] = ep.start;
139 if status ~= 0
140     error('Could not start EnergyPlus: %s.', msg);
141 end
142
143 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
144 % The main simulation loop
145 kStep = 1; % initiliaze simulation step
146 deltaT = 60; % seconds in a time step
147 steps_in_hour = 60; % number of time steps in an hour
148 timestep_OPT = 5; % time step in minutes for the supervisory controllers
149 timestep_PID = 1; % time step in minutes for the local control
150 MAXSTEPS = steps_in_hour*hours; % total time steps (12 / hr, 24 hrs, 5 days)
151
152 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
153 % Output Variables
154 EP_T_oa = zeros(MAXSTEPS,1); % outdoor air temperature
155 EP_Rh_oa = zeros(MAXSTEPS,1); % outdoor relative humidity
156 EP_T_rooms = zeros(MAXSTEPS,11); % zone temperatures
157 EP_dmpr_rooms = zeros(MAXSTEPS,11); % zone damper positions
158 EP_Rh_rooms = zeros(MAXSTEPS,11); % zone relative humidities
159 EP_mdot_rooms = zeros(MAXSTEPS,11); % zone air mass flows
160 EP_T_AHU_out = zeros(MAXSTEPS,1); % AHU discharge air temperature
161 EP_V_AHU_out = zeros(MAXSTEPS,1); % AHU air volume flow
162 EP_T_AHU_CW_flow = zeros(MAXSTEPS,1); % AHU chw mass flow
163 EP_T_AHU_CW_supp = zeros(MAXSTEPS,1); % AHU chw supply temperature
164 EP_T_AHU_CW_ret = zeros(MAXSTEPS,1); % AHU chw return temperature
165 EP_V_rooms = zeros(MAXSTEPS,11); % zone air volume flows
166 error_troom = zeros(10,1); % error between zone predictions and measures

```

```

165
166 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
167 % Optimization Variables
168 % options = optimset('Display','final','DiffMinChange',0.1,'Algorithm','interior-
    point',...
169 %                               'GradObj','on'); % options for fmincon
170
171 % initial values for the optimization
172 T_room_stpt_0 = 29.43*ones(1,10*n_pred);
173 T_AHU_stpt_0 = 12.7*ones(1,n_pred);
174 P_EDS_stpt_0 = 1.2*ones(1,n_pred);
175 setpoints_0 = [T_room_stpt_0,T_AHU_stpt_0,P_EDS_stpt_0];
176 setpoints_raw = [T_room_stpt_0,T_AHU_stpt_0,P_EDS_stpt_0];
177 setpoints = setpoints_0;
178
179 % optimization variable limits
180 % 20 & 28 = zone temperatures
181 % 10 & 20 = AHU discharge air temp
182 % 0.2 & 1.8 = AHU end static pressure
183 lb = [20*ones(1,10*n_pred),10*ones(1,n_pred),0.2*ones(1,n_pred)];
184 ub = [28*ones(1,10*n_pred),20*ones(1,n_pred),1.8*ones(1,n_pred)];
185
186 % initialize previous states for predictions
187 prev_states.x1 = {zeros(15,1)};
188 prev_states.x21 = {zeros(1,1)};
189 prev_states.x22 = {zeros(1,1)};
190 prev_states.x23 = {zeros(9,1)};
191 prev_states.x24 = {zeros(2,1)};
192 prev_states.x25 = {zeros(10,1)};
193 prev_states.x26 = {zeros(10,1)};
194 prev_states.x27 = {zeros(10,1)};
195 prev_states.x28 = {zeros(14,1)};
196 prev_states.x29 = {zeros(12,1)};
197 prev_states.x210 = {zeros(14,1)};
198
199 % initialize case 1 or case 2 status vector
200 case_status = zeros(1,10);
201
202 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
203 % Run the EnergyPlus simulation
204
205 while kStep <= MAXSTEPS
206     % Display progress
207     %     percent = sprintf('%2f',kStep/MAXSTEPS*100);    progress = strcat(percent
        ,'%');    disp(progress);
208     % Read a data packet from E+
209     packet = ep.read;
210     if isempty(packet)
211         error('Could not read outputs from E+.');
212     end
213     % Parse it to obtain building outputs
214     [flag, eptime, outputs] = mlepDecodePacket(packet);
215     if flag ~= 0, break; end
216     disp(kStep)
217
218     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
219     % Parse data from Energy+
220
221     EP_T_oa(kStep,:) = outputs(1);
222     EP_Rh_oa(kStep,:) = outputs(2);
223     EP_T_rooms(kStep,:) = outputs(3:13);
224     EP_dmpr_rooms(kStep,:) = outputs(14:24);
225     EP_Rh_rooms(kStep,:) = outputs(25:35);
226     EP_mdott_rooms(kStep,:) = outputs(36:46);

```

```

227 EP_T_AHU_out(kStep,:) = outputs(47);
228 EP_V_AHU_out(kStep,:) = outputs(48);
229 EP_T_AHU_CW_flow(kStep,:) = outputs(49); % mass flow rate
230 EP_T_AHU_CW_supp(kStep,:) = outputs(50);
231 EP_T_AHU_CW_ret(kStep,:) = outputs(51);
232 EP_V_rooms(kStep,:) = outputs(52:62);
233 disp(EP_T_rooms(kStep,1));
234
235 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
236 % Execute controllers
237
238 %%% Set occupancy
239 Occ_rooms_ref = zone_people(kStep,1:10); % set occupancy for this simulation
      step
240
241 % if either the first timestep or if the timestep is a multiple of the
242 % optimization step, perform optimization
243 if mod(kStep*timestep_PID,timestep_OPT) == 0 || kStep == 1
244
245     %%% If room is occupied, set setpoint to optimal PMV value
246     for z=1:1:n_rooms
247         if Occ_rooms_ref(1,z) == 1
248             x0 = [18,26];
249             f3 = @(ta)find_Ta_for_zero_PMV(ta,EP_Rh_rooms(kStep,z)/100);
250             [x,fval] = fminbnd(f3,x0(1),x0(2));
251
252             T_room_stpt(1,z) = x;
253
254             x0 = [18,26];
255             f4 = @(ta)find_Ta_for_zero_PMV_neg(ta,EP_Rh_rooms(kStep,z)/100);
256             [x1,fval] = fminbnd(f4,x0(1),x0(2));
257             T_room_stpt_neg(1,z) = x1;
258             elseif rem((kStep)/steps_in_hour,24) >= 6.5 && rem((kStep)/steps_in_hour
,24) <= 7
259                 x0 = [18,26];
260                 f3 = @(ta)find_Ta_for_zero_PMV(ta,EP_Rh_rooms(kStep,z)/100);
261                 [x,fval] = fminbnd(f3,x0(1),x0(2));
262
263                 T_room_stpt(1,z) = x;
264                 T_room_stpt_neg(1,z) = 0;
265             % if room is unoccupied, set room to setback temperature
266             else
267                 T_room_stpt(1,z) = 29.44;
268                 T_room_stpt_neg(1,z) = 0;
269             end
270         end
271
272     % initialize arrays
273     Occ_rooms_ref_pred = [];
274     EP_T_ua_pred = [];
275     EP_Rh_ua_pred = [];
276
277     % create future vectors for predictions
278     for i=1:1:n_pred
279         for j=1:1:n_rooms
280             Occ_rooms_ref_pred_sub(j) = zone_people(kStep+i*timestep_OPT - 1,j);
281         end
282         Occ_rooms_ref_pred = horzcat(Occ_rooms_ref_pred,Occ_rooms_ref_pred_sub);
283         EP_T_ua_pred(i) = EP_T_ua_master.EP_T_ua(kStep+i);
284         EP_Rh_ua_pred(i) = EP_Rh_ua_master.EP_Rh_ua(kStep+i);
285     end
286
287     % initialize zone prediction temperatures if first timestep or
288     % beginning of occupied time period

```

```

289     if rem(kStep,1440) == 390 || kStep == 1
290         last_TAHU = 13;
291         last_PEDS = 0.8;
292         T_room_stpt_0 = 22*ones(1,10);
293         setpoints_0 = [22.5*ones(1,n_rooms*n_pred),13*ones(1,n_pred),0.8*ones(1,
                n_pred)];
294     end
295
296     % values for adjusting cost weights in changes of variables; not
297     % used in final simulation; set to zero for normal operation
298     delta_PEDS_weight = 0;
299     delta_TAHU_weight = 0;
300     tpred_diff = 0;
301     diffminchg = 0;
302
303     % function to run optimization
304     [history, setpoints, fval, prev_states, case_status, Tpred_array, J_roompred_all,
        J_chwpred, J_fanpred, ud_pred, ...
305         case_history_array, Troom1_array,
        Troom2_array, grad_act] = ...
306     runfmincon_w_constraints(kStep,
        steps_in_hour, prev_states, setpoints_0
        , n_pred, n_rooms, EP_T_rooms(kStep,:),
        ...
307         timestep_OPT, timestep_PID,
        EP_T_AHU_CW_supp(kStep,1),
        EP_T_AHU_CW_ref(kStep,1),...
308         EP_Rh_rooms(kStep,1:10)/100,
        pyear, T_room_stpt,
        Occ_rooms_ref_pred,
        case1_means, case2_means,...
309         EP_T_oa_pred, EP_Rh_oa_pred, ud,
        lastInput_clg,
        lastInput_flow, ...
310         max_vav_flow, lterm_clg,
        lterm_flow, T_room_mean1,
        T_room_mean2, case_status,
        lb, ub,...
311         last_TAHU, last_PEDS,
        delta_PEDS_weight,
        delta_TAHU_weight,
        tpred_diff,...
        diffminchg, T_room_stpt_0);
312
313     T_room_stpt_0 = setpoints(1:10);
314
315     % if zones are unoccupied, use setback temperature for setpoint
316     for i=1:1:n_rooms
317         if Occ_rooms_ref(1,i) ~= 1 && rem((kStep)/steps_in_hour,24) < 6.5 ||
            Occ_rooms_ref(1,i) ~= 1 && rem((kStep)/steps_in_hour,24) > 7
318             T_room_stpt_0(i) = 29.44;
319         end
320     end
321
322     % assign optimization variables to coder friendly variables
323     T_AHU_stpt_0 = setpoints(n_pred*n_rooms + 1);
324     last_TAHU = T_AHU_stpt_0;
325     P_EDS_stpt_0 = setpoints(n_pred*n_rooms + n_pred + 1);
326     last_PEDS = P_EDS_stpt_0;
327     setpoint_clg = T_room_stpt_0;
328     setpoints_0 = setpoints; % setpoints for next optimization
329 end
330
331 % simulate zone vav controllers to determine air flow fractions to send to
332 % EnergyPlus

```

```

333 for i=1:1:10 %%%%% room vav controllers %%%%%
334 % reset I terms of cooling PID controller is at beginning or end of
335 % occupied time period
336 if rem((kStep)/steps_in_hour,24) == 6.5
337     Iterm_clg(1,i) = 0;
338 elseif rem((kStep)/steps_in_hour,24) == 18 || kStep == 1
339     Iterm_clg(1,i) = 0;
340 end
341
342 % Cooling loop controller
343 input_clg = EP_T_rooms(kStep,i);
344 [output_clg, lastInput_clg(1,i), Iterm_clg(1,i)] = pid_control(setpoint_clg
    (1,i), input_clg, ...
345     lastInput_clg(1,i), Iterm_clg(1,i), kp_clg,...
346     ki_clg, kd_clg, outMax_clg, outMin_clg, bias_clg, control_act_clg,
        timestep_PID);
347
348 % reset I terms of cooling PID controller is at beginning or end of
349 % occupied time period
350 if rem((kStep)/steps_in_hour,24) == 6.5
351     Iterm_flow(1,i) = 0;
352 elseif rem((kStep)/steps_in_hour,24) == 18 || kStep == 1
353     Iterm_flow(1,i) = 0;
354 end
355
356 % Flow loop controller
357 setpoint_flow = output_clg/100;
358 input_flow = EP_V_rooms(kStep,i)/max_vav_flow(1,i);
359
360 [output_flow, lastInput_flow(1,i), Iterm_flow(1,i)] = pid_control(
    setpoint_flow, input_flow, ...
361     lastInput_flow(1,i), Iterm_flow(1,i), kp_flow,...
362     ki_flow, kd_flow, outMax_flow, outMin_flow, bias_flow, control_act_flow,
        timestep_PID);
363
364 end
365
366 % convert controller outputs to damper percents
367 ud = [output_flow_data(kStep,:) * 100]';
368
369 % determine individual VAV flows
370 vav1_flow = vav1(ud(1,1), P_EDS_stpt_0, []);
371 vav2_flow = vav2(ud(2,1), P_EDS_stpt_0, []);
372 vav3_flow = vav3(ud(3,1), P_EDS_stpt_0, []);
373 vav4_flow = vav4(ud(4,1), P_EDS_stpt_0, []);
374 vav5_flow = vav5(ud(5,1), P_EDS_stpt_0, []);
375 vav6_flow = vav6(ud(6,1), P_EDS_stpt_0, []);
376 vav7_flow = vav7(ud(7,1), P_EDS_stpt_0, []);
377 vav8_flow = vav8(ud(8,1), P_EDS_stpt_0, []);
378 vav9_flow = vav9(ud(9,1), P_EDS_stpt_0, []);
379 vav11_flow = vav11(ud(10,1), P_EDS_stpt_0, []);
380
381 vavs = [vav1_flow, vav2_flow, vav3_flow, vav4_flow, vav5_flow, vav6_flow, ...
    vav7_flow, vav8_flow, vav9_flow, vav11_flow];
382 vavs_flow = sum(vavs);
383
384 % convert VAV flows to VAV air flow fractions
385 V_rooms_ref = vavs./(max_vav_flow*2118.88);
386
387 for i=1:1:n_rooms % determine the comfort cost for each room
388     if Occ_rooms_ref(1,i) == 1
389         [J_room_all(1,i), PMV_all(1,i), error_all(1,i), dLOP_dpmv_all(1,i),
            R_all(1,i)] = ...
390             room_cost(timestep_OPT, pyear(i), EP_T_rooms(kStep,i), EP_Rh_rooms(
                kStep,i)/100, T_room_stpt(1,i), EP_T_rooms(kStep,i));

```

