Offline Model Predictive Control of Mixed Mode Buildings for Near-Optimal Supervisory Control Strategy Development

by

Peter T. May-Ostendorp

B.A., Dartmouth College, 2003

B.E., Thayer School of Engineering, 2004

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Civil, Environmental, and Architectural Engineering

2012

This thesis entitled:
Offline Model Predictive Control of Mixed Mode Buildings for Near-Optimal Supervisory Control Strategy Development
written by Peter T. May-Ostendorp
has been approved for the Department of Civil, Environmental, and Architectural Engineering

Gregor P. Henze, Ph.D., P.E.

Balaji Rajagopalan, Ph.D.

Michael J. Brandemuehl, Ph.D., P.E.

Moncef Krarti, Ph.D., P.E.

Gail S. Brager, Ph.D.

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

May-Ostendorp, Peter T. (Ph.D.)

Offline Model Predictive Control of Mixed Mode Buildings for Near-Optimal Supervisory Control Strategy Development

Thesis directed by Prof. Gregor P. Henze, Ph.D., P.E.

Model predictive control (MPC) is a powerful technique that can be used to reduce the operational cost, energy consumption, and environmental footprint of buildings. MPC optimizes control decisions to minimize the objective function produced by a building energy model and has been successfully applied to a range of control problems in buildings, usually thermal mass storage. Parametric simulation studies are typically conducted, and the resulting solution patterns are used to inform control strategies. A model predictive controller can also directly control building equipment, but in order to achieve faster solution convergence needed for real-time implementation, reduced-order gray- and black-box models are often employed that can be optimized through linear or quadratic programming.

Despite the widespread potential for thermal mass control in buildings, MPC of this kind is challenging to implement due to the necessity of reduced-order models and the need to integrate with building automation systems (BAS). This dissertation examines the possibility of using MPC conducted on white-box building energy models—the same types used to evaluate building designs—to develop datasets from which near-optimal control rules can be extracted using supervised learning techniques. This allows for the development of custom supervisory controllers that more closely approximate optimal energy and thermal comfort results compared to conventional control heuristics. Rules are developed in such a form that they can be implemented in a conventional BAS. The dissertation uses the case of mixed-mode (MM) buildings to test these techniques. A proof-of-concept rule extraction case is first presented for a simple binary natural ventilation control problem to test the utility of several data mining and statistical techniques to the problem, including generalized linear models (GLM), classification and regression trees (CART) and adaptive boosting. Next, a simulation study is conducted to explore a variety of more complex MM optimal control problems on four different MM building types and in five different climates. Two of these cases form the training set for further rule extraction, testing the applicability of this technique beyond simple binary decisions. CARTs were found to be successful in reproducing optimal supervisory control sequences, often yielding greater than 90% of optimizer energy savings with minimal thermal comfort consequences. Robustness of extracted rules and

generalizability to broader cases (e.g. other building types and climates) is examined. Finally, an experiment is presented in which the energy and comfort performance of extracted rules are tested on a radiantly cooled test cell. The impacts of model calibration mismatch and weather forecast uncertainty are examined and are found to contribute significantly to the reduced experimental performance of the rules.

The research provides two key outcomes for the larger building community. For designers of MM buildings, the simulation study provides for the first survey of MM performance under optimal control and identifies preferred strategies by climate and building type. For building control engineers, the rule extraction framework provides a new and innovative means for analyzing MPC solutions and implementing near-optimal rules based on those solutions. The research presents the first step in what will hopefully be a new vein of building controls research and eventually, controls practice. Future research must further examine the robustness of the approach and its operational performance in "live" buildings.

To Gregor for the inspiration, to Mariah for sustaining me, and to Tristan, Sophie, and Finnian for the many welcome distractions.

Humankind has not woven the web of life.

We are but one thread within it.

Whatever we do to the web, we do to ourselves.

All things are bound together.

All things connect.

– Chief Seattle –

Acknowledgements

Any doctoral student travels a long road, and fortunately mine was lined with many helpful friends, colleagues, and advisors. I would first like to thank the USGBC and ASHRAE for the funding that helped sustain a doctoral student and his family throughout the research. At the Fraunhofer ISE, Jens Pfafferott, Doreen Kalz, Martin Fischer, and Jan Mehnert made our experiments possible. I admire and appreciate the many dialogues shared with Brian Coffey, whose UC Berkeley dissertation is mandatory reading for anyone interested in this research. Charles Corbin not only served as an inspiration for quality research, but developed the foundation for the MPC software used throughout my work. My investigations would have been impossible without his help, and for this I am forever indebted. Balaji Rajagopalan cultivated the work around rule extraction, which has blossomed into one of the most rewarding pieces of this research. Gregor Henze unconditionally supported this work through the roughest of periods, always sharing resources, advice, and opportunities to maximize the research and my experience.

It is difficult to imagine how one manages a Ph.D. without this kind of support, but for me it is impossible to comprehend where this research or I personally would be without my family. They sacrificed a good deal to see me reach this goal, and I will never forget their unwavering support.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction, Motivation, and Organization

Buildings represent a significant fraction of current electric energy consumption and demand, consuming over 40% of primary energy in the United States and generating a proportionate share of greenhouse gas emissions [95]. One of the most cost-effective means to improve energy efficiency, cut utility expenditures, and reduce emissions associated with commercial buildings is through improved control. A 2005 Pacific Northwest National Laboratory report on advanced control and automation in commercial buildings estimated that improved operational strategies could reduce the total primary energy use of buildings by about 6% [19]. A later study by Lawrence Berkeley National Laboratory for the California Energy Commission indicated that retro-commissioning of commercial buildings across the US has generated cost-effective energy savings of 13–16% [73].

Simultaneous to this heightened interest in improving operational efficiencies, the building design and services sector is seeing increased interest in high-performance building designs that inherently benefit from more advanced control strategies. Mixed mode buildings are the example used throughout this dissertation. Mixed mode (MM) buildings represent a hybrid approach to space conditioning, employing a combination of natural ventilation and mechanical systems and intelligently switching between the two to minimize energy use, while preserving occupant comfort. MM is compatible with a variety of mechanical cooling system choices, ranging from conventional vapor compression air systems to ground- or cooling tower-coupled radiant cooling. They have demonstrated reductions in cooling- and ventilation-related energy use from 20% to 50% over code buildings [91, 44] and consistently outperform conventional buildings on thermal comfort and

occupant satisfaction [15]. However, the performance gains promised by MM buildings hinge to a large degree on their controls. The effectiveness of the MM control strategy directly determines the extent to which natural ventilation is able to displace mechanical cooling and ventilation systems.

Figure 1.1: The California Academy of Sciences, located in San Francisco, is one of the most notable public mixed mode buildings in the United States. Source: Wikimedia Commons.

This dissertation examines MM control through the framework of model predictive control, then applies data analysis techniques in a novel manner to reduce simulated optimal control sequences to rules that can be implemented in practice.

1.1 Mixed Mode Control Today

MM building controls have generally been classified into three topologies. Under **zoned** control, natural ventilation and mechanical conditioning are allowed to occur simultaneously, but in different zones of the building. For example, perimeter offices may be naturally ventilated and core zones mechanically conditioned. In **concurrent** operation, natural ventilation and mechanical conditioning may operate in the same space at the same time. Finally, **changeover** control allows natural ventilation and mechanical conditioning in the same space, but never at the same time. Most MM buildings will not fall cleanly into one of these categories, mainly because at least some amount of zoning is require to provide dedicated mechanical cooling to certain high load spaces like server rooms [16].

In the US, design guidelines and best practices for MM buildings have not yet been codified by professional building services organizations. Pioneering research in Europe, such as the Inter-

national Energy Agency's HybVent project [44], has helped propel MM more into the mainstream. For example, the Chartered Institute of Building Services Engineers (CIBSE) now publishes two application manuals related to MM and naturally ventilated buildings [24, 23]. However, even in Europe, there is no consensus on best practices for MM controls. As such, engineers are left to "start from scratch" or rely on intuition in developing control sequences for these buildings. Algorithms usually involve a series of simple heuristics and if/then statements developed by an HVAC designer for the building's sequence of operations. For example, "if the outdoor temperature drops below 68 °F, open all automated windows and turn off mechanical cooling." An example of one such algorithm is provided in Figure 1.2, in which various logical comparisons are made against the average zone temperature of the building to determine whether openings in the façade should be made.

It should be noted that most MM buildings are not fully automated, and occupants are usually responsible for operating windows in office spaces. This adaptive approach can reduce the complexity of the control system and has been shown to improve occupant thermal comfort by affording them greater latitude to adapt to thermal disturbances [33, 15]. However, introduction of occupant-controlled windows can also undermine the energy savings of MM buildings, since people cannot be expected to operate their windows in an energy-efficient manner all of the time. As a result, some MM buildings incorporate informational systems, such as notification lights, to signal to occupants when windows should be opened [72]. As with automated control, heuristics tend to guide notification systems.

The frequent use of passive thermal energy storage strategies like night flush ventilation in MM buildings suggests that these systems could benefit from more advanced control strategies like model predictive control (MPC) to maximize the use of free cooling opportunities. MPC can be used to optimize window positions, mechanical system operation, or both simultaneously, and can serve as a useful benchmark against which known control heuristics and topologies can be compared.

MPC is a control methodology that seeks strategies through time that minimize an objective