```

391         else
392             J_room_all(1,i) = 0;
393         end
394     end
395
396     [J_chw] = chw_cost_act(T_AHU_stpt_0, EP_T_AHU_CW_flow(kStep,1), EP_T_AHU_CW_supp
        (kStep,1), EP_T_AHU_CW_ret(kStep,1), timestep_OPT); % determine the cost of
        chilled water
397     [J_fan] = fan_cost(ud, P_EDS_stpt_0, timestep_OPT); % determine the cost of fan
        power
398
399     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
400     % Set Points
401
402     SP = [T_AHU_stpt_0, T_rooms_hot_ref, T_rooms_cold_ref, V_rooms_ref,
        Occ_rooms_ref]; % setpoints passed to energy plus
403
404     ep.write(mlepEncodeRealData(VERNUMBER, 0, (kStep-1)*deltaT, SP));
405
406     kStep = kStep + 1;
407 end
408
409 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
410 % Stop EnergyPlus
411 ep.stop;
412 disp(['Stopped with flag ' num2str(flag)]);
413
414 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
415 %
416 % =====FLAGS=====
417 % Flag Description
418 % +1 Simulation reached end time.
419 % 0 Normal operation.
420 % -1 Simulation terminated due to an unspecified error.
421 % -10 Simulation terminated due to an error during the initialization.
422 % -20 Simulation terminated due to an error during the time integration.
423 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Optimization file for the dynamic MPC method.

```

1 function [history, setpoints, fval, prev_states, case_status, Tpred_array,
2     J_roompred_all_array, J_chwpred_array, ...
3     J_fanpred_array, ud_array,
4     case_history_array, ...
5     Troom1_array, Troom2_array,
6     grad] = ...
7     runfmincon(kStep, steps_in_hour, prev_states, setpoints_0,
8         n_pred, n_rooms, EP_T_rooms, ...
9         OPT_timestep, PID_timestep,
10            T_supp, T_ret, Rh_rooms, ...
11            pyear, T_room_stpt,
12            Occ_rooms_ref, case1_means,
13            case2_means, ...
14            EP_T_ua, EP_Rh_ua, ud,
15            lastInput_clg,
16            lastInput_flow, ...
17            max_vav_flow, lterm_clg,
18            lterm_flow, Troom_mean1,
19            Troom_mean2, ...
20            case_status, lb, ub, last_TAHU,
21            last_PEDS, delta_PEDS_weight,
22            ...
23            delta_TAHU_weight, tpred_diff,
24            diffminchg, last_Tstpts)

```

```

11
12 % This function contains all the acutal optimization code to help simplify
13 % the main simulation m-file.
14
15 % Set up shared variables with OUTFUN
16 history.x = [];
17 history.fval = [];
18
19 % call optimization with no gradient
20 options = optimset('OutputFcn',@outfun,'Display','iter',...
21                  'DiffMinChange',diffminchg,'Algorithm','interior-point',...
22                  'GradObj','off','Hessian','lbfgs','MaxFunEvals',5000);
23
24 f1 = @(u)dynamic_objective_func_no_grad(u, n_pred, n_rooms, prev_states, EP_T_rooms,
25    ...
26                                     OPT_timestep, PID_timestep,
27                                     T_supp, T_ret, Rh_rooms,...
28                                     pyear, T_room_stpt,
29                                     Occ_rooms_ref, case1_means,
30                                     case2_means,...
31                                     EP_T_oa, EP_Rh_oa, ud,
32                                     lastInput_clg,
33                                     lastInput_flow,...
34                                     max_vav_flow, Iterm_clg,
35                                     Iterm_flow,...
36                                     Troom_mean1, Troom_mean2,
37                                     case_status, last_TAHU,
38                                     last_PEDS,...
39                                     delta_PEDS_weight,
40                                     delta_TAHU_weight,
41                                     tpred_diff);
42
43 % optimization constraint function
44 f2 = @(x)find_opt_constraints_dyn(x,ud,n_rooms,n_pred,last_TAHU,last_PEDS,
45    last_Tstpts);
46
47 % if during occupied/pre-cooling hours, perform the same cost function
48 % calculation with the chosen optimization variables
49 if rem((kStep)/steps_in_hour,24) >= 6.5 && rem((kStep)/steps_in_hour,24) < 18
50     [setpoints,fval,~,~,grad] = fmincon(f1,setpoints_0,[],[],[],[],lb,ub,f2,
51    options);
52
53     [prev_states,case_status,Tpred_array,J_roompred_all_array,J_chwpred_array,
54     J_fanpred_array,ud_array,...
55     case_history_array, Troom1_array, Troom2_array] = ...
56     dynamic_objective_func_cal(setpoints, n_pred, n_rooms,
57     prev_states, EP_T_rooms,...
58     OPT_timestep, PID_timestep,
59     T_supp, T_ret, Rh_rooms,...
60     pyear, T_room_stpt,
61     Occ_rooms_ref, case1_means,
62     case2_means,...
63     EP_T_oa, EP_Rh_oa, ud,
64     lastInput_clg,
65     lastInput_flow,...
66     max_vav_flow, Iterm_clg,
67     Iterm_flow,...
68     Troom_mean1, Troom_mean2,
69     last_TAHU, last_PEDS,...
70     delta_PEDS_weight,
71     delta_TAHU_weight,
72     tpred_diff);
73
74 else
75     setpoints = [29.44*ones(1,n_rooms*n_pred),18*ones(1,n_pred),zeros(1,n_pred)];

```

```

51     fval = 0;
52     Tpred_array = zeros(1,n_pred*n_rooms);
53     J_roompred_all_array = zeros(1,n_pred*n_rooms);
54     J_chwpred_array = zeros(1,n_pred);
55     J_fanpred_array = zeros(1,n_pred);
56     ud_array = zeros(1,n_pred*n_rooms);
57     case_history_array = zeros(1,n_pred*n_rooms);
58     Troom1_array = zeros(1,n_pred*n_rooms+n_rooms);
59     Troom2_array = zeros(1,n_pred*n_rooms+n_rooms);
60     grad_calc_total = zeros(n_pred*n_rooms +n_pred+n_pred,1);
61     grad = zeros(n_pred*n_rooms+n_pred+n_pred,1);
62 end
63
64 % output function from optimization in case you want to inspect data used
65 % in fmincon optimizations
66 function stop = outfun(x,optimValues,state)
67     stop = false;
68
69     switch state
70     case 'init'
71     case 'iter'
72         % Concatenate current point and objective function
73         % value with history. x must be a row vector.
74         history.fval = [history.fval; optimValues.fval];
75         history.x = [history.x; x];
76     case 'done'
77     otherwise
78     end
79 end
80 end

```

Objective function file for the dynamic MPC method.

```

1 function [J_total] = dynamic_objective_func_no_grad(u, n_pred, n_rooms, prev_states
2     , EP_T_rooms,...
3                                     OPT_timestep, PID_timestep,
4                                     T_supp, T_ret, Rh_rooms,...
5                                     pyear, T_room_stpt,
6                                     Occ_rooms_ref, case1_means,
7                                     case2_means,...
8                                     EP_T_oa, EP_Rh_oa, ud,
9                                     lastInput_clg,
10                                    lastInput_flow,...
11                                    max_vav_flow, Iterm_clg,
12                                    Iterm_flow,...
13                                    Troom_mean1, Troom_mean2,
14                                    case_status, last_TAHU,
15                                    last_PEDS,...
16                                    delta_PEDS_weight,
17                                    delta_TAHU_weight,
18                                    tpred_diff)
19
20 % This function is the objective function for the fmincon optimization for
21 % the dynamic MPC control method.
22
23 % load identified model matrices for making predictions
24 load('model_matrices.mat');
25 % hand tuned model for case 2, zone 1
26 B21 = [0.065, 0.11, 0.43, 0.001, 0.01, 0.01];
27
28 % initialize arrays
29 case_history_array = [];
30 Tpred_array = [];
31 ud_array = [];

```

```

21 % assign optimization variables to coder friendly variables
22 T_stpt_opt = u(1:n_rooms*n_pred);
23 T_DA_stpt = u(length(T_stpt_opt)+1:length(T_stpt_opt) + n_pred);
24 P_EDS_stpt = u(length(T_stpt_opt)+length(T_DA_stpt)+1:length(T_stpt_opt)+length(
    T_DA_stpt)+n_pred);
25
26 y = EP_T_rooms(1:10); % current room temperatures
27 Tpred = y; % initial prediction values will be the current room temps
28
29 % initialize states
30 %%%
31 %%% Initial states need to be the first states from the previous
32 %%% optimization step, or zero if it is the first optimization!!!
33 %%%
34 x1{1} = prev_states.x1{1};
35 x21{1} = prev_states.x21{1};
36 x22{1} = prev_states.x22{1};
37 x23{1} = prev_states.x23{1};
38 x24{1} = prev_states.x24{1};
39 x25{1} = prev_states.x25{1};
40 x26{1} = prev_states.x26{1};
41 x27{1} = prev_states.x27{1};
42 x28{1} = prev_states.x28{1};
43 x29{1} = prev_states.x29{1};
44 x210{1} = prev_states.x210{1};
45
46 y_hat1 = y - Troom_mean1(1:10); % initialize case 1 predictions
47 y_hat2 = y - Troom_mean2(1:10); % initialize case 2 predictions
48
49 % bring states to where the systems currently are
50 x1{1} = x1{1} + K1*(y_hat1(1,:) - C1*x1{1});
51 x21{1} = x21{1} + K21*(y_hat2(1,1) - C21*x21{1});
52 x22{1} = x22{1} + K22*(y_hat2(1,2) - C22*x22{1});
53 x23{1} = x23{1} + K23*(y_hat2(1,3) - C23*x23{1});
54 x24{1} = x24{1} + K24*(y_hat2(1,4) - C24*x24{1});
55 x25{1} = x25{1} + K25*(y_hat2(1,5) - C25*x25{1});
56 x26{1} = x26{1} + K26*(y_hat2(1,6) - C26*x26{1});
57 x27{1} = x27{1} + K27*(y_hat2(1,7) - C27*x27{1});
58 x28{1} = x28{1} + K28*(y_hat2(1,8) - C28*x28{1});
59 x29{1} = x29{1} + K29*(y_hat2(1,9) - C29*x29{1});
60 x210{1} = x210{1} + K210*(y_hat2(1,10) - C210*x210{1});
61
62 % get initial zone temperature predictions
63 y_hat1(1,:) = (C1*x1{1})';
64 y_hat21(1) = (C21*x21{1})'; y_hat22(1) = (C22*x22{1})'; y_hat23(1) = (C23*x23{1})';
    y_hat24(1) = (C24*x24{1})'; y_hat25(1) = (C25*x25{1})';
65 y_hat26(1) = (C26*x26{1})'; y_hat27(1) = (C27*x27{1})'; y_hat28(1) = (C28*x28{1})';
    y_hat29(1) = (C29*x29{1})'; y_hat210(1) = (C210*x210{1})';
66 y_hat2(1,:) = [y_hat21(1), y_hat22(1), y_hat23(1), y_hat24(1), y_hat25(1), y_hat26
    (1), y_hat27(1), y_hat28(1), y_hat29(1), y_hat210(1)];
67 y_hat2(1,:) = [y_hat21(1), y_hat22(1), y_hat23(1), y_hat24(1), y_hat25(1),...
68     y_hat26(1), y_hat27(1), y_hat28(1), y_hat29(1), y_hat210(1)
    ];
69
70 Troom1(1,:) = y_hat1(1,:) + Troom_mean1(1:10);
71 Troom21(1) = y_hat21(1) + Troom_mean2(1); Troom22(1) = y_hat22(1) + Troom_mean2(2);
    Troom23(1) = y_hat23(1) + Troom_mean2(3);
72 Troom24(1) = y_hat24(1) + Troom_mean2(4); Troom25(1) = y_hat25(1) + Troom_mean2(5);
    Troom26(1) = y_hat26(1) + Troom_mean2(6);
73 Troom27(1) = y_hat27(1) + Troom_mean2(7); Troom28(1) = y_hat28(1) + Troom_mean2(8);
    Troom29(1) = y_hat29(1) + Troom_mean2(9);
74 Troom210(1) = y_hat210(1) + Troom_mean2(10);
75 Troom2(1,:) = [Troom21(1), Troom22(1), Troom23(1), Troom24(1), Troom25(1), ...
76     Troom26(1), Troom27(1), Troom28(1), Troom29(1), Troom210(1)];

```

```

77
78 % for the prediction horizon, loop over each prediction for all the zones
79 for k=2:1:n_pred+1
80
81     T_room_start = Tpred; % starting temps for simulatiing damper controls
82
83     % build model inputs for each case
84     [case1_input, case2_input] = model_input_builder(EP_T_oa(k-1), T_stpt_opt(1 + (k
        -1)*n_rooms - n_rooms:n_rooms*k - n_rooms), T_DA_stpt(k-1), P_EDS_stpt(k-1),
85         ...
        EP_Rh_oa(k-1), [Tpred,
        EP_T_rooms(11)],
        case1_means, case2_means);
86
87     % compute states for each case/zone
88     x1_pred{k} = (A1*x1{k-1} + B1*case1_input ');
89     x21_pred{k} = (A21*x21{k-1} + B21*case2_input {1} ');
90     x22_pred{k} = (A22*x22{k-1} + B22*case2_input {2} ');
91     x23_pred{k} = (A23*x23{k-1} + B23*case2_input {3} ');
92     x24_pred{k} = (A24*x24{k-1} + B24*case2_input {4} ');
93     x25_pred{k} = (A25*x25{k-1} + B25*case2_input {5} ');
94     x26_pred{k} = (A26*x26{k-1} + B26*case2_input {6} ');
95     x27_pred{k} = (A27*x27{k-1} + B27*case2_input {7} ');
96     x28_pred{k} = (A28*x28{k-1} + B28*case2_input {8} ');
97     x29_pred{k} = (A29*x29{k-1} + B29*case2_input {9} ');
98     x210_pred{k} = (A210*x210{k-1} + B210*case2_input {10} ');
99
100    % make zone predictions for each case/zone
101    y_hat1_pred(k,:) = (C1*x1_pred{k}) ';
102    y_hat21_pred(k) = (C21*x21_pred{k}) '; y_hat22_pred(k) = (C22*x22_pred{k}) ';
103    y_hat23_pred(k) = (C23*x23_pred{k}) '; y_hat24_pred(k) = (C24*x24_pred{k}) ';
104    y_hat25_pred(k) = (C25*x25_pred{k}) '; y_hat26_pred(k) = (C26*x26_pred{k}) ';
105    y_hat27_pred(k) = (C27*x27_pred{k}) '; y_hat28_pred(k) = (C28*x28_pred{k}) ';
106    y_hat29_pred(k) = (C29*x29_pred{k}) '; y_hat210_pred(k) = (C210*x210_pred{k}) ';
107    y_hat2_pred(k,:) = [y_hat21_pred(k), y_hat22_pred(k), y_hat23_pred(k),
        y_hat24_pred(k), y_hat25_pred(k), ...
        y_hat26_pred(k), y_hat27_pred(k), y_hat28_pred(k),
        y_hat29_pred(k), y_hat210_pred(k)];
108
109    Troom1_pred(k,:) = y_hat1_pred(k,:) + Troom_mean1(1:10);
110    Troom2_pred(k,:) = y_hat2_pred(k,:) + Troom_mean2(1:10);
111
112    % determine if zones are in case 1 or case 2
113    for i=1:1:n_rooms
114        % case 1
115        if (Troom1_pred(k,i) - Troom2_pred(k,i)) > tpred_diff && ud(i) < 95
116            pred_case_history(i) = 0;
117            T_room_model_input(1,i) = T_stpt_opt(i + (k-1)*n_rooms - n_rooms);
118        % case 2
119        else
120            pred_case_history(i) = 1;
121            T_room_model_input(1,i) = Tpred(i);
122        end
123    end
124
125    % adjust model inputs for zones that are in case 2; specifically, make
126    % the zone temperature setpoints for the zones in case 2 equal to the
127    % respective predicted zone temperatures.
128    [case1_input, case2_input] = model_input_builder(EP_T_oa(k-1),
        T_room_model_input, T_DA_stpt(k-1), P_EDS_stpt(k-1), ...
        EP_Rh_oa(k-1), [Tpred,
        EP_T_rooms(11)],
        case1_means, case2_means);
129

```

```

130 % determine states for correct case 1/case 2 calculations
131 x1_pred{k} = (A1*x1{k-1} + B1*case1_input ');
132 x21_pred{k} = (A21*x21{k-1} + B21*case2_input {1} ');
133 x22_pred{k} = (A22*x22{k-1} + B22*case2_input {2} ');
134 x23_pred{k} = (A23*x23{k-1} + B23*case2_input {3} ');
135 x24_pred{k} = (A24*x24{k-1} + B24*case2_input {4} ');
136 x25_pred{k} = (A25*x25{k-1} + B25*case2_input {5} ');
137 x26_pred{k} = (A26*x26{k-1} + B26*case2_input {6} ');
138 x27_pred{k} = (A27*x27{k-1} + B27*case2_input {7} ');
139 x28_pred{k} = (A28*x28{k-1} + B28*case2_input {8} ');
140 x29_pred{k} = (A29*x29{k-1} + B29*case2_input {9} ');
141 x210_pred{k} = (A210*x210{k-1} + B210*case2_input {10} ');
142
143 x1{k} = x1_pred{k} + K1*(pred_case_history.*(y_hat2_pred(k,:) - C1*x1_pred{k}
144 + Troom_mean2(1:10) - Troom_mean1(1:10)));
145
146 x21{k} = x21_pred{k} + K21*((~pred_case_history(1))*(y_hat1_pred(k,1) - C21*
147 x21_pred{k} + Troom_mean1(1) - Troom_mean2(1)));
148 x22{k} = x22_pred{k} + K22*((~pred_case_history(2))*(y_hat1_pred(k,2) - C22*
149 x22_pred{k} + Troom_mean1(2) - Troom_mean2(2)));
150 x23{k} = x23_pred{k} + K23*((~pred_case_history(3))*(y_hat1_pred(k,3) - C23*
151 x23_pred{k} + Troom_mean1(3) - Troom_mean2(3)));
152 x24{k} = x24_pred{k} + K24*((~pred_case_history(4))*(y_hat1_pred(k,4) - C24*
153 x24_pred{k} + Troom_mean1(4) - Troom_mean2(4)));
154 x25{k} = x25_pred{k} + K25*((~pred_case_history(5))*(y_hat1_pred(k,5) - C25*
155 x25_pred{k} + Troom_mean1(5) - Troom_mean2(5)));
156 x26{k} = x26_pred{k} + K26*((~pred_case_history(6))*(y_hat1_pred(k,6) - C26*
157 x26_pred{k} + Troom_mean1(6) - Troom_mean2(6)));
158 x27{k} = x27_pred{k} + K27*((~pred_case_history(7))*(y_hat1_pred(k,7) - C27*
159 x27_pred{k} + Troom_mean1(7) - Troom_mean2(7)));
160 x28{k} = x28_pred{k} + K28*((~pred_case_history(8))*(y_hat1_pred(k,8) - C28*
161 x28_pred{k} + Troom_mean1(8) - Troom_mean2(8)));
162 x29{k} = x29_pred{k} + K29*((~pred_case_history(9))*(y_hat1_pred(k,9) - C29*
163 x29_pred{k} + Troom_mean1(9) - Troom_mean2(9)));
164 x210{k} = x210_pred{k} + K210*((~pred_case_history(10))*(y_hat1_pred(k,10) -
165 C210*x210_pred{k} + Troom_mean1(10) - Troom_mean2(10)));
166
167 % make correct room predictions for case 1/case 2 situations
168 y_hat1(k,:) = (C1*x1{k})';
169 y_hat21(k) = (C21*x21{k})'; y_hat22(k) = (C22*x22{k})'; y_hat23(k) = (C23*x23{k}
170 )'; y_hat24(k) = (C24*x24{k})'; y_hat25(k) = (C25*x25{k})';
171 y_hat26(k) = (C26*x26{k})'; y_hat27(k) = (C27*x27{k})'; y_hat28(k) = (C28*x28{k}
172 )'; y_hat29(k) = (C29*x29{k})'; y_hat210(k) = (C210*x210{k})';
173 y_hat2(k,:) = [y_hat21(k), y_hat22(k), y_hat23(k), y_hat24(k), y_hat25(k),
174 y_hat26(k), y_hat27(k), y_hat28(k), y_hat29(k), y_hat210(k)];
175
176 Troom1(k,:) = y_hat1(k,:) + Troom_mean1(1:10);
177 Troom21(k) = y_hat21(k) + Troom_mean2(1); Troom22(k) = y_hat22(k) + Troom_mean2
178 (2); Troom23(k) = y_hat23(k) + Troom_mean2(3);
179 Troom24(k) = y_hat24(k) + Troom_mean2(4); Troom25(k) = y_hat25(k) + Troom_mean2
180 (5); Troom26(k) = y_hat26(k) + Troom_mean2(6);
181 Troom27(k) = y_hat27(k) + Troom_mean2(7); Troom28(k) = y_hat28(k) + Troom_mean2
182 (8); Troom29(k) = y_hat29(k) + Troom_mean2(9);
183 Troom210(k) = y_hat210(k) + Troom_mean2(10);
184 Troom2(k,:) = [Troom21(k), Troom22(k), Troom23(k), Troom24(k), Troom25(k), ...
185 Troom26(k), Troom27(k), Troom28(k), Troom29(k), Troom210(k)];
186
187 % loop over rooms and assign prediction temperature for correct case
188 for i=1:1:n_rooms
189     switch i
190         case 1 % room 1
191             if pred_case_history(i) == 0 % if not fully open, and able to reach
192                 temperature
193                 Tpred(i) = Troom1(k,i);

```

```

176         case_history(i) = 0;
177     else % if damper is fully open
178         Tpred(i) = Troom21(k);
179         case_history(i) = 1;
180     end
181 case 2 % room 2
182     if pred_case_history(i) == 0 % if not fully open, and able to reach
        temperature
183         Tpred(i) = Troom1(k,i);
184         case_history(i) = 0;
185     else % if damper is fully open
186         Tpred(i) = Troom22(k);
187         case_history(i) = 1;
188     end
189 case 3 % room 3
190     if pred_case_history(i) == 0 % if not fully open, and able to reach
        temperature
191         Tpred(i) = Troom1(k,i);
192         case_history(i) = 0;
193     else % if damper is fully open
194         Tpred(i) = Troom23(k);
195         case_history(i) = 1;
196     end
197 case 4 % room 4
198     if pred_case_history(i) == 0 % if not fully open, and able to reach
        temperature
199         Tpred(i) = Troom1(k,i);
200         case_history(i) = 0;
201     else % if damper is fully open
202         Tpred(i) = Troom24(k);
203         case_history(i) = 1;
204     end
205 case 5 % room 5
206     if pred_case_history(i) == 0 % if not fully open, and able to reach
        temperature
207         Tpred(i) = Troom1(k,i);
208         case_history(i) = 0;
209     else % if damper is fully open
210         Tpred(i) = Troom25(k);
211         case_history(i) = 1;
212     end
213 case 6 % room 6
214     if pred_case_history(i) == 0 % if not fully open, and able to reach
        temperature
215         Tpred(i) = Troom1(k,i);
216         case_history(i) = 0;
217     else % if damper is fully open
218         Tpred(i) = Troom26(k);
219         case_history(i) = 1;
220     end
221 case 7 % room 7
222     if pred_case_history(i) == 0 % if not fully open, and able to reach
        temperature
223         Tpred(i) = Troom1(k,i);
224         case_history(i) = 0;
225     else % if damper is fully open
226         Tpred(i) = Troom27(k);
227         case_history(i) = 1;
228     end
229 case 8 % room 8
230     if pred_case_history(i) == 0 % if not fully open, and able to reach
        temperature
231         Tpred(i) = Troom1(k,i);
232         case_history(i) = 0;