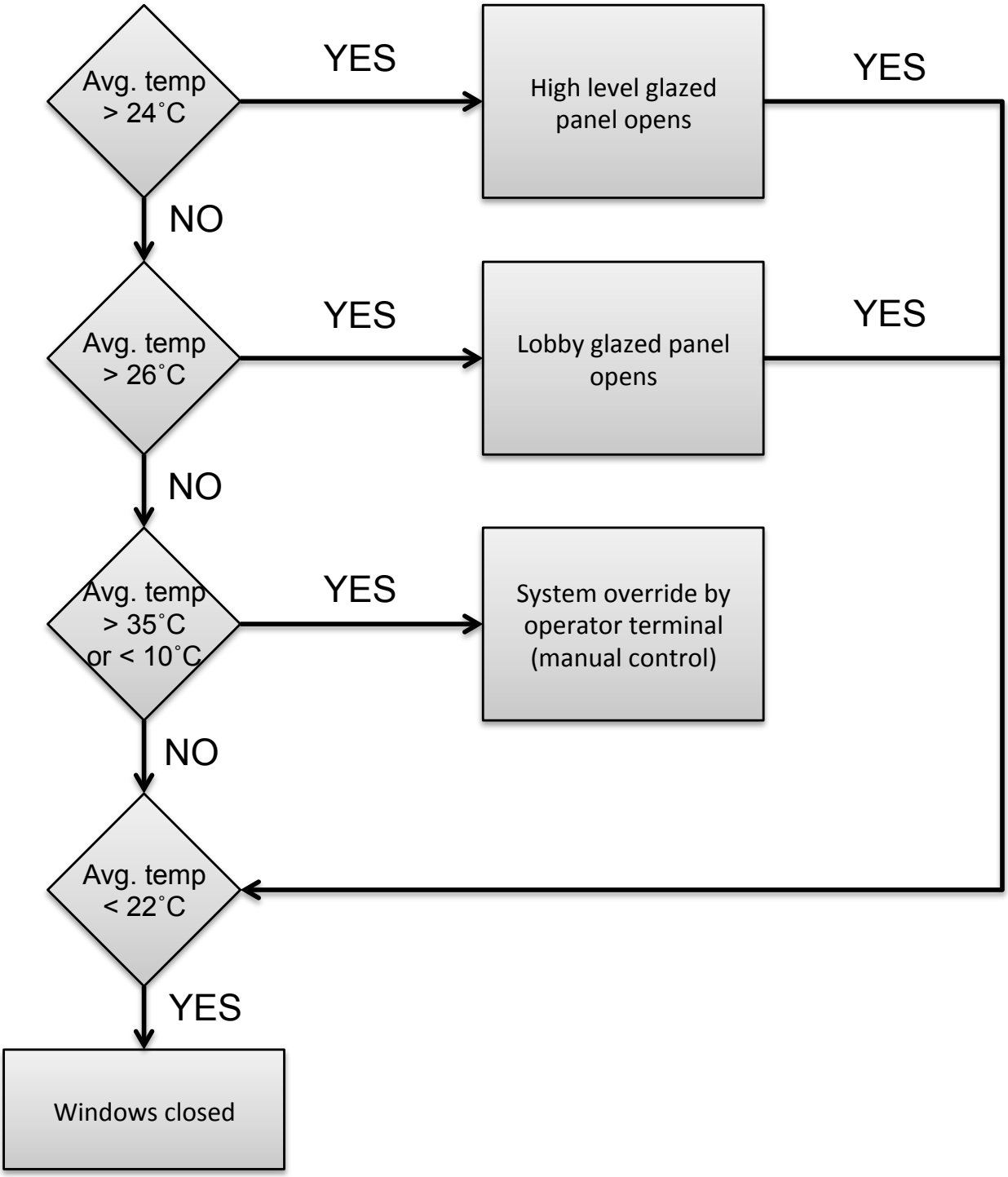
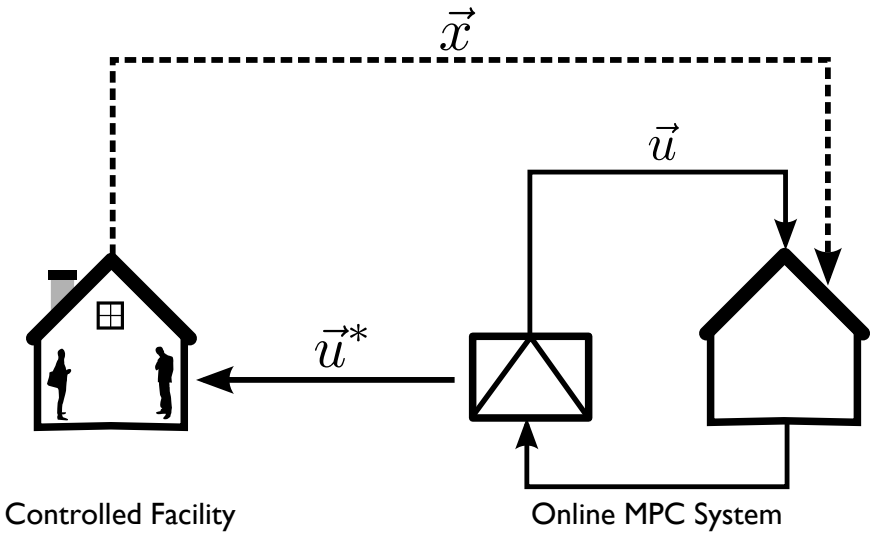


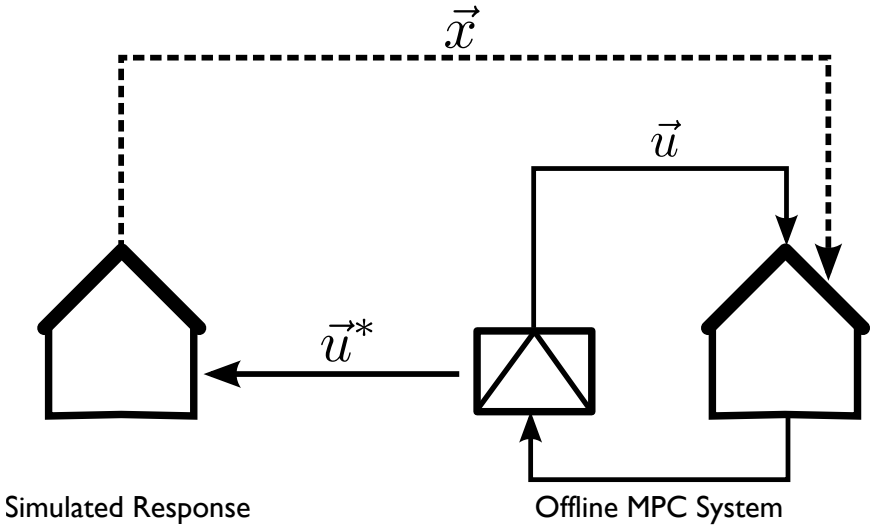
Figure 1.2: A MM control algorithm for the Scottish Parliamentary Building, adapted from [16].

or cost function, based on the predictions of a building-level or system-level model. In the context of building systems, MPC allows for the development of optimal operation strategies that minimize the energy use, carbon dioxide emissions, or operating cost of a facility. Although the potential of MPC has become known in the HVAC engineering field in the past decade [46, 49], it has only recently been applied to MM buildings by Spindler and Norford through the optimization of inverse models specifically trained on two unique buildings [87, 88, 89].

The vast majority of MPC studies presented in the literature are so-called **offline** MPC investigations in which a receding-horizon MPC problem is solved and implemented on the same building energy model, often in a deterministic fashion that ignores model mismatch and uncertainties in weather and occupant behavior. This is in contrast to **realtime** or **online** MPC which optimizes strategies on model predictions, then implements the solution on a physical plant. In online MPC, model mismatch and stochastic influences are present, and the impacts of those phenomena are fed back into the system. The two processes are illustrated and described in Figure 1.3.



(a) Online/Real-Time MPC



(b) Offline MPC

Figure 1.3: Online MPC evaluates potential control vectors ($\vec{u}$) on a building model and implements an optimal control ($\vec{u}^*$) on a controlled facility. The resulting states in the building ($\vec{x}$) may differ from predictions and can be fed back into the model. In offline MPC, the optimal control vector is instead applied to a model. This model can be identical to the one employed in the offline MPC loop (implying perfect predictions, as is the case in this research), or may contain randomly introduced errors (as might be the case when evaluating impacts of uncertainty on the solution).

1.2 Challenges in Extending Model Predictive Control to MM

For a variety of reasons, online MPC applications for supervisory control in buildings effectively do not exist outside of a few custom test cases in industry and academia. For one, the model predictive controller must be able to communicate with a building automation system (BAS), and this kind of integration work is challenging, even with open protocols like BACNet. Another obvious hurdle facing online MPC applications is psychological in nature. Building operators and facility managers may not be willing to cede control of their building to an off-site server running energy simulations and optimization algorithms of considerable complexity. Facility managers may also perceive security risks by opening up their BAS to external network traffic. Network security concerns might be alleviated by conducting MPC using an on-site, dedicated computer, but this still does not eliminate the need to integrate the system with the BAS.

Finally, there is the issue of computing time. MPC can be a computationally burdensome process, especially if performed on energy models of any sort of complexity (e.g. non-linear, multi-zone, multi-physics models). This problem is usually overcome by developing reduced-order models that can be solved with linear or quadratic programming techniques. Although a number of examples of MPC using simplified building or plant models exist in the literature [20, 88], such approaches may not be appropriate for every MPC application and detailed models may be preferred. For example, if thermal comfort is constrained or penalized in the optimization, radiant heat balances must be calculated, a common task in detailed simulation engines. More importantly, as practitioners attempt to bring MPC to scale in commercial facilities, it may be desirable to use existing, validated simulation engines rather than purpose-built reduced-order models. This dissertation assumes that it is ultimately desirable to use validated thermal simulation engines for MPC, and as such, MPC is applied to physical/white box models of typical MM buildings. This approach allows the use of freely available and validated building energy simulation tools, like EnergyPlus [38]. Secondly, the research examines the form of MPC solutions in a variety of climates. But most importantly, the ultimate intent of this research is not to demonstrate the performance of online MPC applied to a

live MM building. Rather, this dissertation examines whether similar performance can be achieved using a much simpler approach.

1.3 An Alternative Approach: Rule Extraction

With improvements in computational power, increased affordability and access to cloud computing resources, and better communications standardization and security for BAS equipment, online MPC should start to see increased penetration for sophisticated commercial facilities in the coming years. Until this becomes a reality, other approaches are required for achieving near-optimal control at scale. This dissertation asks the question, can commercial buildings—MM buildings in particular—reap some of the benefits of MPC while foregoing the complexities, cost, or perceived risks of an online MPC implementation?

This research examines whether this objective could be achieved with **rule extraction**, a process that utilizes supervised learning methods to approximate the performance of MPC by learning from offline solution patterns. This approach first saw application in water resource management where simplified rules for reservoir management were developed to approximate optimal management policies for a reservoir network [12, 100]. The only prior related work in the HVAC field was recently completed by Coffey, who explored the use of parametric MPC studies to develop lookup tables of near-optimal control policies [29]. These lookup tables can then be used to quickly approximate optimizer responses in real-time implementations. This dissertation's approach is closer to the previous work in water management in that supervised learning methods are trained on the results of offline MPC solutions.

The rule extraction process, as envisioned and developed in this research, is illustrated in Figure 1.4. In steps 1 through 3, a building energy model is developed and offline MPC is used to develop a training set of optimal solutions and corresponding building states. In conventional offline MPC studies, solution sets would be evaluated "manually" through statistical or graphical techniques. The analyst might develop improved control logic from the solution based on a semi-quantitative "paraphrasing" of the observed solutions; however, in the rule extraction framework,

supervised learning techniques are used to directly generate the control rule in the form of an inferred function (step 4). Once a rule is trained, performance is tested to examine the skill of the rule in reproducing optimal behavior, seen in steps 5 and 6. The specifics of the **open** and **closed loop tests** will be clarified in Chapter 3. Steps 4 through 6 generally involve some iteration, since it is often necessary to introduce expert knowledge as the solution is better understood. Finally in step 7, the final rule is converted into control logic scripted in an applicable BAS language.