```

```

233         else % if damper is fully open
234             Tpred(i) = Troom28(k);
235             case_history(i) = 1;
236         end
237     case 9 % room 9
238         if pred_case_history(i) == 0 % if not fully open, and able to reach
                temperature
239             Tpred(i) = Troom1(k,i);
240             case_history(i) = 0;
241         else % if damper is fully open
242             Tpred(i) = Troom29(k);
243             case_history(i) = 1;
244         end
245     case 10 % room 10
246         if pred_case_history(i) == 0 % if not fully open, and able to reach
                temperature
247             Tpred(i) = Troom1(k,i);
248             case_history(i) = 0;
249         else % if damper is fully open
250             Tpred(i) = Troom210(k);
251             case_history(i) = 1;
252         end
253     end
254
255     % determine room cost if room is occupied, otherwise set room cost
256     % to zero
257     if Occ_rooms_ref(i + (k-1)*n_rooms-n_rooms) == 1
258         [J_room_all(1,i), PMV_all(1,i), error_all(1,i), dLOP_dpmv_all(1,i),
                Rcalc(1,i)] = ...
259             room_cost(OPT_timestep, pyear(1,i), Tpred(i), Rh_rooms(i),
                T_room_stpt(i), Tpred(i));
260     else
261         J_room_all(1,i) = 0;
262     end
263 end
264
265 % calculate future damper positions as required for horizon predictions
266 [ud, lastInput_clg, lastInput_flow, Iterm_clg, Iterm_flow] = ...
267     determine_future_ud(n_rooms, T_room_start, Tpred, P_EDS_stpt(k-1),
                T_stpt_opt,...
268     lastInput_clg, lastInput_flow, max_vav_flow, PID_timestep, Iterm_clg,
                Iterm_flow);
269
270 % arrays for storing data for later review
271 Tpred_array = horzcat(Tpred_array, Tpred);
272 case_history_array = horzcat(case_history_array, case_history);
273
274 % determine total cost to be returned by function
275 J_room_total(k-1,:) = sum(J_room_all);
276 chw_flow = 3.949579252219686; % maximum mass flow rate [kg/s]
277 J_chw(k-1,:) = chw_cost(T_DA_stpt(k-1), chw_flow, T_supp, T_ret, OPT_timestep);
278 % determine the cost of chilled water
279 J_fan(k-1,:) = fan_cost(ud, P_EDS_stpt(k-1), OPT_timestep); % determine the
                cost of fan power
280 total_cost(k-1,:) = J_room_total(k-1,:) + J_chw(k-1,:) + J_fan(k-1,:)...
                + delta_PEDS_weight*abs(last_PEDS - P_EDS_stpt(k-1)) +
                delta_TAHU_weight*abs(last_TAHU - T_DA_stpt(k-1));
281
282 % setup values for next optimization iteration
283 last_PEDS = P_EDS_stpt(k-1);
284 last_TAHU = T_DA_stpt(k-1);
285 ud_array = horzcat(ud_array, ud);
286 end
287

```

```

288 % store states for optimization at next timestep
289 prev_states.x1{1} = x1{2};
290 prev_states.x21{1} = x21{2};
291 prev_states.x22{1} = x22{2};
292 prev_states.x23{1} = x23{2};
293 prev_states.x24{1} = x24{2};
294 prev_states.x25{1} = x25{2};
295 prev_states.x26{1} = x26{2};
296 prev_states.x27{1} = x27{2};
297 prev_states.x28{1} = x28{2};
298 prev_states.x29{1} = x29{2};
299 prev_states.x210{1} = x210{2};
300 case_status = case_history_array(1:10);
301 J_total = sum(total_cost);
302 end

```

Objective function file for computing values of the dynamic MPC method after the optimization has completed.

```

1 function [prev_states_new, case_status, Tpred_array, J_room_all_array, J_chw_pass,
2         J_fan_pass, ud_array_pass, ...
3         case_history_array, Troom1_array, Troom2_array]
4         = ...
5         dynamic_objective_func_cal(u, n_pred, n_rooms,
6         prev_states, EP_T_rooms, ...
7         OPT_timestep, PID_timestep,
8         T_supp, T_ret, Rh_rooms, ...
9         pyear, T_room_stpt,
10        Occ_rooms_ref, case1_means,
11        case2_means, ...
12        EP_T_oa, EP_Rh_oa, ud,
13        lastInput_clg,
14        lastInput_flow, ...
15        max_vav_flow, Iterm_clg,
16        Iterm_flow, ...
17        Troom_mean1, Troom_mean2,
18        last_TAHU, last_PEDS, ...
19        delta_PEDS_weight,
20        delta_TAHU_weight,
21        tpred_diff)
22
23 % This function is the objective function after the fmincon optimization to
24 % be able to inspect values computed within the optimization.
25
26 % load identified model matrices for making predictions
27 load('model_matrices.mat');
28 % hand tuned model for case 2, zone 1
29 B21 = [0.065, 0.11, 0.43, 0.001, 0.01, 0.01];
30
31 % initialize arrays
32 case_history_array = [];
33 Tpred_array = [];
34 ud_array = [];
35 J_chw_array = [];
36 J_fan_array = [];
37 J_room_all_array = [];
38 Troom1_array = [];
39 Troom2_array = [];
40
41 % assign optimization variables to coder friendly variables
42 T_stpt_opt = u(1:10*n_pred);
43 T_DA_stpt = u(length(T_stpt_opt)+1:length(T_stpt_opt) + n_pred);
44 P_EDS_stpt = u(length(T_stpt_opt)+length(T_DA_stpt)+1:length(T_stpt_opt)+length(
45     T_DA_stpt)+n_pred);
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

33 y = EP_T_rooms(1:10); % current room temperatures
34 Tpred = y; % initial prediction values will be the current room temps
35
36 % initialize states
37 %%%
38 %%% Initial states need to be the first states from the previous
39 %%% optimization step, or zero if it is the first optimization!!!
40 %%%
41 x1{1} = prev_states.x1{1};
42 x21{1} = prev_states.x21{1};
43 x22{1} = prev_states.x22{1};
44 x23{1} = prev_states.x23{1};
45 x24{1} = prev_states.x24{1};
46 x25{1} = prev_states.x25{1};
47 x26{1} = prev_states.x26{1};
48 x27{1} = prev_states.x27{1};
49 x28{1} = prev_states.x28{1};
50 x29{1} = prev_states.x29{1};
51 x210{1} = prev_states.x210{1};
52
53 y_hat1 = y - Troom_mean1(1:10); % initialize case 1 predictions
54 y_hat2 = y - Troom_mean2(1:10); % initialize case 2 predictions
55
56 % bring states to where the systems currently are
57 x1{1} = x1{1} + K1*(y_hat1(1,:) - C1*x1{1});
58 x21{1} = x21{1} + K21*(y_hat2(1,1) - C21*x21{1});
59 x22{1} = x22{1} + K22*(y_hat2(1,2) - C22*x22{1});
60 x23{1} = x23{1} + K23*(y_hat2(1,3) - C23*x23{1});
61 x24{1} = x24{1} + K24*(y_hat2(1,4) - C24*x24{1});
62 x25{1} = x25{1} + K25*(y_hat2(1,5) - C25*x25{1});
63 x26{1} = x26{1} + K26*(y_hat2(1,6) - C26*x26{1});
64 x27{1} = x27{1} + K27*(y_hat2(1,7) - C27*x27{1});
65 x28{1} = x28{1} + K28*(y_hat2(1,8) - C28*x28{1});
66 x29{1} = x29{1} + K29*(y_hat2(1,9) - C29*x29{1});
67 x210{1} = x210{1} + K210*(y_hat2(1,10) - C210*x210{1});
68
69 % get initial zone temperature predictions
70 y_hat1(1,:) = (C1*x1{1})';
71 y_hat21(1) = (C21*x21{1})'; y_hat22(1) = (C22*x22{1})'; y_hat23(1) = (C23*x23{1})';
72 y_hat24(1) = (C24*x24{1})'; y_hat25(1) = (C25*x25{1})';
73 y_hat26(1) = (C26*x26{1})'; y_hat27(1) = (C27*x27{1})'; y_hat28(1) = (C28*x28{1})';
74 y_hat29(1) = (C29*x29{1})'; y_hat210(1) = (C210*x210{1})';
75 y_hat2(1,:) = [y_hat21(1), y_hat22(1), y_hat23(1), y_hat24(1), y_hat25(1), y_hat26(1),
76 y_hat27(1), y_hat28(1), y_hat29(1), y_hat210(1)];
77 Troom1(1,:) = y_hat1(1,:) + Troom_mean1(1:10);
78 Troom21(1) = y_hat21(1) + Troom_mean2(1); Troom22(1) = y_hat22(1) + Troom_mean2(2);
79 Troom23(1) = y_hat23(1) + Troom_mean2(3);
80 Troom24(1) = y_hat24(1) + Troom_mean2(4); Troom25(1) = y_hat25(1) + Troom_mean2(5);
81 Troom26(1) = y_hat26(1) + Troom_mean2(6);
82 Troom27(1) = y_hat27(1) + Troom_mean2(7); Troom28(1) = y_hat28(1) + Troom_mean2(8);
83 Troom29(1) = y_hat29(1) + Troom_mean2(9);
84 Troom210(1) = y_hat210(1) + Troom_mean2(10);
85 Troom2(1,:) = [Troom21(1), Troom22(1), Troom23(1), Troom24(1), Troom25(1), ...
86 Troom26(1), Troom27(1), Troom28(1), Troom29(1), Troom210(1)];
87
88 % for the prediction horizon, loop over each prediction for all the zones
89 for k=2:1:n_pred+1
90     T_room_start = Tpred; % starting temps for simulatiing damper controls

```

```

90 % build model inputs for each case
91 [case1_input, case2_input] = model_input_builder(EP_T_oa(k-1), T_stpt_opt(1 + (k
    -1)*n_rooms - n_rooms:n_rooms*k - n_rooms), T_DA_stpt(k-1), P_EDS_stpt(k-1),
    ...
92                                     EP_Rh_oa(k-1), [Tpred,
    EP_T_rooms(11)],
    case1_means, case2_means);

93
94 % compute states for each case/zone
95 x1_pred{k} = (A1*x1{k-1} + B1*case1_input');
96 x21_pred{k} = (A21*x21{k-1} + B21*case2_input{1}');
97 x22_pred{k} = (A22*x22{k-1} + B22*case2_input{2}');
98 x23_pred{k} = (A23*x23{k-1} + B23*case2_input{3}');
99 x24_pred{k} = (A24*x24{k-1} + B24*case2_input{4}');
100 x25_pred{k} = (A25*x25{k-1} + B25*case2_input{5}');
101 x26_pred{k} = (A26*x26{k-1} + B26*case2_input{6}');
102 x27_pred{k} = (A27*x27{k-1} + B27*case2_input{7}');
103 x28_pred{k} = (A28*x28{k-1} + B28*case2_input{8}');
104 x29_pred{k} = (A29*x29{k-1} + B29*case2_input{9}');
105 x210_pred{k} = (A210*x210{k-1} + B210*case2_input{10}');
106
107 % make zone predictions for each case/zone
108 y_hat1_pred(k,:) = (C1*x1_pred{k})';
109 y_hat21_pred(k) = (C21*x21_pred{k})'; y_hat22_pred(k) = (C22*x22_pred{k})';
    y_hat23_pred(k) = (C23*x23_pred{k})'; y_hat24_pred(k) = (C24*x24_pred{k})';
110 y_hat25_pred(k) = (C25*x25_pred{k})'; y_hat26_pred(k) = (C26*x26_pred{k})';
    y_hat27_pred(k) = (C27*x27_pred{k})'; y_hat28_pred(k) = (C28*x28_pred{k})';
111 y_hat29_pred(k) = (C29*x29_pred{k})'; y_hat210_pred(k) = (C210*x210_pred{k})';
112 y_hat2_pred(k,:) = [y_hat21_pred(k), y_hat22_pred(k), y_hat23_pred(k),
    y_hat24_pred(k), y_hat25_pred(k), ...
113                     y_hat26_pred(k), y_hat27_pred(k), y_hat28_pred(k),
    y_hat29_pred(k), y_hat210_pred(k)];
114
115 Troom1_pred(k,:) = y_hat1_pred(k,:) + Troom_mean1(1:10);
116 Troom2_pred(k,:) = y_hat2_pred(k,:) + Troom_mean2(1:10);
117
118 % determine if zones are in case 1 or case 2
119 for i=1:1:n_rooms
120     % case 1
121     if (Troom1_pred(k,i) - Troom2_pred(k,i)) > tpred_diff && ud(i) < 95
122         pred_case_history(i) = 0;
123         T_room_model_input(1,i) = T_stpt_opt(i + (k-1)*n_rooms - n_rooms);
124     % case 2
125     else
126         pred_case_history(i) = 1;
127         T_room_model_input(1,i) = Tpred(i);
128     end
129 end
130
131 % adjust model inputs for zones that are in case 2; specifically, make
132 % the zone temperature setpoints for the zones in case 2 equal to the
133 % respective predicted zone temperatures.
134 [case1_input, case2_input] = model_input_builder(EP_T_oa(k-1),
    T_room_model_input, T_DA_stpt(k-1), P_EDS_stpt(k-1), ...
135                                     EP_Rh_oa(k-1), [Tpred,
    EP_T_rooms(11)],
    case1_means, case2_means);
136
137 % determine states for correct case 1/case 2 calculations
138 x1_pred{k} = (A1*x1{k-1} + B1*case1_input');
139 x21_pred{k} = (A21*x21{k-1} + B21*case2_input{1}');
140 x22_pred{k} = (A22*x22{k-1} + B22*case2_input{2}');
141 x23_pred{k} = (A23*x23{k-1} + B23*case2_input{3}');
142 x24_pred{k} = (A24*x24{k-1} + B24*case2_input{4}');

```

```

143 x25_pred{k} = (A25*x25{k-1} + B25*case2_input{5}')';
144 x26_pred{k} = (A26*x26{k-1} + B26*case2_input{6}')';
145 x27_pred{k} = (A27*x27{k-1} + B27*case2_input{7}')';
146 x28_pred{k} = (A28*x28{k-1} + B28*case2_input{8}')';
147 x29_pred{k} = (A29*x29{k-1} + B29*case2_input{9}')';
148 x210_pred{k} = (A210*x210{k-1} + B210*case2_input{10}')';
149
150 x1{k} = x1_pred{k} + K1*(pred_case_history'*(y_hat2_pred(k,:) - C1*x1_pred{k}
    + Troom_mean2(1:10)' - Troom_mean1(1:10)'));
151
152 x21{k} = x21_pred{k} + K21*((~pred_case_history(1))*(y_hat1_pred(k,1) - C21*
    x21_pred{k} + Troom_mean1(1)' - Troom_mean2(1)'));
153 x22{k} = x22_pred{k} + K22*((~pred_case_history(2))*(y_hat1_pred(k,2) - C22*
    x22_pred{k} + Troom_mean1(2)' - Troom_mean2(2)'));
154 x23{k} = x23_pred{k} + K23*((~pred_case_history(3))*(y_hat1_pred(k,3) - C23*
    x23_pred{k} + Troom_mean1(3)' - Troom_mean2(3)'));
155 x24{k} = x24_pred{k} + K24*((~pred_case_history(4))*(y_hat1_pred(k,4) - C24*
    x24_pred{k} + Troom_mean1(4)' - Troom_mean2(4)'));
156 x25{k} = x25_pred{k} + K25*((~pred_case_history(5))*(y_hat1_pred(k,5) - C25*
    x25_pred{k} + Troom_mean1(5)' - Troom_mean2(5)'));
157 x26{k} = x26_pred{k} + K26*((~pred_case_history(6))*(y_hat1_pred(k,6) - C26*
    x26_pred{k} + Troom_mean1(6)' - Troom_mean2(6)'));
158 x27{k} = x27_pred{k} + K27*((~pred_case_history(7))*(y_hat1_pred(k,7) - C27*
    x27_pred{k} + Troom_mean1(7)' - Troom_mean2(7)'));
159 x28{k} = x28_pred{k} + K28*((~pred_case_history(8))*(y_hat1_pred(k,8) - C28*
    x28_pred{k} + Troom_mean1(8)' - Troom_mean2(8)'));
160 x29{k} = x29_pred{k} + K29*((~pred_case_history(9))*(y_hat1_pred(k,9) - C29*
    x29_pred{k} + Troom_mean1(9)' - Troom_mean2(9)'));
161 x210{k} = x210_pred{k} + K210*((~pred_case_history(10))*(y_hat1_pred(k,10) -
    C210*x210_pred{k} + Troom_mean1(10)' - Troom_mean2(10)'));
162
163 % make correct room predictions for case 1/case 2 situations
164 y_hat1(k,:) = (C1*x1{k})';
165 y_hat21(k) = (C21*x21{k})'; y_hat22(k) = (C22*x22{k})'; y_hat23(k) = (C23*x23{k}
    )'; y_hat24(k) = (C24*x24{k})'; y_hat25(k) = (C25*x25{k})';
166 y_hat26(k) = (C26*x26{k})'; y_hat27(k) = (C27*x27{k})'; y_hat28(k) = (C28*x28{k}
    )'; y_hat29(k) = (C29*x29{k})'; y_hat210(k) = (C210*x210{k})';
167 y_hat2(k,:) = [y_hat21(k), y_hat22(k), y_hat23(k), y_hat24(k), y_hat25(k),
    y_hat26(k), y_hat27(k), y_hat28(k), y_hat29(k), y_hat210(k)];
168
169 Troom1(k,:) = y_hat1(k,:) + Troom_mean1(1:10);
170 Troom21(k) = y_hat21(k) + Troom_mean2(1); Troom22(k) = y_hat22(k) + Troom_mean2
    (2); Troom23(k) = y_hat23(k) + Troom_mean2(3);
171 Troom24(k) = y_hat24(k) + Troom_mean2(4); Troom25(k) = y_hat25(k) + Troom_mean2
    (5); Troom26(k) = y_hat26(k) + Troom_mean2(6);
172 Troom27(k) = y_hat27(k) + Troom_mean2(7); Troom28(k) = y_hat28(k) + Troom_mean2
    (8); Troom29(k) = y_hat29(k) + Troom_mean2(9);
173 Troom210(k) = y_hat210(k) + Troom_mean2(10);
174 Troom2(k,:) = [Troom21(k), Troom22(k), Troom23(k), Troom24(k), Troom25(k), ...
    Troom26(k), Troom27(k), Troom28(k), Troom29(k), Troom210(k)];
175
176
177 % loop over rooms and assign prediction temperature for correct case
178 for i=1:1:n_rooms
179     switch i
180         case 1 % room 1
181             if pred_case_history(i) == 0 % if not fully open, and able to reach
                temperature
182                 Tpred(i) = Troom1(k,i);
183                 case_history(i) = 0;
184             else % if damper is fully open
185                 Tpred(i) = Troom21(k);
186                 case_history(i) = 1;
187             end
188         case 2 % room 2

```

```

189         if pred_case_history(i) == 0 % if not fully open, and able to reach
190             temperature
191             Tpred(i) = Troom1(k,i);
192             case_history(i) = 0;
193         else % if damper is fully open
194             Tpred(i) = Troom22(k);
195             case_history(i) = 1;
196         end
197     case 3 % room 3
198         if pred_case_history(i) == 0 % if not fully open, and able to reach
199             temperature
200             Tpred(i) = Troom1(k,i);
201             case_history(i) = 0;
202         else % if damper is fully open
203             Tpred(i) = Troom23(k);
204             case_history(i) = 1;
205         end
206     case 4 % room 4
207         if pred_case_history(i) == 0 % if not fully open, and able to reach
208             temperature
209             Tpred(i) = Troom1(k,i);
210             case_history(i) = 0;
211         else % if damper is fully open
212             Tpred(i) = Troom24(k);
213             case_history(i) = 1;
214         end
215     case 5 % room 5
216         if pred_case_history(i) == 0 % if not fully open, and able to reach
217             temperature
218             Tpred(i) = Troom1(k,i);
219             case_history(i) = 0;
220         else % if damper is fully open
221             Tpred(i) = Troom25(k);
222             case_history(i) = 1;
223         end
224     case 6 % room 6
225         if pred_case_history(i) == 0 % if not fully open, and able to reach
226             temperature
227             Tpred(i) = Troom1(k,i);
228             case_history(i) = 0;
229         else % if damper is fully open
230             Tpred(i) = Troom26(k);
231             case_history(i) = 1;
232         end
233     case 7 % room 7
234         if pred_case_history(i) == 0 % if not fully open, and able to reach
235             temperature
236             Tpred(i) = Troom1(k,i);
237             case_history(i) = 0;
238         else % if damper is fully open
239             Tpred(i) = Troom27(k);
240             case_history(i) = 1;
241         end
242     case 8 % room 8
243         if pred_case_history(i) == 0 % if not fully open, and able to reach
244             temperature
245             Tpred(i) = Troom1(k,i);
246             case_history(i) = 0;
247         else % if damper is fully open
248             Tpred(i) = Troom28(k);
249             case_history(i) = 1;
250         end
251     case 9 % room 9
252         if pred_case_history(i) == 0 % if not fully open, and able to reach

```

```

246         temperature
247         Tpred(i) = Troom1(k,i);
248         case_history(i) = 0;
249     else % if damper is fully open
250         Tpred(i) = Troom29(k);
251         case_history(i) = 1;
252     end
253 case 10 % room 10
254     if pred_case_history(i) == 0 % if not fully open, and able to reach
255         temperature
256         Tpred(i) = Troom1(k,i);
257         case_history(i) = 0;
258     else % if damper is fully open
259         Tpred(i) = Troom210(k);
260         case_history(i) = 1;
261     end
262 end
263 % determine room cost if room is occupied, otherwise set room cost
264 % to zero
265 if Occ_rooms_ref(i + (k-1)*n_rooms-n_rooms) == 1
266     [J_room_all(1,i), PMV_all(1,i), error_all(1,i), dLOP_dpmv_all(1,i),
267     Rcalc(1,i)] = ...
268     room_cost(OPT_timestep, pyear(1,i), Tpred(i), Rh_rooms(i),
269     T_room_stpt(i), Tpred(i));
270 else
271     J_room_all(1,i) = 0;
272 end
273 % calculate future damper positions as required for horizon predictions
274 [ud, lastInput_clg, lastInput_flow, lterm_clg, lterm_flow] = ...
275     determine_future_ud(n_rooms, T_room_start, Tpred, P_EDS_stpt(k-1),
276     T_stpt_opt,...
277     lastInput_clg, lastInput_flow, max_vav_flow, PID_timestep, lterm_clg,
278     lterm_flow);
279 % arrays for storing data for later review
280 Troom1_array = horzcat(Troom1_array,Troom1_pred(k,:));
281 Troom2_array = horzcat(Troom2_array,Troom2_pred(k,:));
282 Tpred_array = horzcat(Tpred_array,Tpred);
283 case_history_array = horzcat(case_history_array,case_history);
284 J_room_all_array = horzcat(J_room_all_array,J_room_all);
285 % determine total cost to be returned by function
286 J_room_total(k-1,:) = sum(J_room_all);
287 chw_flow = 3.949579252219686; % maximum mass flow rate [kg/s]
288 J_chw(k-1,:) = chw_cost(T_DA_stpt(k-1), chw_flow, T_supp, T_ret, OPT_timestep);
289 % determine the cost of chilled water
290 J_fan(k-1,:) = fan_cost(ud, P_EDS_stpt(k-1), OPT_timestep); % determine the cost
291 % of fan power
292 total_cost(k-1,:) = J_room_total(k-1,:) + J_chw(k-1,:) + J_fan(k-1,:)+...
293     + delta_PEDS_weight*abs(last_PEDS - P_EDS_stpt(k-1)) +
294     delta_TAHU_weight*abs(last_TAHU - T_DA_stpt(k-1));
295 % setup values for next optimization iteration
296 last_PEDS = P_EDS_stpt(k-1);
297 last_TAHU = T_DA_stpt(k-1);
298 ud_array = horzcat(ud_array,ud);
299 end
300 % store data for later inspection
301 Troom1_array = horzcat(Troom1(1,:),Troom1_array);
302 Troom2_array = horzcat(Troom2(1,:),Troom2_array);

```

```

301
302 % store states for optimization at next timestep
303 prev_states_new.x1{1} = x1{2};
304 prev_states_new.x21{1} = x21{2};
305 prev_states_new.x22{1} = x22{2};
306 prev_states_new.x23{1} = x23{2};
307 prev_states_new.x24{1} = x24{2};
308 prev_states_new.x25{1} = x25{2};
309 prev_states_new.x26{1} = x26{2};
310 prev_states_new.x27{1} = x27{2};
311 prev_states_new.x28{1} = x28{2};
312 prev_states_new.x29{1} = x29{2};
313 prev_states_new.x210{1} = x210{2};
314 case_status = case_history_array(1:10);
315 J_total = sum(total_cost);
316 J_chw_pass = J_chw';
317 J_fan_pass = J_fan';
318 ud_array_pass = ud_array;
319 end