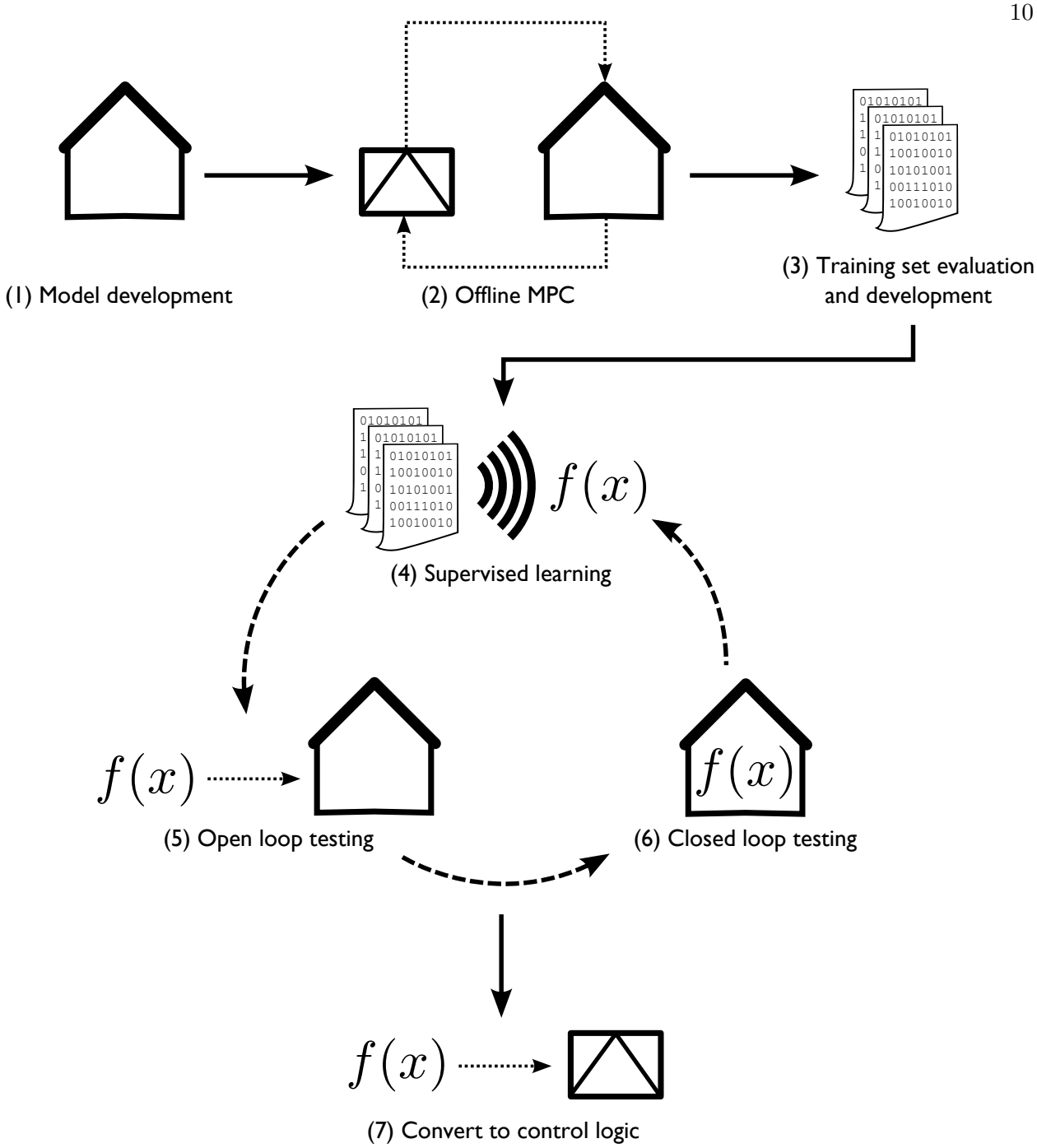


Figure 1.4: Rule extraction proceeds in a seven-step process. In steps 1 through 3, a building energy model is developed and offline MPC is used to develop a training set of optimal solutions and corresponding building states. Supervised learning techniques are used to directly generate the control rule in step 4. Once a rule is trained, performance is tested to examine the skill of the rule in reproducing optimal behavior, seen in steps 5 and 6. Steps 4 through 6 generally involve some iteration, since it is often necessary to introduce expert knowledge as the solution is better understood. Finally in step 7, the final rule is converted into control logic scripted in an applicable BAS language.

1.4 Problem Statement and Objectives

Despite heightened interest in improving operational efficiencies of buildings and the promise demonstrated by a variety of MPC research over the past two decades, MPC has seen effectively no implementation in buildings. In an attempt to bridge the very promising energy and comfort benefits of MPC with the realities of conventional control installations in most of today's commercial buildings, this dissertation aims to apply statistical and data mining techniques to:

- More rigorously analyze the patterns and relationships in offline MPC solutions
- Automatically formulate high-performing decision models/rules from the results
- Enable implementation of near-optimal control rules in a format that could easily be programmed using BAS script languages
- Investigate the ability of finely tuned heuristics to approximate MPC, offering insight into cases where online MPC may provide meaningful performance benefits that cannot be duplicated in any other way

MM buildings, whose performance is highly dependent on well-orchestrated supervisory controls, will be used as a test case for rule extraction. As general guidelines for MM control design currently do not exist, examining MM building control through the lens of MPC is a worthwhile goal unto itself, an exercise that should provide a meaningful contribution to the design community as well as existing MM building operators. Findings from the simulation study presented in Chapter 6 will be presented in greater detail and with an eye toward design recommendations in a separate report. The offline MPC simulation study will achieve several objectives, including:

- The first attempt to benchmark existing MM building control schemes against optimal results generated by MPC
- Investigation of the sub-optimality of predicted mean occupant window opening behavior in MM buildings, according to accepted behavioral models

- Evaluation of different MM control topologies (i.e. changeover vs. concurrent) over a range of building types and climates

1.5 Organization

The dissertation has been organized into the following chapters, as follows:

- **Literature Review and State of the Art:** This chapter examines present knowledge in the areas of mixed mode buildings (both theoretical and applied), model predictive control, and data mining/knowledge discovery techniques to be applied in rule extraction.
- **Methodologies:** Relevant methodology and nomenclature is introduced in this chapter. The basic MPC framework used (including a description of the MPC software environment developed), origins of MM building models, design of a broader simulation study, and rule extraction techniques are discussed.
- **MPC Validation Cases:** This brief chapter covers the validation work that was conducted during the development of the MPC environment.
- **MPC and Rule Extraction for a Binary Window Control Problem:** This chapter presents the proof of concept for the rule extraction technique based on the results of a simplified MM control problem. The results have been published in a series of two papers [69, 70].
- **Offline MPC Simulation Study:** Results of an in-depth offline MPC simulation study are presented in this chapter. The study provides benchmark optimal control results for a range of MM building types under different comfort considerations and climates. Results are compared against non-MM buildings as well as MM buildings using more conventional control heuristics. The results from this section form the basis for the remaining rule extraction cases.

- **Near-Optimal Supervisory Control for Select MPC Cases:** Two MPC cases from the simulation study are used to demonstrate the performance of the rule extraction approach. The rules are applied in different climates and buildings to examine their robustness and sensitivity to training conditions.

- **Field Test and Experimental Validation:** The results of a field experiment are presented in which extracted rules are applied to the control of a test cell with a chilled ceiling and fan-assisted natural ventilation. Energy and comfort performance as well as model mismatch impacts are addressed.

- **Conclusions, Discussion, and Outlook:** The concluding chapter discusses more broadly the effectiveness and practicality of the rule extraction technique and provides recommendations for future research. It also secondarily discusses the broader implications of the offline MPC simulation study for the operation of MM buildings.

Several conventions are used throughout this document. Any terms of art are introduced in boldface text. Numbers are generally presented in SI units. Supplementary materials are provided in several appendices, including source code for certain algorithms, detailed simulation results, and experimental error analysis.

Chapter 2

Literature Review and State of the Art

The following section provides an overview of several research areas integral to this research. First, MM design and control issues are reviewed. A select literature review related to MPC in buildings is then provided, with one prominent application to MM buildings. Next, a survey of papers mostly from other engineering fields provides some context for the proposed rule extraction approach. Finally, a survey of adaptive thermal comfort literature and perspectives on its application to MM buildings is provided, as thermal comfort is a crucial consideration in the proposed MPC problems.

2.1 MM Design and Control Case Studies

The concepts of mixed-mode ventilation and cooling for commercial buildings have been around and in practice for two decades in Europe and somewhat less in the United States. Numerous case studies and summary reports have documented the performance, energy savings potential, and occupant satisfaction with these buildings over that period of time; however, only the largest meta-surveys incorporating multiple buildings have been included.

2.1.1 IEA Annex 35: HybVent

The first major research to deal specifically with the unique control needs of mixed mode buildings was the International Energy Agency's Annex 35 HybVent project, whose contributors published extensively on control strategies for hybrid ventilation, namely Aggerholm [2, 1]. The

Annex 35 project conducted studies on 12 buildings in diverse locations, from central Europe to Japan to Australia.[1] As a result of a lack of design guidelines and a scarcity of natural ventilation-specific building components, several buildings required costly equipment modifications during the commissioning and post-occupancy phases. Similarly, the lack of performance guidelines, measurement methods, and robust control algorithms meant that many of the buildings surveyed did not reach their anticipated maximum energy savings potential. For example, hybrid ventilation in one case study was only utilized for 15% of the year due to poorly designed controls. In short, both performance and cost for the natural/hybrid ventilation implementations were very diverse, ranging from projects that achieved high energy savings at minimal cost (up to 50% reductions in typical electricity consumption for ventilation) to buildings that actually consumed more energy and cost significantly more than a conventional mechanical system due to poor design (a 40% increase in heating energy was reported in one case) [44].

The project provides useful information on the MM strategies employed in the case study buildings. Perhaps the most useful finding: it is currently very difficult to define a "typical" MM system. Most buildings studied used a combination of demand-controlled ventilation (both infrared and CO_2 sensor implementations) coupled with thermostatic controls. There was no prevailing trend toward centralized/supervisory control versus a decentralized approach.In fact, many of the buildings that fared best in the case studies employed very simple or manual controls and did not have any communication between NV components and mechanical systems.

There also did not appear to be strong preferences toward any one hybrid ventilation topology or control strategy. Special use spaces, such as interior conference rooms, might have a dedicated split HVAC system, and thus could be considered "zoned"; however, a mixture of concurrent and changeover strategies were used with similar effectiveness throughout the case studies. No one strategy dominated in terms of effectiveness or energy savings. One key finding from three of the Japanese case studies was that, for changeover type buildings, care must be taken in development

[1] Hybrid ventilation is used here because many buildings involved in the project did not contain vapor compression refrigeration equipment, which is a key system in most American MM building designs. Thus, NV was paired with mechanical ventilation systems only, rather than full mechanical air conditioning.

of heuristics for switching from mechanical cooling to NV. In one building with an enthalpy-driven changeover rule, when mechanical cooling had been used throughout the day, the building had a tendency to become "stuck" in mechanical mode because the enthalpy of the outside air would effectively never drop below that of the precisely conditioned indoor air.

With respect to adaptability and user control, it was found that occupants of cellular offices in particular were very capable of controlling their own thermal environment in a way that minimized the intervention of mechanical HVAC systems, within a fairly broad temperature range. Occupants were more likely to adjust window positions and clothing than to simply revert to mechanical ventilation. This finding was borne out in several office buildings in various nations and climates ranging from Australia to Belgium that had either entirely manual control of window opening or at least a manual override.

Conversely, the study also found situations in which manual control was preferable to fully automated façade openings. In landscape offices where multiple occupants inhabit the zone, automatic controls of inlets in the breathing/occupied zone were attempted, but it was impossible to avoid thermal comfort complaints from at least some of the occupants. In the case of the I Guzzini Illuminazione building in Italy, a fully automated system had to be decommissioned and reverted to manual control due to extensive occupant complaints. Thus, even a fully automated and relatively sophisticated control system could not satisfy all of the occupants all of the time.

Manual control also had its downsides. Although users were able to effectively control cellular offices for thermal comfort, occupants proved to be poorer judges of indoor air quality, often allowing CO_2 concentrations to exceed 1,000 ppm. In classroom settings, individuals were able to control for thermal comfort reasonably well, but IAQ often reached unacceptable levels, and thus automated controls with manual override seem a better approach than a fully manual system.

Finally, one area in which centralized, automatic control proved to be absolutely crucial to optimal performance and energy savings was in the application of night cooling. Perhaps the clearest demonstration of this concept was in the failure of occupants of the Belgian PROBE building to effectively control window openings for night cooling. Users would either over- or under-cool the

building due to errors in prediction of nighttime temperatures and misperceptions of the ability of the building's mass to reject heat. [1, 2]

Other meta-surveys conducted by NREL [91] and Liddament et al. [65] have echoed the general recommendations and energy savings potential seen in the Annex 35 case studies. Further review is provided on these particular studies in Appendix A.