```

E.3 Shared Code

File for determining chilled water cost.

```

1 function [J_chw] = chw_cost(T_DA_STPT, chw_flow, T_supp, T_ret, ts)
2 % function for calculating the cost associated with producing chilled water
3 c_h20 = 4.187; % specific heat of water [J/g/degC]
4 delta_T_h20 = T_ret - T_supp; % difference in supply and return h20 temps [degC]
5
6 % fit to relationship between discharge air temperature setpoint and
7 % chilled water valve position
8 fit_TDA = (-332.9*T_DA_STPT^2 + 1.089e+04*T_DA_STPT - 6.802e+04)/...
9           (T_DA_STPT^2 + 7787*T_DA_STPT - 7.606e+04);
10
11 % objective function for chilled water cost [$]
12 J_chw = fit_TDA*c_h20*delta_T_h20*ts/60*0.00341214*3682/241;
13
14 end

```

File for determining chilled water cost based on the chilled water mass flow reported from EnergyPlus.

```

1 function [J_chw] = chw_cost_act(T_DA_STPT, chw_flow, T_supp, T_ret, ts)
2 % function for calculating the cost associated with producing chilled water
3 % using the mass flow provided from EnergyPlus
4
5 c_h20 = 4.187; % specific heat of water [J/g/degC]
6 delta_T_h20 = T_ret - T_supp; % difference in supply and return h20 temps [degC]
7
8 % objective function for chilled water cost [$]
9 J_chw = chw_flow*c_h20*delta_T_h20*ts/60*0.00341214*3682/241;
10
11 end

```

File for determining the temperature of clothing for the predicted mean value function.

```

1 function F = clothing_temp_fun(tclg,M,W,Icl,ta,tr,var,pa,fcl)
2 % function required for predicted mean vote (PMV); determines the surface
3 % temperature of clothing on an individual; refer to Fanger's work on PMV
4 % for details regarding the equations.
5
6 if 2.38*abs(tclg-ta)^(.25) > 12.1*sqrt(var)

```

```

7     hc = 2.38*abs(tclg - ta)^(.25);
8 elseif 2.38*abs(tclg-ta)^(.25) < 12.1*sqrt(var)
9     hc = 12.1*sqrt(var);
10 end
11
12 F = norm(35.7 - 0.028*(M-W) - Icl*(3.96*10^(-8)*fcl*((tclg + 273)^4 - (tr + 273)^4)
13     + fcl*hc*(tclg-ta)) - tclg);
14 end

```

File for determining future values of damper positions used in the horizon predictions of the dynamic MPC control method.

```

1 function [ud, lastInput_clg, lastInput_flow, Iterm_clg, Iterm_flow] = ...
2     determine_future_ud(n_rooms, T_room_start, T_room_pred, P_EDS_stpt_0, T_stpt_opt
3         ,...
4         lastInput_clg, lastInput_flow, max_vav_flow, timestep_PID, Iterm_clg, Iterm_flow
5     )
6 % Function for determining the future damper positions of the VAVs for the
7 % zones; used in the dynamic simulation for the prediction horizon
8 % associated with MPC.
9
10 %%% Controller Parameters
11 % Cooling loop
12 kp_clg = 20;
13 ki_clg = 1; % 5
14 kd_clg = 10; % 2
15 % Iterm_clg = zeros(1,10);
16 outMax_clg = 100;
17 outMin_clg = 0;
18 bias_clg = 0;
19 control_act_clg = 0; % reverse acting
20
21 % Flow loop
22 kp_flow = 0.07;
23 ki_flow = 0.15; % 0.08
24 kd_flow = 0.1;
25 % Iterm_flow = zeros(1,10);
26 outMax_flow = 1;
27 outMin_flow = 0;
28 bias_flow = 0;
29 control_act_flow = 1; % normal acting
30 input_flow = lastInput_flow;
31
32 %%% Simulate Damper Controls
33 for j=1:1:5 % simulate PID loops for 5 minutes
34     for i=1:1:n_rooms
35         input_clg = (T_room_start(i) - T_room_pred(i))*j/5 + T_room_start(i); %
36             feedback temperature
37         % cooling PID loop
38         [output_clg, lastInput_clg(i), Iterm_clg(i)] = pid_control(T_stpt_opt(i),
39             input_clg, ...
40             lastInput_clg(i), Iterm_clg(i), kp_clg, ...
41             ki_clg, kd_clg, outMax_clg, outMin_clg, bias_clg, control_act_clg,
42             timestep_PID);
43
44         setpoint_flow = output_clg/100; % flow setpoint for flow loop
45         % flow PID loop
46         [output_flow, lastInput_flow(i), Iterm_flow(i)] = pid_control(setpoint_flow,
47             input_flow(i), ...
48             lastInput_flow(i), Iterm_flow(i), kp_flow, ...
49             ki_flow, kd_flow, outMax_flow, outMin_flow, bias_flow, control_act_flow,
50             timestep_PID);
51
52         ud(i) = output_flow*100; % damper positions
53     end
54 end

```

```

46     end
47
48     % VAV flow predictions
49     vav1_flow = vav1(ud(1), P_EDS_stpt_0, []);
50     vav2_flow = vav2(ud(2), P_EDS_stpt_0, []);
51     vav3_flow = vav3(ud(3), P_EDS_stpt_0, []);
52     vav4_flow = vav4(ud(4), P_EDS_stpt_0, []);
53     vav5_flow = vav5(ud(5), P_EDS_stpt_0, []);
54     vav6_flow = vav6(ud(6), P_EDS_stpt_0, []);
55     vav7_flow = vav7(ud(7), P_EDS_stpt_0, []);
56     vav8_flow = vav8(ud(8), P_EDS_stpt_0, []);
57     vav9_flow = vav9(ud(9), P_EDS_stpt_0, []);
58     vav11_flow = vav11(ud(10), P_EDS_stpt_0, []);
59
60     vavs = [vav1_flow, vav2_flow, vav3_flow, vav4_flow, vav5_flow, vav6_flow, ...
61            vav7_flow, vav8_flow, vav9_flow, vav11_flow];
62
63     input_flow = vavs./max_vav_flow./2118.88; % feedback flow for next iteration
64 end
65 end

```

File for determining fan cost.

```

1 function [J_fan] = fan_cost (ud, P_EDS_STPT, ts)
2 % This function calculates the cost associated with the electricity used by
3 % the fan in the AHU.
4
5 % determine the individual zone flows
6 vav1_flow = vav1(ud(1), P_EDS_STPT, []);
7 vav2_flow = vav2(ud(2), P_EDS_STPT, []);
8 vav3_flow = vav3(ud(3), P_EDS_STPT, []);
9 vav4_flow = vav4(ud(4), P_EDS_STPT, []);
10 vav5_flow = vav5(ud(5), P_EDS_STPT, []);
11 vav6_flow = vav6(ud(6), P_EDS_STPT, []);
12 vav7_flow = vav7(ud(7), P_EDS_STPT, []);
13 vav8_flow = vav8(ud(8), P_EDS_STPT, []);
14 vav9_flow = vav9(ud(9), P_EDS_STPT, []);
15 vav11_flow = vav11(ud(10), P_EDS_STPT, []);
16 vavs = [vav1_flow, vav2_flow, vav3_flow, vav4_flow, vav5_flow, vav6_flow, ...
17        vav7_flow, vav8_flow, vav9_flow, vav11_flow];
18
19 % determine the total air flow through the AHU
20 q_cfm = sum(vavs);
21
22 % determine the cost of the electricity used by the fan
23 C_elec = 0.12; % cost of electricity [$/kWh]
24 um = 0.9; % fan motor efficiency
25 ub = 0.9; % fan belt efficiency
26 uf = 0.9; % fan blade efficiency
27
28 J_fan = P_EDS_STPT*(0.1175*C_elec*q_cfm*ts/60)/(1000*um*ub*uf);
29 end

```

File for calculating the optimization constraints for both methods.

```

1 function [c, ceq] = find_opt_constraints(x, ud, n_rooms, n_pred, last_TAHU,
2    last_PEDS, last_Tstpts)
3 % This function is used by fmincon to determine the constraints placed on
4 % the optimization and its variables.
5
6 if length(x) == 1
7     P_EDS = x; % if running baseline UBO file
8 else
9     T_stpts = x(1:n_rooms*n_pred); % if running optimization

```

```

9      T_AHU = x(n_rooms*n_pred + 1: n_rooms*n_pred + n_pred); % if running
      optimization
10     P_EDS = x(n_rooms*n_pred + n_pred + 1: n_rooms*n_pred + n_pred + n_pred); % if
      running optimization
11 end
12
13 % modify variables as needed if the optimization is for the dynmaic MPC
14 % method with the prediction horizon
15 if length(T_stpts) > n_rooms
16     for i=2:1:n_pred
17         for j=1:1:n_rooms
18             last_Tstpts(i,j) = T_stpts(i-1,j);
19         end
20     end
21 end
22
23 if length(T_AHU) > 1
24     for i=2:1:n_pred
25         last_TAHU(i) = T_AHU(i-1);
26     end
27 end
28
29 if length(P_EDS) > 1
30     for i=2:1:n_pred
31         last_PEDS(i) = P_EDS(i-1);
32     end
33 end
34
35 % determine the total air flow through the AHU for each end static pressure
36 % setpoint and damper position given
37 for i=1:1:length(P_EDS)
38     vav1_flow = vav1(ud(1,1), P_EDS(i), []);
39     vav2_flow = vav2(ud(2,1), P_EDS(i), []);
40     vav3_flow = vav3(ud(3,1), P_EDS(i), []);
41     vav4_flow = vav4(ud(4,1), P_EDS(i), []);
42     vav5_flow = vav5(ud(5,1), P_EDS(i), []);
43     vav6_flow = vav6(ud(6,1), P_EDS(i), []);
44     vav7_flow = vav7(ud(7,1), P_EDS(i), []);
45     vav8_flow = vav8(ud(8,1), P_EDS(i), []);
46     vav9_flow = vav9(ud(9,1), P_EDS(i), []);
47     vav11_flow = vav11(ud(10,1), P_EDS(i), []);
48
49     vavs = [vav1_flow, vav2_flow, vav3_flow, vav4_flow, vav5_flow, vav6_flow, ...
50            vav7_flow, vav8_flow, vav9_flow, vav11_flow];
51     vavs_flow(i) = sum(vavs);
52 end
53
54 % determine the constrain inequalities
55
56 % this inequality is the power constraint on the fan
57 for i=1:1:length(P_EDS)
58     c(i) = 0.1175*vavs_flow(i)*P_EDS(i)/0.9/0.9/0.9 - 910; % 7.5 HP = 5.59275 kW;
      910 comes from UBO data
59 end
60
61 % this inequality is a change in control action limit for the discharge air
62 % temepature setpoint
63 ind_start = length(c);
64 for i=ind_start + 1:1:ind_start + n_pred
65     c(i) = abs(last_TAHU(i-ind_start) - T_AHU(i-ind_start)) - 0.5;
66 end
67
68 % this inequality is a change in control action limit for the end static
69 % pressure setpoint

```

```

70 ind_start2 = length(c);
71 for i=ind_start2 + 1:ind_start2 + n_pred
72     c(i) = abs(last_PEDS(i-ind_start2) - P_EDS(i-ind_start2)) - 0.15;
73 end
74
75 % there are no equality constraints
76 ceq = [];
77 end

```

File for determining the air temperature that would provide a value of zero PMV.

```

1 function [error] = find_Ta_for_zero_PMV(ta,rh)
2 % This function determines the zone air temperature that will give a PMV
3 % value equal to zero.
4
5 % Parameters for the PMV equation. Refer to Fanger's work.
6 Icl = 0.75;
7 var = 0.2;
8 M = 70;
9 W = 0;
10 tr = ta;
11 tro = tr;
12
13 [PMV,~,~,~,~,~,~] = predicted_mean_vote(M,W,Icl,ta,tr,var,rh);
14
15 error = PMV^2;
16 end

```

File for determining the air temperature that would provide a value of -0.5 PMV.

```

1 function [error] = find_Ta_for_zero_PMV_neg(ta,rh)
2 % This function determines the zone air temperature that will give a PMV
3 % value equal to -0.5.
4
5 % Parameters for the PMV equation. Refer to Fanger's work.
6 Icl = 0.75;
7 var = 0.2;
8 M = 70;
9 W = 0;
10 tr = ta;
11 tro = tr;
12
13 [PMV,~,~,~,~,~,~] = predicted_mean_vote(M,W,Icl,ta,tr,var,rh);
14
15 error = (-0.5 - PMV)^2;
16 end

```

File for simulating a PID controller for the local control in the UBO.

```

1 function [output, lastInput, lterm] = pid_control(setpoint, input, lastInput, lterm,
2     kp, ki, kd, outMax, outMin, bias, controller_action, timestep)
3 % This function simulates a PID controller with the supplied parameters.
4
5 % calculate relevant values for controller
6 error = setpoint - input; % calculate error
7 dInput = (input - lastInput)/timestep; % change from last input
8
9 if controller_action == 1; % direct acting
10     error = error;
11     dInput = dInput;
12 elseif controller_action == 0; % reverse acting
13     error = -error;
14     dInput = -dInput;
15 end

```

```

15
16 lastTerm = Iterm; % the previous intergral term
17 Iterm = Iterm + ki*error*timestep; % determine new integral term
18
19 output = kp*error + Iterm + kd*dInput + bias; % determine output of controller
20
21 % clamping to prevent windup of controller
22 if output > outMax
23     output = outMax;
24     Iterm = lastIterm;
25 elseif output < outMin
26     output = outMin;
27     Iterm = lastIterm;
28 end
29
30 lastInput = input; % store last input for next iteration of controller
31
32 end

```

File for calculating the predicted mean vote.

```

1 function [PMV,PPD,tcl,fval,fcl,hc,Icl,M] = predicted_mean_vote(M,W,Icl,ta,tr,var,rh)
2 % This function is the coded version of Fanger's Predicted Mean Vote (PMV)
3 % calculation.
4
5 % required and not required variables and their names for PMV calculation
6 M = M*1.1622; % metabolic rate (W/m^2)
7 W = 0; % effective mechanical power (W/m^2)
8 Icl = Icl*0.155; % clothing insulation (m^2K/W)
9 fcl = ; % clothing surface area factor
10 ta = 32; % air temperature (Celsius)
11 tr = ta; % mean radiant temperature (Celsius)
12 var = 0.15; % relative air velocity (m/s)
13 rh = 0.5; % relative humidity (%)
14 pa = (9.272523*10^(-5)*(ta + 273)^3 - 0.07804973*(ta + 273)^2 + 21.99986*(ta + 273)
15     - 2075.53)*1000*rh;
16 pa = exp(20.386 - 5132/(ta+273))*133.322368; % water vapor partial pressure (Pa)
17 pa = 0.5*(exp(-16.6536 - 4030.183/(ta + 235)));
18 hc = ; % convective heat transfer coefficient (W/(m^2K))
19 tclg = ta + 2; % clothing surface temperature guess (Celsius)
20
21 if Icl <= 0.078
22     fcl = 1 + 1.290*Icl;
23 elseif Icl > 0.078
24     fcl = 1.05 + 0.645*Icl;
25 end
26
27 % optimiztion to determine clothing temperature
28 f = @(tclg) clothing_temp_fun(tclg,M,W,Icl,ta,tr,var,pa,fcl);
29 opts=optimset('display','none','LargeScale','off');
30 [tcl,fval] = fminunc(f,tclg,opts);
31
32 if 2.38*abs(tclg-ta)^(.25) > 12.1*sqrt(var)
33     hc = 2.38*abs(tclg - ta)^(.25);
34 elseif 2.38*abs(tclg-ta)^(.25) < 12.1*sqrt(var)
35     hc = 12.1*sqrt(var);
36 end
37
38 % predicted mean vote calculation
39 PMV = (0.303*exp(-0.036*M) + 0.028)*(M - W - 3.05*10^(-3)*(5733 - 6.99*(M-W) - pa) -
40     0.42*(M - W - 58.15)...
41     - 1.7*10^(-5)*M*(5867 - pa) - 0.0014*M*(34 - ta) - 3.96*10^(-8)*fcl*((tcl + 273)
42     ^4 - (tr + 273)^4) - fcl*hc*(tcl - ta));

```

```

41
42 % percent of population dissatisfied with environmental conditions based on
43 % PMV value
44 PPD = 100 - 95*exp(-0.03353*PMV^4 - 0.2179*PMV^2);

```

File for determining the zone level discomfort cost.

```

1 function [J_room, PMV, error, dLOP_dpmv, R] = room_cost(ts, pyear, ta, rh,
    troom_stpt, troom)
2 % This function calculates the economic cost of loss of productivity (LOP)
3 % due to discomfort in a zone.
4
5 % fit of PMV
6 PMV = 0.5542*rh + 0.23*ta - 5.44;
7 % partial derivative of PMV; sensitivity to changes in air temperature
8 dpmv_dta = 0.23;
9
10 % regression coefficients for LOP fit
11 b01 = 1.2802070; b02 = -0.153979397;
12 b11 = 15.995451; b12 = 3.8820297;
13 b21 = 31.507402; b22 = 25.176447;
14 b31 = 11.754937; b32 = -26.641366;
15 b41 = 1.4737526; b42 = 13.110120;
16 b51 = 0; b52 = -3.1296854;
17 b61 = 0; b62 = 0.29260920;
18
19 % determine LOP sensitivity to PMV
20 if PMV < -0.5 % occupants too cold
21     dLOP_dpmv = b11 + 2*b21*PMV + 3*b31*PMV^2 + 4*b41*PMV^3 + 5*b51*PMV^4 + 6*b61*
        PMV^5;
22 elseif PMV > 0 % occupants too warm
23     dLOP_dpmv = b12 + 2*b22*PMV + 3*b32*PMV^2 + 4*b42*PMV^3 + 5*b52*PMV^4 + 6*b62*
        PMV^5;
24 else % zero LOP in PMV range of -0.5 to 0
25     dLOP_dpmv = 0;
26 end
27
28 % sensitivity of economic cost to change in LOP
29 dmoney_dLOP = pyear*ts/60/52/40; % sign always positive
30
31 % determine objective function values
32 R = dpmv_dta*dLOP_dpmv*dmoney_dLOP;
33 error = troom - troom_stpt;
34
35 % return economic cost of LOP due to discomfort
36 J_room = error*R;

```

File for determining the zone 1 air volume flow.

```

1 function [varout] = vav1( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.07766; b = 16.08;
7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
12 end
13
14 end

```

File for determining the zone 2 air volume flow.

```
1 function [varout] = vav2( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.02071; b = 4.288;
7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
12 end
13
14 end
```

File for determining the zone 3 air volume flow.

```
1 function [varout] = vav3( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.03849; b = 7.969;
7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
12 end
13
14 end
```

File for determining the zone 4 air volume flow.

```
1 function [varout] = vav4( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.1144; b = 23.69;
7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
12 end
13
14 end
```

File for determining the zone 5 air volume flow.

```
1 function [varout] = vav5( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.04725; b = 9.783;
7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
```

```

12 end
13
14 end

```

File for determining the zone 6 air volume flow.

```

1 function [varout] = vav6( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.06338; b = 13.12;
7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
12 end
13
14 end

```

File for determining the zone 7 air volume flow.

```

1 function [varout] = vav7( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.07664; b = 15.87;
7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
12 end
13
14 end

```

File for determining the zone 8 air volume flow.

```

1 function [varout] = vav8( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.03224; b = 6.675;
7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
12 end
13
14 end

```

File for determining the zone 9 air volume flow.

```

1 function [varout] = vav9( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.2522; b = 52.22;

```

```

7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
12 end
13
14 end

```

File for determining the zone 10 air volume flow.

```

1 function [varout] = vav11( vav_dmpr, P_EDS, vav_flow )
2 % This function returns the VAV airflow value when the vav_flow input is
3 % left empty.
4
5 % fit coefficients
6 a = -0.08071; b = 16.71;
7
8 if isempty(vav_flow)
9     varout = (a*vav_dmpr^2 + b*vav_dmpr)*sqrt(P_EDS);
10 else
11     varout = [];
12 end
13
14 end

```

APPENDIX F

MATLAB APOGEE INTERFACE

To interface with the Texas A&M University's building energy management system, the OPC toolbox within MATLAB was used. OPC is an open communication protocol used by the HVAC industry. Instructions and code for establishing connections to the Apogee system as well as recording data are detailed in the following sections.

F.1 Interface Instructions

Within the MATLAB OPC toolbox there is a function called "opctool". This is the manual way to interface with the OPC server. The tutorial provided by MATLAB at <https://www.mathworks.com/help/opc/ug/access-data-with-opc-data-access-explorer.html> covers in detail the steps to manually connect to the OPC server and examine the Apogee points that are available.

F.2 Datalogging Code

I developed some MATLAB code that utilizes the functions within the OPC toolbox to read from, write to, and record data from points within Apogee. The respective code, with comments, is shown below. At the end of this section I include an example MATLAB m-file that details recording data at a regular rate.

File for reading the value of a single setpoint or the values of a group of setpoints from Apogee.

```
1 function [setpoint_value] = opc_read_setpoint(setpoint)
2 % This function will read the values associated with the respective Apogee
3 % points defined in "setpoint".
4
5 try
6     hostInfo = opcserverinfo('AP-Server.tamu.edu'); % Gets the info about the server
7     , bln3.tamu.edu
8     allServers = hostInfo.ServerID'; % Grabs just the server ID, to be used later
9
10    da = opcda('AP-Server.tamu.edu', allServers{1}); % This creates an OPC data
11    % access client object for our server
12    da.Timeout = 30; % This is the timeout value to close the connection if a
13    % process is taking too long
14
15    connect(da); % This connects the client, allowing us to communicate with the
16    % server
17
18    grp = addgroup(da); % This creates an OPC data access group object, which will
19    % contain item objects
20
21    itms = additem(grp, {setpoint}); % for example, WCAMPUS4.VAV.2:CTL STPT_CV
22
23    data = read(itms); % This reads the values for the points defined in itms
24
25    setpoint_value = data.Value; % Data is a structure with several items
26    disconnect(da); % Be sure to disconnect from the server once you are finished
```

```

23 catch err
24     %open file
25     fid = fopen('logFile.txt','a+t');
26     % write the error to file
27     % first line: message
28     fprintf(fid, '%s\n', err.message);
29
30     % following lines: stack
31     for e=1:length(err.stack)
32         fprintf(fid, '%s\r\n', err.getReport('extended', 'hyperlinks','off'));
33     end
34
35     % close file
36     fclose(fid);
37     setpoint_value = '';
38 end
39 end

```

File for writing a value to a single setpoint or values to a group of setpoints from Apogee.

```

1 function [] = opc_write_setpoint(setpoints, stpt_values)
2 % This function will write the values "stpt_values" to the respective
3 % points within Apogee defined by "setpoints".
4
5 try
6     hostInfo = opcserverinfo('AP-Server.tamu.edu'); % Gets the info about the server
7     , bln3.tamu.edu
8     allServers = hostInfo.ServerID'; % Grabs just the server ID, to be used later
9
10    da = opcda('AP-Server.tamu.edu', allServers{1}); % This creates an OPC data
11    % access client object for our server
12    da.Timeout = 30; % This is the timeout value to close the connection if a
13    % process is taking too long
14
15    connect(da); % This connects the client, allowing us to communicate with the
16    % server
17
18    grp = addgroup(da); % This creates an OPC data access group object, which will
19    % contain item objects
20
21    itms = additem(grp, {setpoints}); % for example, WCAMPUS4.VAV.2:CTL STPT_CV
22
23    write(itms, stpt_values); % This writes the values (stpt_values) to the points
24    % defined in itms
25    disconnect(da); % Be sure to disconnect from the server once you are finished
26
27 catch err
28     %open file
29     fid = fopen('logFile.txt','a+t');
30     % write the error to file
31     % first line: message
32     fprintf(fid, '%s\n', err.message);
33
34     % following lines: stack
35     for e=1:length(err.stack)
36         fprintf(fid, '%s\r\n', err.getReport('extended', 'hyperlinks','off'));
37     end
38
39     % close file
40     fclose(fid);
41 end
42 end

```

File for reading the values of a group of setpoints from Apogee.

```

1 function [data] = opc_get_data(items)
2 % This function will retrieve a set of data for the defined Apogee points
3 % given by "items".
4
5 try
6     hostInfo = opcserversinfo('AP-Server.tamu.edu'); % Gets the info about the server
7     , bln3.tamu.edu
8     allServers = hostInfo.ServerID'; % Grabs just the server ID, to be used later
9
10    da = opcda('AP-Server.tamu.edu', allServers{1}); % This creates an OPC data
11        access client object for our server
12    da.Timeout = 30;
13
14    connect(da); % This connects the client, allowing us to communicate with the
15        server
16
17    grp = addgroup(da); % This creates an OPC data access group object, which will
18        contain item objects
19
20    itms = additem(grp, items); % This adds data access items to a dagroup object.
21    data = read(grp); % This command grabs data for all the items in the OPC data
22        access group object 'grp'
23    disconnect(da); % Be sure to disconnect from the server once you are finished
24
25 catch err
26     %open file
27     fid = fopen('logfile.txt','a+t');
28     % write the error to file
29     % first line: message
30     fprintf(fid, '%s\n', err.message);
31
32     % following lines: stack
33     for e=1:length(err.stack)
34         fprintf(fid, '%s\r\n', err.getReport('extended', 'hyperlinks','off'));
35     end
36
37     % close file
38     fclose(fid);
39     data = '';
40 end
41 end