2.1.2 UC Berkeley Center for the Built Environment

The Center for the Built Environment (CBE) at UC Berkeley is arguably the leading US research institution for MM building performance. A variety of projects under the guidance of Dr. Gail Brager have investigated the control of MM buildings, occupant satisfaction, and adaptive thermal comfort. In 2007, Brager et al. of CBE at UC Berkeley conducted a survey of existing MM buildings and their control strategies. The objectives of the work were to better understand the decision-making framework that informs MM designs as well as to document control strategies for existing MM buildings. The study had a predominantly US focus but drew on a handful of international buildings as well. Several detailed case studies were examined for which CBE was able to obtain detailed control algorithm and performance information.[2]

Based on the buildings examined, Brager et al. offered general guidelines on the implementation of MM ventilation controls, which they describe as a continuum of fully manual to fully automated solutions. Automatically opened windows are recommended for a variety of purposes in the building, including hard-to-reach areas (e.g. a tall atrium), to enable nighttime cooling, to achieve minimum ventilation requirements, or for spaces in which there is no dominant "owner" that might open and close the window. The report makes a strong recommendation to use manually controlled windows liberally in the occupied zone, affording occupants the greatest degree of control of their environment possible. The authors warn strongly against excessively "engineered" solutions that, although perhaps optimally designed from an energy standpoint, may risk "losing

[2] Buildings included the Aldo Leopold Legacy Center (Baraboo, WI), the San Francisco Federal Building (San Francisco, CA), the William and Flora Hewlett Foundation (Menlo Park, CA), the University of Nottingham (Nottingham, UK), the Waterland School (the Netherlands), the Scottish Parliamentary Building (Edinburgh, Scotland), and the Zoomazium Woodland Park Zoo (Seattle, WA).

the adaptive opportunity." [16]

The report also conducted highly detailed studies of several buildings and documented their control algorithms to examine whether there were any commonalities between the most successful buildings. With regards to algorithm inputs, temperature was the most common, followed by CO_2. Moisture sensors were seen in use in several cases for the purpose of limiting condensation on indoor surfaces (moisture was not considered for comfort purposes). Modifiers or overrides used to prevent NV operation during unpleasant conditions included outdoor temperature, wind speed, and sometimes rain indicators.

Several different flavors of NV control logic prevailed. In some cases, NV was controlled to provide for hygienic ventilation requirements, whereas in other cases NV was controlled to achieve zonal cooling set points (with or without consideration for ventilation air volume requirements). Still other schemes controlled NV solely with structural cooling in mind (i.e. nighttime pre-cooling of exposed concrete slabs). The report concludes that further work must be undertaken to optimize and generalize control algorithms for MM buildings, such that a library of control strategies applicable to specific climates can be developed. [16]

A review of several other case studies that relate to the control of NV and MM buildings is provided in Appendix A.

2.2 Thermal Comfort Standards and the Adaptive Principle

Thermal comfort will play a pivotal role in guiding the MPC process in MM buildings, and therefore a review of applicable thermal comfort literature is provided.

The longstanding thermal comfort standards in the HVAC&R industries, namely ASHRAE 55 and ISO 7730, were established around statistical models of occupant thermal comfort for controlled indoor environments. The widely used PMV-PPD model attempts to estimate the predicted percent of building occupants dissatisfied with thermal conditions (PPD) based on correlations with the predicted mean vote (PMV) of those occupants, based on research conducted by Olesen, Bassing and Fanger (1972) [79]. Votes are predicted on what is now termed the ASHRAE thermal sensation

scale and are influenced by a variety of factors, including the clothing levels of the occupants, activity levels, the operative temperature of the indoor environment, and various other physical parameters [3]. Thermal comfort standards then provide guidance on the acceptable operative temperatures for a range of psychrometric conditions that are expected to satisfy 80% or greater of the occupants.

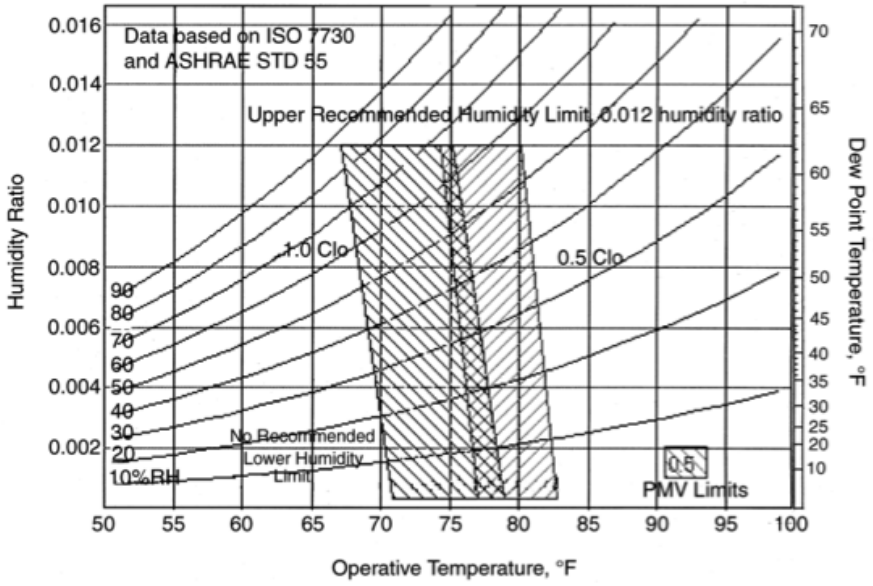


Figure 2.1: Current thermal comfort guidelines as specified by ASHRAE for conventionally conditioned buildings. Source: ASHRAE 55-2004 [3]

However, until standards were updated in 2004, it was extremely difficult for naturally ventilated buildings—in fact, any building lacking extensive use of mechanical systems—to meet these thermal comfort requirements, despite the fact that post-occupancy surveys for these buildings indicated that their occupants were generally satisfied from a thermal comfort standpoint. According to de Dear and Brager (2000), the predictive model of thermal comfort could not explain findings from many passive buildings in which "more person-centered strategies for climate control" had been deployed [18]. Brager, de Dear and others advocated for a modification and relaxation of these standards to include consideration of occupant adaptation to thermal conditions. An extensive review of the various modes of adaptation and the other biological, psychological, and social

processes that research has suggested may influence thermal comfort is beyond the scope of this research.

For buildings in which thermal adaptation occurs, research has found improved correlation between occupant comfort and mean outdoor monthly temperatures compared to the existing PPD-PMV model that only considers local effects like air temperature, air speed, and clothing levels. Brager and de Dear first proposed modifications to thermal comfort standards in 1998 to account for occupant adaptation in naturally ventilated buildings, later defined by ASHRAE 55-2004 as "those spaces where the thermal conditions of the space are regulated primarily by the opening and closing of windows by the occupants" [3]. The early published works and eventual standard provide a linear band of acceptable operative temperatures based on mean outdoor monthly temperature [18, 17, 34]. The currently allowed operative temperature range in the standard is presented in Figure 2.2. What the adaptive model implies is that an occupant's temperature preferences are influenced by seasonal or monthly weather conditions, providing that occupants have control of their thermal environment through natural ventilation elements like operable windows.

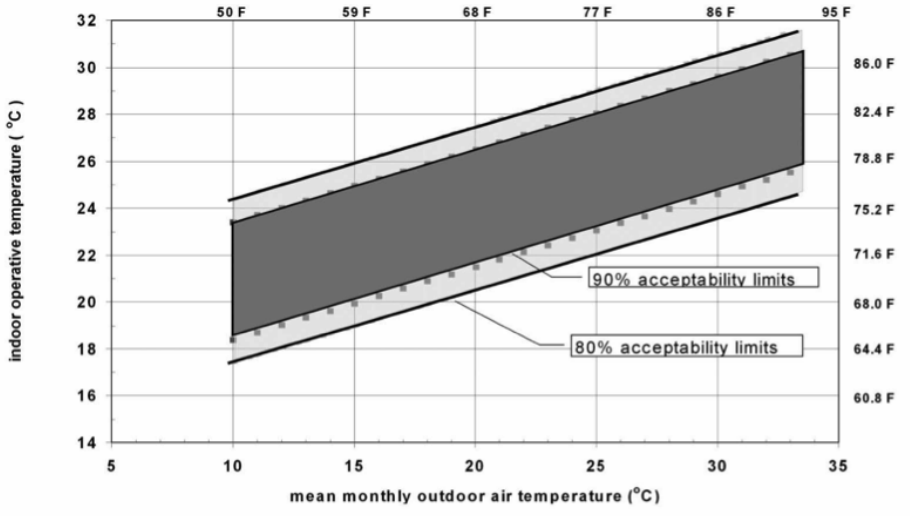


Figure 2.2: Indoor operative temperature acceptable ranges are presented, with 80% and 90% user satisfaction shown. Allowable operative temperature varies with mean monthly outdoor temperature. Source: ASHRAE 55-2004 [3].

A related but not identical interpretation of adaptive thermal comfort has also been incorporated into EN15251:2007, a European standard that, among other things, specifies thermal comfort criteria for buildings. Under the standard, so-called "buildings without mechanical cooling" (i.e. buildings that do not use vapor compression refrigeration equipment such as chillers to cool water and air) may utilize a set of thermal comfort criteria for cooling months to account for occupant adaptation [39]. Specifically, the standard allows the indoor operative temperature to vary linearly with the running mean outdoor temperature. The running mean outdoor temperature is an exponentially weighted average of the mean outdoor temperatures of the last month. A chart of allowable temperature bands for cooling season operative temperatures is provided in Figure 2.3.

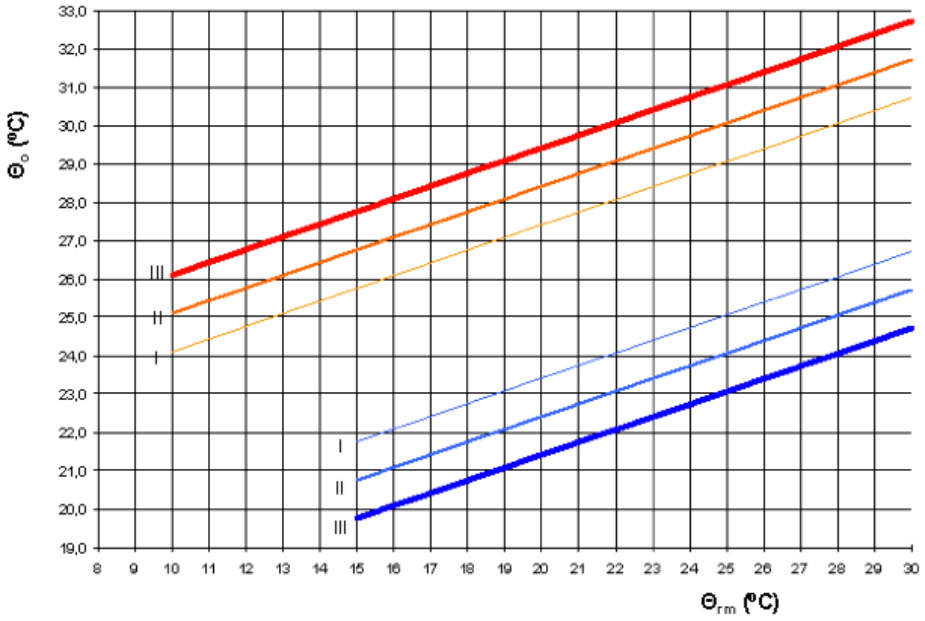


Figure 2.3: Indoor operative temperature limits for cooling season as a function of the running mean outdoor temperature. Source: EN15251:2007 [39]

2.2.1 Applicability of Adaptive Comfort to MM Buildings

Given the emphasis of standards language on free-running buildings, the question arises, does the adaptive principal apply to MM buildings at all? More fundamentally: does thermal

adaptation even apply entirely to all naturally ventilated buildings, as allowed in ASHRAE 55-2004. The answers to these questions are unclear.