```

Example file in which I autorecorded data at a set interval from a group of points within Apogee.

```

1 % Program to be run automatically everyday to record cascaded loop
2 % performance
3
4 clear all; close all;
5 %%
6
7 n = 1*260*4*2; % Number of data points you wish to have
8 dt = 15; % The update rate, in seconds, that you wish to read the data
9 start_hour = 17; % Hour during which the bias values will be written and data
10    recording will start
11 start_minute = 51; % Minute during which the bias values will be written and data
12    recording will start
13
14 items = {'CAMPUS_OADBT.M_CV', 'CAMPUS_OADWP_CV', ...
15         '1497_AH1.SVF_CV', '1497_AH1.DAT_CV', '1497_AH1.DAT.S_CV', '1497_AH1.OAD_CV', '
16         1497_AH1.RAD_CV', '1497_AH1.EDS_CV', '1497_AH1.EDS.S_CV', '1497_AH1.CCV_CV',
17         ...
18         '1497_CHW.PST_CV', '1497_CHW.PRT_CV', '1497_CHW.BPV_CV', '1497_HHW.PST_CV', '1497

```

```

    HHW.BPV_CV',...
15 'WCAMPUS4.VAV.1:AIR VOLUME_CV', 'WCAMPUS4.VAV.1:AUX TEMP_CV', 'WCAMPUS4.VAV.1:
    CTL STPT_CV', 'WCAMPUS4.VAV.1:ROOM TEMP_CV', 'WCAMPUS4.VAV.1:VLV1 COMD_CV',...
16 'WCAMPUS4.VAV.1:VLV1 POS_CV', 'WCAMPUS4.VAV.1:DMPR COMD_CV', 'WCAMPUS4.VAV.1:
    DMPR POS_CV', 'WCAMPUS4.VAV.1:HEAT.COOL_CV', 'WCAMPUS4.VAV.1:DAY.NGT_CV',...
17 'WCAMPUS4.VAV.2:AIR VOLUME_CV', 'WCAMPUS4.VAV.2:AUX TEMP_CV', 'WCAMPUS4.VAV.2:
    CTL STPT_CV', 'WCAMPUS4.VAV.2:ROOM TEMP_CV', 'WCAMPUS4.VAV.2:VLV1 COMD_CV',...
18 'WCAMPUS4.VAV.2:VLV1 POS_CV', 'WCAMPUS4.VAV.2:DMPR COMD_CV', 'WCAMPUS4.VAV.2:
    DMPR POS_CV', 'WCAMPUS4.VAV.2:HEAT.COOL_CV', 'WCAMPUS4.VAV.2:DAY.NGT_CV',...
19 'WCAMPUS4.VAV.3:AIR VOLUME_CV', 'WCAMPUS4.VAV.3:AUX TEMP_CV', 'WCAMPUS4.VAV.3:
    CTL STPT_CV', 'WCAMPUS4.VAV.3:ROOM TEMP_CV', 'WCAMPUS4.VAV.3:VLV1 COMD_CV',...
20 'WCAMPUS4.VAV.3:VLV1 POS_CV', 'WCAMPUS4.VAV.3:DMPR COMD_CV', 'WCAMPUS4.VAV.3:
    DMPR POS_CV', 'WCAMPUS4.VAV.3:HEAT.COOL_CV', 'WCAMPUS4.VAV.3:DAY.NGT_CV',...
21 'WCAMPUS4.VAV.4:AIR VOLUME_CV', 'WCAMPUS4.VAV.4:AUX TEMP_CV', 'WCAMPUS4.VAV.4:
    CTL STPT_CV', 'WCAMPUS4.VAV.4:ROOM TEMP_CV', 'WCAMPUS4.VAV.4:VLV1 COMD_CV',...
22 'WCAMPUS4.VAV.4:VLV1 POS_CV', 'WCAMPUS4.VAV.4:DMPR COMD_CV', 'WCAMPUS4.VAV.4:
    DMPR POS_CV', 'WCAMPUS4.VAV.4:DAY.NGT_CV', 'WCAMPUS4.VAV.4:HEAT.COOL_CV',...
23 'WCAMPUS4.VAV.5:AIR VOLUME_CV', 'WCAMPUS4.VAV.5:AUX TEMP_CV', 'WCAMPUS4.VAV.5:
    CTL STPT_CV', 'WCAMPUS4.VAV.5:ROOM TEMP_CV', 'WCAMPUS4.VAV.5:VLV1 COMD_CV',...
24 'WCAMPUS4.VAV.5:VLV1 POS_CV', 'WCAMPUS4.VAV.5:DMPR COMD_CV', 'WCAMPUS4.VAV.5:
    DMPR POS_CV', 'WCAMPUS4.VAV.5:HEAT.COOL_CV', 'WCAMPUS4.VAV.5:DAY.NGT_CV',...
25 'WCAMPUS4.VAV.6:AIR VOLUME_CV', 'WCAMPUS4.VAV.6:AUX TEMP_CV', 'WCAMPUS4.VAV.6:
    CTL STPT_CV', 'WCAMPUS4.VAV.6:ROOM TEMP_CV', 'WCAMPUS4.VAV.6:VLV1 COMD_CV',...
26 'WCAMPUS4.VAV.6:VLV1 POS_CV', 'WCAMPUS4.VAV.6:DMPR COMD_CV', 'WCAMPUS4.VAV.6:
    DMPR POS_CV', 'WCAMPUS4.VAV.6:HEAT.COOL_CV', 'WCAMPUS4.VAV.6:DAY.NGT_CV',...
27 'WCAMPUS4.VAV.7:AIR VOLUME_CV', 'WCAMPUS4.VAV.7:AUX TEMP_CV', 'WCAMPUS4.VAV.7:
    CTL STPT_CV', 'WCAMPUS4.VAV.7:ROOM TEMP_CV', 'WCAMPUS4.VAV.7:VLV1 COMD_CV',...
28 'WCAMPUS4.VAV.7:VLV1 POS_CV', 'WCAMPUS4.VAV.7:DMPR COMD_CV', 'WCAMPUS4.VAV.7:
    DMPR POS_CV', 'WCAMPUS4.VAV.7:HEAT.COOL_CV', 'WCAMPUS4.VAV.7:DAY.NGT_CV',...
29 'WCAMPUS4.VAV.8:AIR VOLUME_CV', 'WCAMPUS4.VAV.8:AUX TEMP_CV', 'WCAMPUS4.VAV.8:
    CTL STPT_CV', 'WCAMPUS4.VAV.8:ROOM TEMP_CV', 'WCAMPUS4.VAV.8:VLV1 COMD_CV',...
30 'WCAMPUS4.VAV.8:VLV1 POS_CV', 'WCAMPUS4.VAV.8:DMPR COMD_CV', 'WCAMPUS4.VAV.8:
    DMPR POS_CV', 'WCAMPUS4.VAV.8:HEAT.COOL_CV', 'WCAMPUS4.VAV.8:DAY.NGT_CV',...
31 'WCAMPUS4.VAV.9:AIR VOLUME_CV', 'WCAMPUS4.VAV.9:AUX TEMP_CV', 'WCAMPUS4.VAV.9:
    CTL STPT_CV', 'WCAMPUS4.VAV.9:ROOM TEMP_CV', 'WCAMPUS4.VAV.9:VLV1 COMD_CV',...
32 'WCAMPUS4.VAV.9:VLV1 POS_CV', 'WCAMPUS4.VAV.9:DMPR COMD_CV', 'WCAMPUS4.VAV.9:
    DMPR POS_CV', 'WCAMPUS4.VAV.9:HEAT.COOL_CV', 'WCAMPUS4.VAV.9:DAY.NGT_CV',...
33 'WCAMPUS4.VAV.11:AIR VOLUME_CV', 'WCAMPUS4.VAV.11:AUX TEMP_CV', 'WCAMPUS4.VAV
    .11:CTL STPT_CV', 'WCAMPUS4.VAV.11:ROOM TEMP_CV',...
34 'WCAMPUS4.VAV.11:DMPR COMD_CV', 'WCAMPUS4.VAV.11:DMPR POS_CV', 'WCAMPUS4.VAV
    .11:HEAT.COOL_CV', 'WCAMPUS4.VAV.11:DAY.NGT_CV', '1497_AH1.DATLOOP2.ILSP_CV'
    ,...
35 '1497_AH1.DMDCLG_CV', '1497_AH1.DMEDEDS_CV'}; % List of points to gather data
    for
36
37 cur_time = clock;
38 cur_hour = cur_time(4);
39 cur_minute = cur_time(5);
40
41 % Wait until the defined time to start recording data
42 while ~(cur_hour == start_hour && cur_minute == start_minute)
43     display('Not time yet...');
44     pause(25);
45     cur_time = clock;
46     cur_hour = cur_time(4);
47     cur_minute = cur_time(5);
48 end
49
50 % Values that were to be set for the recorded time period
51 iL_type = 0;
52 iL_P = 4000;
53 iL_I = 0;
54 iL_D = 0;
55 iL_time = 1;

```

```

56 iL_bias = 40;
57 iL_min = 0;
58 iL_max = 100;
59 oL_type = 128;
60 oL_P = 1250;
61 oL_I = 13.2;
62 oL_D = 0;
63 oL_time = 15;
64 oL_bias = 60;
65 oL_min = 45;
66 oL_max = 75;
67
68 % Writing specific values regarding this data collection method
69 set_cascaded_loop_gains(iL_type,iL_P,iL_I,iL_D,iL_time,iL_bias,iL_min,iL_max,oL_type
    ,oL_P,oL_I,oL_D,oL_time,oL_bias,oL_min,oL_max);
70 opc_write_setpoint('1497_AH1.DAT.S_PRIOC',5);
71 opc_write_setpoint('1497_AH1.DAT.S_CV',55);
72
73 iL_bias = opc_read_setpoint('1497_AH1.CCV_CV');
74 pause(10);
75 iL_bias = opc_read_setpoint('1497_AH1.CCV_CV');
76 while iL_bias == 0
77     pause(10);
78     iL_bias = opc_read_setpoint('1497_AH1.CCV_CV');
79     try
80         send_text_message('719-930-7006','Sprint','Zero reading for CCV.')
81     catch
82         disp('Error sending message. ');
83     end
84 end
85
86 opc_write_setpoint('1497_AH1.DATLOOP.BIAS_PRIOC',5);
87 opc_write_setpoint('1497_AH1.DATLOOP.BIAS_CV',iL_bias);
88
89 oL_bias = opc_read_setpoint('1497_AH1.DAT_CV');
90 pause(10);
91 oL_bias = opc_read_setpoint('1497_AH1.DAT_CV');
92 while iL_bias == 0
93     pause(10);
94     iL_bias = opc_read_setpoint('1497_AH1.DAT_CV');
95 end
96
97 opc_write_setpoint('1497_AH1.DATLOOP2.BIAS_PRIOC',5);
98 opc_write_setpoint('1497_AH1.DATLOOP2.BIAS_CV',oL_bias);
99
100 % To record at a regular rate, it is necessary to know the duration it
101 % takes to read the data as it can vary. With this time known, the data
102 % recording can happen at regular intervals.
103
104 beg_time = clock;
105
106 for i=1:n
107     tic;
108
109     disp('Reading data...')
110     data{i,1} = opc_get_data(items); % Read data for items from server
111     m=size(items,2);
112     ser_time = now;
113     for j=1:m
114         data{i,1}(j).time = ser_time;
115     end
116     if ~isequal(i,n) % If we are not on the last case, then continue with the timing
117         pause
118         text = 'Accesed data for iteration %d. Data logging operation %2.2f%%'

```

```

118         complete, time remaining: %2.0f hours %2.0f minutes\n';
119         fprintf(text,i,i/n*100,fix((n-i)*dt/3600),mod((n-i)*dt,3600)/60)
120         t_end = toc;
121         fprintf('Time: %2.2f%\n',t_end)
122         pause(dt-t_end);
123     else % Last iteration, thus no pause is needed.
124         text = 'Accesed data for iteration %d.\n Data logging COMPLETE.\n Beginning
125               time: %2.2f.\n Current time: %2.2f.\n';
126         cur_time = clock;
127         fprintf(text,i,beg_time,cur_time)
128     end
129 end
130 % Saving the data to a specific file name
131 file_name = strcat('cascaded_control_data/',num2str(cur_time(2)),'.',num2str(
132     cur_time(3)),'.',num2str(cur_time(1)),'_cascaded_loop_on.mat');
133 save(file_name,'data','iL_bias','oL_bias');
quit;

```