Even in naturally ventilated buildings, there will be times of year or regions of the building in which the thermal environment clearly is not controlled by the opening and closing of windows, making adaptation impossible. For example, the core region of a naturally ventilated office building will be influenced very little by the opening and closing of windows. Similarly, in the heating season when windows are primarily kept closed to avoid drafts, the zone thermostat is the primary means of thermal control, not windows. Research into occupant adaptation suggests that occupants of naturally ventilated buildings only truly exhibit thermal adaptation when the building is free-running, that is, when all active heating and cooling systems are off [9, 75, 76]. In this way, not even all naturally ventilated buildings will adhere to adaptive thermal comfort models all of the time and in all regions of the building.

The issue becomes more complicated when one extends adaptive thermal comfort models beyond naturally ventilated buildings to the more general case of "buildings without mechanical cooling," as in EN 15251:2007, or to MM buildings, which very well could employ mechanical cooling in some form. A recent Dutch thermal comfort standard, NPR-CR 1752, may provide a useful framework for answering these questions. NPR-CR 1752 provides thermal comfort guidelines specifying upper and lower bounds for indoor operative temperatures as a function of the outdoor running mean temperature, very similar to EN 15251:2007. In an attempt to generalize the application of adaptive comfort models, the standard specifies two different types of buildings: alpha—those buildings for which adaptive comfort models apply—and beta—those for which "static" models apply. Determination of the building type is made through a decision tree, shown in Figure 2.4, accounting for several types of adaptive behaviors that could be present in the building. Even buildings with mechanical cooling are not excluded from an adaptive classification as long as they provide sufficient adaptive opportunities for occupants. [74, 96]

Any classification scheme for adaptive comfort that is premised on the underlying systems of the building will most likely be made obsolete by the continual diversification in building systems,

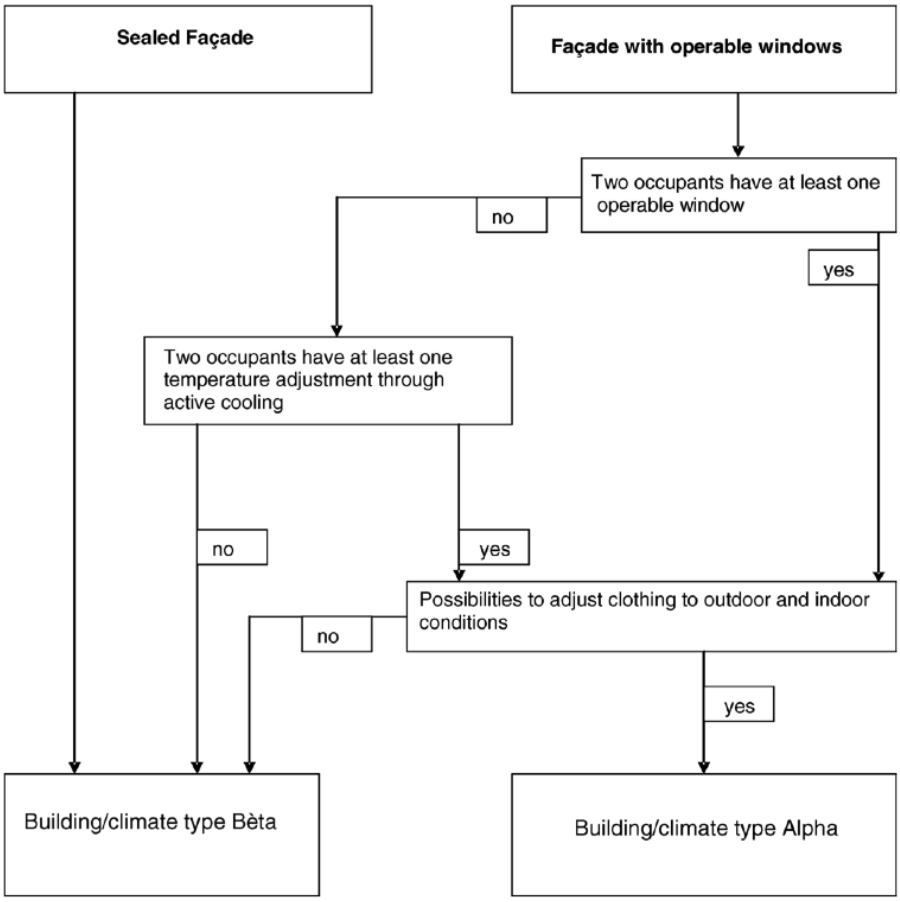


Figure 2.4: Flowchart for classification of alpha and beta buildings. Traditional sealed facade buildings clearly fall into the beta category, whereas various combinations of measures allow buildings to be considered as alpha (following adaptive comfort theory). Source: NPR-CR 1752 [74]

according to Rijal, Humphreys and Nicol (2009) [86]. Their recent work posits that the application of adaptive comfort models to buildings be guided more by the expectations of the occupant in a given space than by the actual building systems in place that enable adaptive behavior.

2.2.2 Applications of Adaptive Comfort Models to MM Buildings

A variety of European research has applied adaptive thermal comfort models to MM buildings. Voss et al. (2007) used EN 15251 to examine the thermal comfort of 22 buildings, each using different passive cooling techniques, including from vertical heat pipes, air-to-earth heat exchangers, and slab cooling. Natural ventilation and night flush cooling were common features of these cooling strategies [98]. Henze et al. (2007) used EN 15251 and NPR-CR 1752 model to examine the impact of adaptive thermal comfort criteria on the optimal control of building thermal mass [51]. Pfafferott et al. (2007) applied the EN 15251:2007 standard in a similar way to evaluate the performance of 12 office buildings from a thermal comfort standpoint, showing that buildings in Germany which utilize natural heat sinks for cooling purposes can provide adequate thermal comfort under most conditions [80].

Beyond this precedent for applying adaptive comfort to MM buildings, some recent papers support the hypothesis that adaptive comfort can be applied to some MM buildings. A series of papers by Rijal et al. have demonstrated that adaptive behavior does in fact occur in MM buildings much in the same way as in totally free-running, NV buildings [84, 86]. Through case studies of MM buildings in the UK, the authors observed occupant adaptation (both clothing change and window operation) and comfort expectations that closely matched those for NV buildings.

Even though none of the thermal comfort standards now in place explicitly allow MM buildings to follow adaptive comfort requirements, the current research will apply adaptive comfort models and associated occupant behavior models to the their evaluation, following the precedent established in previous research.

2.3 Model-Predictive Control

Optimization techniques have been extensively applied to building design problems in recent years, perhaps most notably through the National Renewable Energy Laboratory's BEOpt research program [25]. In terms of advancing control in buildings, a smaller subset of the literature examines optimal, model predictive control for building systems.

MPC has traditionally been applied in buildings to thermal storage problems, beginning with active thermal storage [20, 53, 54, 55, 52]. In many cases, active storage systems could be modeled in an isolated fashion (i.e. decoupled from a complete building thermal simulation), enabling optimization problem formulations that lend themselves well to traditional techniques like linear/quadratic programming, branch-and-bound, or dynamic programming. In such reduced order formulations, the state of a tank storage system could be captured in a single variable (e.g. a state of charge for ice storage), allowing the application of terminal constraints. For example, one could specify that, regardless of the control policy explored, an ice storage tank must be fully charged at the end of a 24-hour period to be prepared for operation on the following day. These additional simplifications help to reduce the decision space and corresponding time to converge for the MPC problem and are even tractable in online MPC implementations.

With regards to passive thermal storage problems, Braun proposed the application of MPC techniques to the control of building thermal mass for demand limiting purposes in 1990 [20]. Braun initially proposed an application of the technique that utilized inverse models for cost evaluation. The idea continued to gain traction in the 2000s through a series of publications by Henze and colleagues Krarti, Florita, Felsmann, and Brandemuehl. Many of the most recent investigations have employed complete building energy simulation tools (such as TRNSYS and EnergyPlus) as cost function evaluators, rather than simplified models. The dispersed nature of the thermal storage medium and its interactions with other heat transfer phenomena in the building—including internal gains and plant equipment—require more sophisticated energy models.

Henze and Krarti conducted US DOE-funded research into the cost savings potential of uti-

lizing both active and passive thermal storage mechanisms available in buildings today, with an emphasis on load-shifting techniques for variable utility rates. Two different optimization environments were compared and cross-validated, yielding similar results. Studies employed a variety of optimization techniques, including direct search (e.g. Nelder-Mead simplex), gradient search, and dynamic programming. It was found that simplifications to building model geometry and zoning could be made without introducing significant sub-optimality to the solution, all while increasing optimizer speed. Model mismatch in material thermal properties as well as internal heat gains were also shown to have a significant impact on the effectiveness of the MPC scheme. When implemented as an online model predictive controller, it was found that highly simplified weather forecasting was sufficient for the optimizer to find near-optimal policies. [46, 47, 48, 50]

Research by Henze et al. in a 2007 ASHRAE project examined optimal control of passive thermal storage in buildings through a sensitivity analysis to examine potential for cost savings by optimal load shifting. A variety of utility rate structures, building construction, internal gains, and weather characteristics were analyzed as factors in the study. PMV comfort violations were generally avoided by limiting the range of cooling set points to within the summer comfort band of ASHRAE 55. Building mass, internal gains, and diurnal temperature swings were shown, as expected, to be decisive in determining the available pre-cooling capacity. Strong pre-cooling incentives through high on-peak pricing led to deeper pre-cooling, but pre-cooling energy cost penalties were limited to 8% even in the worst cases. [49, 45]

2.3.1 Supervisory Model-Predictive Control of MM Buildings

Spindler and Norford first examined the optimal control of an entire MM building through Spindler's Ph.D. dissertation at MIT (2004), entitled System Identification and Optimal Control for Mixed-Mode Cooling, later publishing the work as a two-part paper (2009) [87, 88, 89]. The control of a MM office building was examined in two parts. First, an inverse modeling system identification framework was developed to assist in assembling an accurate black-box model of the building that provided lower prediction error than those published for many other naturally

ventilated buildings. A Principle Hessian Direction Regression Tree (PHDRT), neural network, and Kernel Recursive Least Squares (KRLS) model were each developed, trained, and compared against measured building data. Despite the inherently non-linear nature of the processes of interest (particularly with regards to airflow), the linear PHDRT model provided the best overall predictive capability and about half the RMS error of a corresponding physical model of the building in question (0.42K, compared to 0.74K for the physical model). [88]

In the second phase, a combination of optimization techniques, including dynamic programming, integer programming, genetic algorithms, and simulated annealing were investigated to optimally control the model with the goals of achieving thermal comfort and minimizing fan operation. Automatic control for a fan and certain operable windows was investigated. Control decisions were optimized over a 24-hour planning horizon, although multi-day horizons of 48 and 72 hours were also investigated. To reduce the dimensionality of the problem and reduce computing time, a number of different decision variables were discretized into lookup tables. For example, the number of different operating modes of the building was fixed and enumerated in an output table, representing a discrete set of combinations of fan operation and operable window opening areas. A binary search was then performed on the resulting simplified model, reducing computing time. The optimal control of night cooling was shown to maintain indoor temperatures over 4 °C below ambient throughout the hottest parts of the day. [89]

Spindler applied the same modeling and optimal control concepts to a second building, but was unable to obtain significant data on window openings to be able to either validate optimal control algorithms or implement a supervisory, MPC strategy for the building [87]. Thus, it remains to be seen whether a similar strategy (linear, data-driven building models optimized with a binary search) could be effectively extended and generalized to a larger number of buildings. Furthermore, the use of black-box regressive building models effectively limits the approach to existing buildings with sufficient monitoring equipment and data for a training set. This may significantly limit the practical application of this technique.

2.3.2 Rule Extraction from Offline MPC

Literature in the building sciences most commonly presents MPC as a benchmarking tool, useful for exploring maximum energy/cost savings potential for certain control problems that involve slow system dynamics, energy storage, or day-ahead planning. Observations on the underlying strategy pursued by the optimizer can then inform improvements to existing heuristics, as in [49]. In passive thermal storage problems, MPC is almost always employed in this "offline" mode.

If online optimization is too cost-prohibitive or computationally burdensome for the vast majority of facilities to implement, then some tools are required to help extract the useful logic and strategies of offline simulation studies with analytical rigor. If these techniques yield relationships that can be directly employed as control logic in a BAS, all the better. Publications in other fields contain several prospects. Authors Bobbin and Yao published in the late 1990s on the application of genetic algorithms to develop optimal if/then control rules. The technique was applied to extremely simple mechanical systems for validation purposes, and has since seen traction in other fields. However, the if/then rules must be formulated at the time of optimization, meaning that the optimizer must search literally any feasible rule structure during optimization for solutions to be found. The approach is tractable when developing a controller for a two-state-variable system as in the original paper, but it is unclear whether application to more complex control problems is feasible. [13, 14]

There have been some more promising developments in the water management field, where data mining techniques have been employed to generate near-optimal operation policies for reservoir networks. The management rules are mined from a set of MPC results using techniques ranging from simple linear regression to induction decision trees. Wei and Bessler have published on these techniques in the past decade, indicating that they have experienced the greatest model skill and fidelity using various forms of decision trees. [100, 12, 99]

Brian Coffey's Ph.D. dissertation at the University of California Berkeley, entitled Using Building Simulation and Optimization to Calculate Loop Tables for Control, is perhaps the closest

attempt to reduce the complexity of online MPC implementation in the HVAC field [29]. Coffey utilized offline MPC results from parametric studies to develop lookup tables, which can be used for online implementations or to inform control sequences in the design process. His research shows that ideal cases for the lookup table approach can be reduced to parametric investigations on six or seven variables, thus comprising a problem (usually a sub-problem) that can be solved offline in a tractable and economical manner.

Chapter 3

Methodologies

An outline of the general methodologies used throughout the research is provided in the following chapter. Some specialized methodologies appear throughout the remainder of this dissertation.

3.1 MPC Problem Formulation and Software Environment

When applying MPC generally to MM buildings, decision vectors include a mixture of continuous and integer/binary variables, resulting in a mixed integer problem. Furthermore, when the objective function is demonstrably nonlinear—as is the case in the chosen approach which utilizes white-box models—we have a mixed-integer nonlinear programming (MINLP) problem, formulated as follows:

$$\begin{aligned} \text{Minimize } C_{tot} &= f(\vec{x}, \vec{y}) = C_e + C_c \\ \text{Subject to} &: \\ & \vec{x} \in \{0, 1, 2, ...N\}^m \\ & \vec{y} \in \mathbb{R}^m \\ & \vec{l} \leq \vec{y} \leq \vec{u} \end{aligned} \tag{3.1}$$

where C_{tot} is the cost or objective function, which is the sum of energy costs (C_e) and comfort

penalties (C_c). The cost is a function of the m-dimensional vectors $\vec{x}$ and $\vec{y}$, each representing control decisions in time over m time blocks. Decisions in $\vec{x}$ are positive integers or binary, and decisions in $\vec{y}$ are real-valued and continuous. They may further be subject to upper and lower box constraints ($\vec{u}$ and $\vec{l}$, respectively), which can vary over time to enforce acceptable decisions (e.g. allowable ranges of cooling temperature set points). We are not theoretically restricted to only two decision vectors; there may be any number or combination of vectors like those described above. However, limitations on computing resources require that the number of decision vectors and the time dimensionality of the problem are kept low.

Because one might want to investigate optimal control policies over periods of weeks or over an entire cooling season, the size of m could become quite large. Consider the case in which one wishes to optimize the operation of a building's operable windows in conjunction with global cooling set points, generating decisions for each hour of the day. If this problem were examined as a traditional optimal control problem over a period of a week, we would have a decision space of size 2^{168} just for a binary variable. Therefore as the literature suggests, it is sensible to employ a receding horizon MPC approach in which decisions are optimized one or two days at a time, and the problem is broken up in time into a series of sub-problems identical to the one formulated in (3.1), but with significantly less time dimensionality [49]. In this instance, the present cost is also a function of past control policies due to thermal storage effects in the building mass, and to estimate the present cost correctly, it is necessary to either explicitly initialize state variables or pre-condition the building for a sufficiently long period using past optimal policies. The chosen receding horizon MPC approach is illustrated in Figure 3.1, with decisions further discretized in time. Previous decisions (black) determine the thermal history of the building, thus impacting cost and decisions under the current planning/execution horizon (gray).

For a variety of reasons, it was decided early in the research that EnergyPlus was necessary to more accurately model mixed mode and other passively cooled buildings (e.g. radiantly cooled). Unfortunately, the building energy analyst currently lacks easily adaptable tools to enable MPC investigations using standard building simulation engines (DOE-2, EnergyPlus, TRNSYS, etc.) as

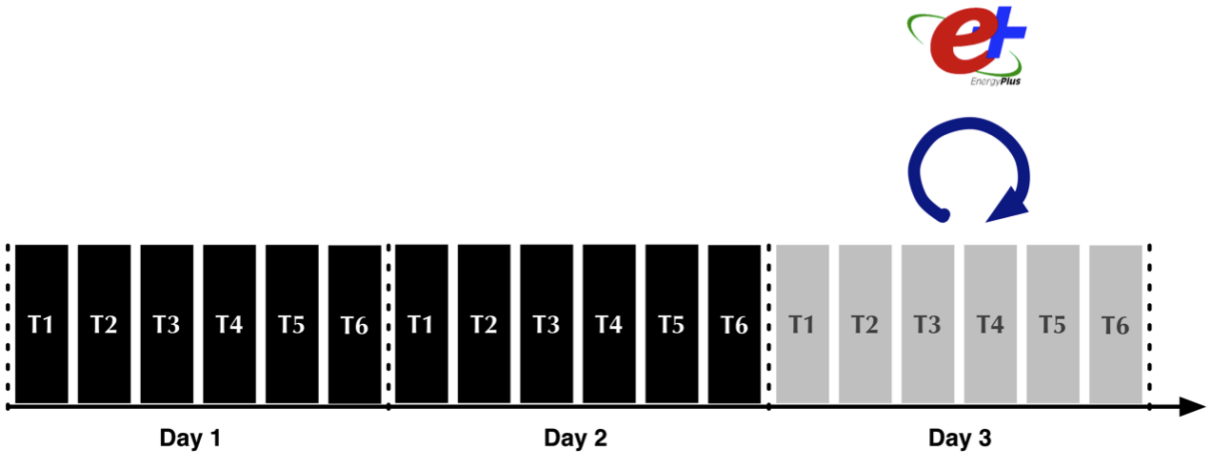


Figure 3.1: Procession of receding horizon MPC approach using six four-hour time blocks to discretize decisions in time.

the objective function evaluator. Although Henze, Felsmann and Florita had previously provided a link between TRNSYS and Matlab [45], no such comparable tool was available at the onset of research to allow MPC investigations in EnergyPlus without some modification to the source code. Wetter has developed a generic optimization interface (GenOpt) that was later adapted to MPC applications through a software framework developed by Coffey [102, 28], but these tools were not available at the outset of the research.

A customized optimization interface should be developed with the following general specifications in mind:

- Enable MPC using "off-the-shelf" software rather than custom-compiled versions.
- Provide the ability to manipulate decision variables of multiple numerical (continuous, integer, binary) and schedule types (setpoints, availability, etc.).
- Preserve the thermal state from previously implemented policies.
- Treat the objective function as a "black box," eliminating the need for the optimizer to gain access to objective function differential properties, such as gradient, Hessian, etc.

- Enable comparisons with multiple optimization algorithms.

Through extensive collaboration with CU Boulder graduate student Charles Corbin, an optimization environment bridging EnergyPlus and Matlab (ME+) has been developed that is capable of addressing the general MINLP problem described above. EnergyPlus is used as a black box cost function evaluator, and Matlab is used for optimization and file I/O. In Figure 3.2, a block diagram schematic of the overall optimization environment is presented to demonstrate the general solution approach. Building models are read in and modified by the chosen optimization algorithm in Matlab by manipulating EnergyPlus schedules. Both continuous and integer/binary decisions are possible, so decision vectors can include values ranging from global cooling set points to pump availability. Matlab writes out the manipulated .idf files, and models are passed to EnergyPlus for simulation. Results are read back into Matlab from EnergyPlus .csv outputs, and the cost function is computed based on a user-customizable objective function calculator. Any output variable from the simulation could be used in the objective function evaluation (energy, demand, and comfort are most common). Detailed descriptions of the environment and several case studies are provided in [30].

This process is used to optimize decisions over a specified **planning horizon**, P, and decisions are then implemented for an **execution horizon**, E, which in most cases is equal in length to the planning horizon. Cost is evaluated over a **cost horizon**, C, which may be longer than the planning horizon to account for multi-day impacts of decisions in more massive structures. It is used in favor of extending the planning horizon to keep the dimensionality of the problem small. In cases where $C > P$, decisions in the planning horizon are replicated in the cost horizon. Finally, because EnergyPlus does not allow the building to be initialized at a precise state, an **initialization horizon**, I, is required to ensure that the terminal thermal state from previous execution horizons is preserved as the optimization advances through time. This process is illustrated in Figure 3.3.

Even though the time step of the building energy simulation is sub-hourly, the planning horizon is segmented into multi-hour blocks of time during which the optimizer is allowed one

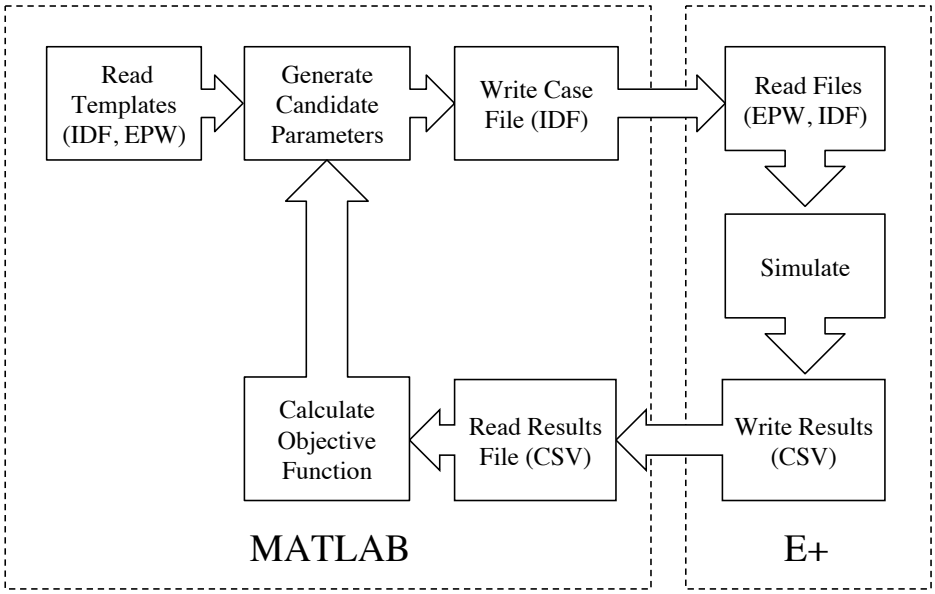


Figure 3.2: The ME+ environment, coupling Matlab and EnergyPlus.

decision on a given variable. This temporal aggregation of decisions significantly reduces the size of the decision space and the computational expense of the optimization. Thus the result for each control parameter under optimization is a decision vector $\vec{x}^*$; it is $n * m$ hours long, where n is the length of each mode and m is the number of modes per planning horizon. In cases where $E < P$, only a subset of these decisions will actually be implemented based on the number of modes in E.

Since elements of the optimization environment are currently being used by a commercial enterprise for conducting online MPC of large commercial buildings, detailed Matlab source code related to Matlab-EnergyPlus coupling has not been provided as part of this dissertation. However, several examples of optimizer code have been made available through Appendix C.

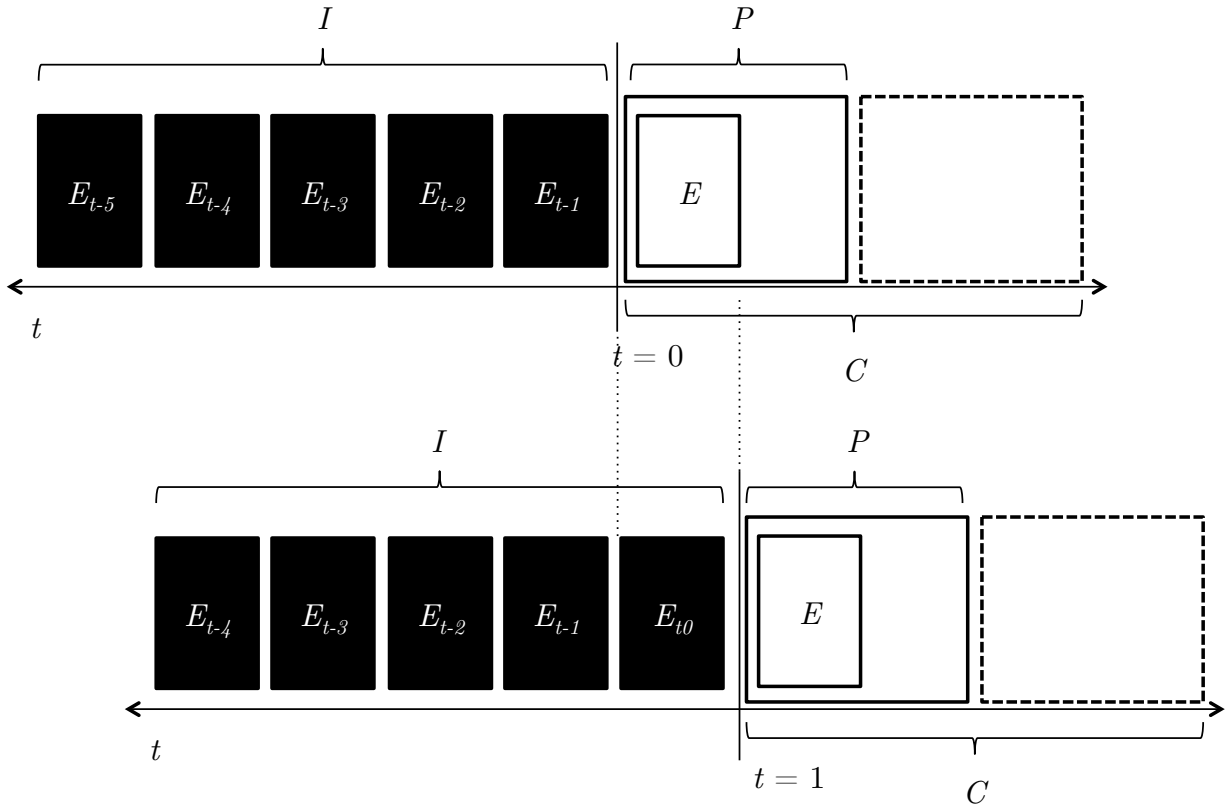


Figure 3.3: The MPC optimization marches through time one execution horizon at a time, optimizing decisions over the planning horizon, examining cost impacts over the cost horizon, and preserving state using the initialization horizon. Decisions replicated in the cost horizon are denoted with a dashed line.

3.2 Particle Swarm Optimization

3.2.1 Canonical PSO Algorithm

Previous research has demonstrated the necessity of appropriately selecting optimization algorithms and choosing appropriate optimizer seed values to avoid premature convergence on local minima that are inherent in passive thermal storage problems [45, 49]. Early test cases conducted on EnergyPlus models underscored the necessity of striking a balance between robust global search accuracy and computational time. Pattern searches like Nelder-Mead simplex are known to become attracted by local minima, thus avoiding more global exploration. On the other hand, many global

optimizers like the ones provided with technical computing packages can search extensively—tens of thousands of iterations have been observed in the course of this research—without converging on a global optimum.

As a result, the meta-heuristic particle swarm optimization (PSO) was adopted. PSO combines simple rules with randomized weighting factors to generate complex search behavior in a population of "particles" evaluating the search space. The action is akin to the flocking behavior of birds and schooling behavior of fish. As with these organisms, information shared between individuals in the swarm affects the decisions of others, all of whom eventually converge on the best solution found by the group. The algorithm is non-deterministic and therefore the search pattern of any swarm is impossible to determine a priori. This characteristic of the algorithm decreases the likelihood that it will become stuck in local minima, at the expense of guaranteed convergence upon the true global minimum. Early optimizations were conducted using a variant of the algorithm presented in the foundational work conducted by Kennedy and Eberhardt [60]. For clarity we will call this "canonical" PSO or PSO-CA. At each new time step, $t+1$, particle velocities are computed from previous velocities/momenta, and then the position of each particle is updated, as given by

$$\begin{aligned} \vec{v}_i(t+1) &= \omega\vec{v}_i(t) + \alpha_1\vec{\gamma}_1\left[\vec{p}_i(t) - \vec{x}_i(t)\right] + \alpha_2\vec{\gamma}_2\left[\vec{p}_g(t) - \vec{x}_i(t)\right] \\ \vec{x}_i(t+1) &= \vec{x}_i(t) + \vec{v}_i(t+1), \end{aligned} \tag{3.2}$$

where:

$\vec{v}_i$ is the velocity vector of the i^{th} particle,

$\vec{x}_i$ is the position vector of the i^{th} particle,

ω is an inertial weighting term,

α_1 is a coefficient weighting the strength of personal best cost,

$\vec{\gamma}_1$ and $\vec{\gamma}_2$ are vectors of uniformly distributed random numbers,

$\vec{p}_i$ is the personal best position vector of the i^{th} particle,

and $\vec{p}_g$ is the global best position vector for the entire swarm.

The position vector is identical to the decision vectors, $\vec{x}$ or $\vec{y}$, presented in (3.1). The algorithm proceeds until a convergence tolerance is reached or the optimizer times out. In this implementation, the coefficient of variance root mean square error (CV-RMSE) of all current objective function values in the swarm compared to the global best value is employed as the measure of convergence

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{\sum_{i=1}^{n}(C_i - C_{\text{best}})^2}{n}}, \\ \text{CV-RMSE} &= \frac{\text{RMSE}}{\text{var}(\vec{C})}. \end{aligned} \tag{3.3}$$

where n is the number of particles in the swarm, C_i is the objective function value of the i^{th} particle, C_{best} is the global best objective function value, and $\text{var}(\vec{C})$ is the variance of all objective function values in the swarm.

More recent investigations by Clerc and Kennedy (2002) demonstrate rigorously that a "constriction factor," χ can be applied to the traditional PSO to ensure stability and prevent "explosion" or divergence of the swarm [26]. We will refer to this formulation as PSO-C. The inertial weighting term disappears, and the governing equations then become

$$\begin{aligned} \vec{v}_i(t+1) &= \chi\left[\vec{v}_i(t) + \alpha_1\vec{\gamma}_1\left[\vec{p}_i(t) - \vec{x}_i(t)\right] + \alpha_2\vec{\gamma}_2\left[\vec{p}_g(t) - \vec{x}_i(t)\right]\right] \\ \vec{x}_i(t+1) &= \vec{x}_i(t) + \vec{v}_i(t+1). \end{aligned} \tag{3.4}$$

The authors demonstrated that the critical value of χ for stability is

$$\chi = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}, \tag{3.5}$$

where ϕ is simply the sum of α_1 and α_2. Common values are $\alpha_1 = \alpha_2 = 2.05$, yielding a χ of 0.729. This formulation of the algorithm and the suggested constriction factors are employed at the heart of the PSO algorithm currently in use.

3.2.2 Constraints and Binary Variable Treatment

In its purest form, PSO is unconstrained, but several techniques do exist to apply constraints. A common technique is to set particle positions equal to boundary constraints once particle values fall outside of the feasible region; however, this technique is known to lead to premature algorithm convergence [101]. This behavior was in fact observed in early test cases. Thus, a penalty-based approach has instead been implemented to coerce particles back into the feasible decision space. The penalty increases with the Euclidian distance to the nearest feasible solution, per techniques employed in [27].

Traditional PSO is meant to operate on continuous variables. The decision space can be discretized by rounding particle positions to the nearest integer, for example. For binary decision variables such as binary open-close signals for windows, this technique does not perform well, especially in conjunction with the aforementioned boundary constraint enforcement techniques. An alternate technique for binary variables was adapted from a 1997 paper by Kennedy and Eberhart, whose technique ensures that $\vec{x} \in \{0, 1\}$ [61]. Velocities are treated as probabilities of state change. They are updated as usual, but are transformed logistically by

$$S = \log \frac{v}{1 - v}. \tag{3.6}$$

The value of S is then compared to a randomly chosen threshold value. If S exceeds the threshold, the value of $\vec{x}$ on that dimension is 1, otherwise 0.

The algorithm also supports a simple neighborhood topology in which the population of particles is divided into groups/neighborhoods whose members can communicate amongst themselves, but not with members of other neighborhoods. This topology allows for groups of particles to "scout" different regions of the decision space, but also proved valuable in parallelizing the algorithm, as discussed below.

3.2.3 Speed Concerns and Parallelization

Many of the modifications on the standard PSO algorithm described above address the global search performance, but overall speed can still be an issue, especially when using building simulation software as an objective function evaluator. Even relatively simplistic EnergyPlus building models can take upwards of 30 seconds to run when including initialization horizon. Add to this the overhead associated with reading/writing model files and results, and individual objective function evaluations can consume more than 40 seconds. With optimizations taking on the order of 500 to 1,000 iterations to converge, this results in wall times of at least 5 hours to optimize a single 24-hour planning horizon using very simple models; one-day optimizations can easily consume nearly a day of wall time with more complicated models. This is clearly too slow for online MPC applications where decisions may be required on an hourly basis, but it proves to be frustratingly slow for offline optimizations as well because of the significant feedback time between launching cases and receiving results.

To this end, a parallel PSO algorithm was developed following the "synchronous" parallelization topology of Koh et al. (2006) [62]. Under these parallelization scheme, each neighborhood can be considered an instance of the PSO algorithm running on a separate processor. Periodically, neighborhoods broadcast information to each other to determine if convergence criteria have been reached and whether new global optima have been detected (e.g. at the end of a specified number of iterations or a period of wall time). The function is implemented in two parts: a master/controller that supervises the various neighborhoods and a slave/client that executes the bulk of the instructions on separate cores. Code for both can be found in Appendix C.

The algorithm utilizes high-level Matlab libraries that implement Message Passing Infrastructure (MPI) commands, which are the standard for parallel communication in high-performance computing. It can be run on a local multi-core workstation or in cluster computing environments. As with most parallel algorithms, parallel PSO does not result in n-fold reductions in computing time for n-tuple processor configurations; however, significant reductions in wall time (greater than 80%) have been achieved, all with more robust exploration of the decision space.

3.3 Design of an Offline MPC Simulation Study

3.3.1 Typical Mixed Mode Building Model Development

A simulation study was developed to apply MPC to "typical" mixed mode buildings for the dual purposes of benchmarking existing control strategies against MPC and to develop training sets upon which to train rules. Of course the first challenge in such a study is to define a typical MM buildings. A variety of sources were examined to inform these decisions, including the Center for the Built Environment's online mixed mode buildings database [22] and data from the IEA's Annex 35 HybVent project [58]. Initial attempts were made to better classify typical MM systems, resulting in additional research and proposals on a MM building taxonomic classification, described in Appendix B.

One primary factor simply involves building size. An examination of data available on MM building floorspace indicates a bimodal distribution, with most facilities clustering in the less than 100,000 sf range and a cluster of larger facilities (Figure 3.4). Given the computational expense of optimizing even simplified MM building models and the higher occurrence of small- to medium-sized MM buildings in the known stock, it was decided to limit examinations to smaller office-type facilities.

A matrix of features corresponding to various configurations of mixed mode buildings observed in the literature was developed (Table 3.1). This matrix is by no means exhaustive, but captures some of the most common system configurations observed in the literature review. The

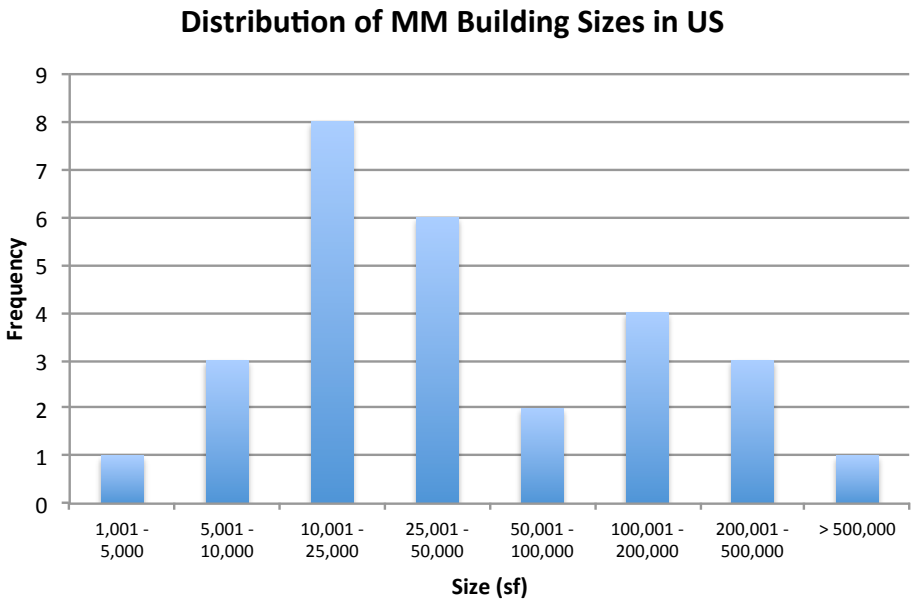


Figure 3.4: Histogram of mixed mode building sizes in the US. Source: CBE mixed mode database [22].

lefthand side of the matrix begins with a base case, a building reflecting conventional HVAC systems (an all-air system with rooftop units and VAV distribution). As one moves to the right on the matrix, mixed mode and other passive cooling features are added.

The base case is an approximately 18,000 sf (1,750 m^2), three-story office building and forms the foundation of all models in the matrix. The basic model—including surface geometries, materials, and systems—was adapted from the US DOE reference commercial building models [36]. It employs an all-air HVAC system with rooftop AHUs, DX cooling, and VAV terminal devices (hot water reheat). To make NV feasible in models MM1-4, the floor plan was narrowed slightly per general design rules of thumb presented in the trade press [24, 23, 72]. The building contains a total of 11 occupied thermal zones. The first floor employs standard core-perimeter zoning, whereas the second and third floors have a large open office and two perimeter office zones on the east and west orientations. An isometric view of the building is presented in Figure 3.5.

Building models contain both manually operable and automated windows. In the case of

Table 3.1: Matrix of features and models included in simulation study.

	BASE	MM1	MM2	MM3	MM4
Mechanical Ventilation					
Fully Mixed Air-Side Sizing	✖	✖	✖		
Occupied Zone Air-Side Sizing				✖	✖
Dedicated Outdoor Air System					✖
Natural Ventilation					
Sealed Façade	✖				
Manual Operable Windows		✖	✖	✖	✖
Automated Operable Windows				✖	✖
Secondary Systems					
Single-Duct VAV	✖	✖	✖		
Under-Floor Air Distribution				✖	✖
Radiant Slab / TABS					✖
Primary Systems (Cooling Only)					
Rooftop DX	✖	✖	✖	✖	
Ground-Source HP					✖
Envelope					
Standard Construction	✖	✖			
Heavy Massing and Low U-val			✖	✖	✖
Passive Shading Devices			✖	✖	✖
Active Shading System					✖
Advanced MM Control Features					
None	✖	✖	✖		
Window-HVAC Interlock				✖	✖

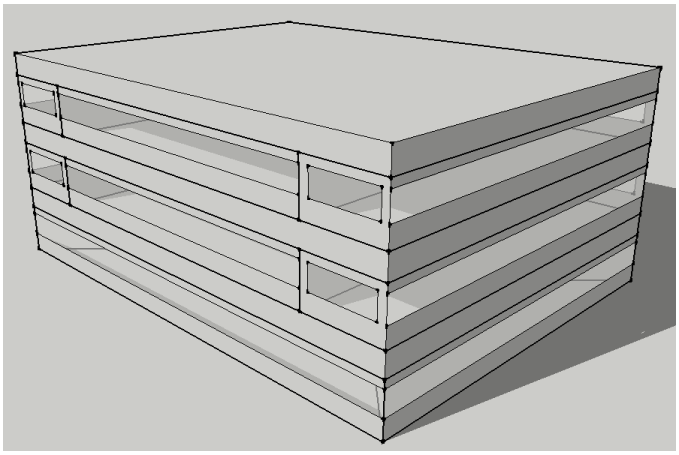
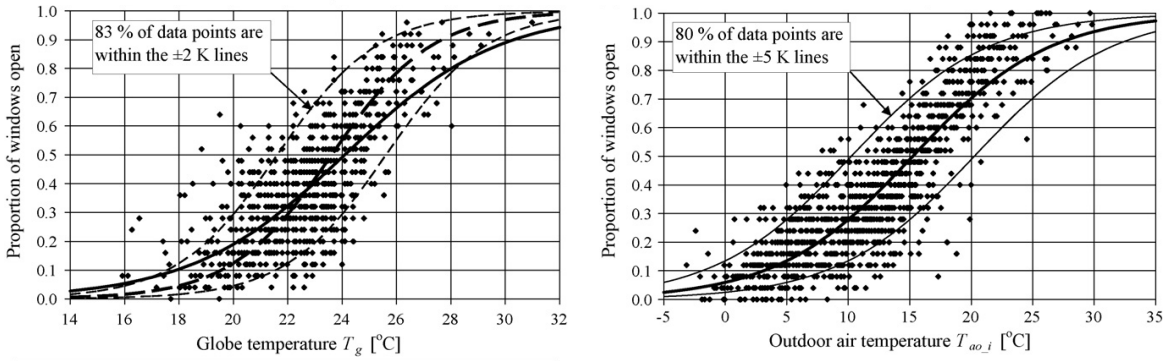


Figure 3.5: Isometric view of small MM office building model.

manually operable windows, assumptions needed to be made regarding human behavior. Occupant window opening behavior was deterministically simulated through an implementation of the "Humphreys algorithm" enforced through an EnergyPlus `EnergyManagementSystem` program. The algorithm was developed based on field studies of occupant behavior in free-running buildings by Rijal et al. [85, 83], but has also been shown to adequately describe the behavior of occupants in some MM buildings as well [84]. Whereas the traditional algorithm operates in a stochastic fashion, this deterministic implementation is meant to track mean window opening behavior. We compare the probability of window open value generated by the algorithm, p_w, and compare it to a threshold of 0.5, opening windows to their full effective opening area when p_w exceeds the threshold and vice versa. This probability is governed by the following logistic relationship:

$$\text{logit}(p_w) = 0.171T_{op} + 0.166T_{db} - 6.4. \tag{3.7}$$

T_{op} is the zone operative temperature and T_{db} is the ambient dry bulb temperature. It should be noted that this implementation ignores the hysteresis deadband imposed by the original algorithm, so openings can change more frequently (limited to the hourly timestep on which the algorithm is executed). Openings still track the basic logistic relationship depicted in the original work (Figure 3.6).



(a) Operative/globe temperature relationship

(b) Outdoor temperature relationship

Figure 3.6: The surveys of Rijal et al. identified relationships between the proportion of windows open in a free running building vs. globe/operative and outdoor temperatures. Source: Rijal et al., 2007 [85].

Airflow through the building is computed through EnergyPlus' nodal airflow model, including flows resulting from window openings, inter-zonal leakage, and infiltration. Wind pressure coefficients for the building have been estimated using an online database of wind pressure coefficient relationships resulting from parametric studies of typical building geometries [93].

Ultimately, model simplifications were required to reduce simulation times for offline MPC runs. Each whole-building model was simplified down to its second story, using adiabatic boundary conditions on the ceiling and floor. This follows the findings of Henze et al. who found that zone simplifications in energy models still yielded meaningful MPC solutions [46, 47, 48, 50]. Individual zone temperature responses were still basically congruent with the original whole-building response, with zone temperature RMSE values of less than 1K. Simulation time was reduced by up to 80% in some cases.

3.3.2 Optimization Parameters

Naturally each combination of systems in the matrix provides countless combinations of decision variables that could be optimized, even at the supervisory level. For example, the simple decision of window openings can be optimized at the building, façade/orientation, zone, or indi-

vidual levels. Similarly, window controls could be considered binary or continuous, depending on the level of granularity of the investigation. Decisions can be made for each hour of the day or in aggregated blocks of multiple hours. Without placing some boundaries on these choices, the scope of the simulation study very quickly becomes unmanageable and computationally intractable due to the large decision spaces. To this end, Table 3.2 provides a list of decision variables that were considered for optimization, depending on the mix of systems in the building. Decisions were executed at the building level to avoid defining the MPC problem in such a building-specific way that the results could not be generalized.

Table 3.2: Example Decision Variables

Variable Name	Type	Relevant Systems
Window position	binary	Manual and automated windows
Cooling setpoint	continuous	Air-side cooling systems
CW pump PWM duty cycle	continuous	Radiant cooling (TABS)

3.3.3 Climatic Parameters

Offline optimizations were conducted for each MM building in the matrix, using comparisons against base and reference case models to compare comfort and energy savings. **Base case** models represent baseline control strategies and no natural ventilation, whereas **reference case** models take advantage of MM cooling opportunities using more conventional control heuristics. A subset of optimizations were conducted across a range of climates to investigate whether the optimal patterns of operation remained consistent. One anticipates that strategies will vary by climate region as a result of differences in diurnal temperature fluctuations—expected to drive night flush ventilation strategies—and moisture levels, limiting the hours during which outdoor air is suitable for natural ventilation. It should be emphasized that this research is not intended to be an exhaustive climate feasibility study for natural ventilation and mixed mode cooling. Rather, climate is included as a parameter to investigate the sensitivity of supervisory control strategies to this factor. The research

considered a range of hot/cold and dry/moist climates, shown in Table 3.3. Heating-dominated climates were excluded from the study (i.e. above climate region 5). Las Vegas was included as an extreme case, with higher cooling degree-days than any of the other climates.

Table 3.3: Locations for Climate Sensitivity Analysis

Location	ASHRAE Climate Zone	Description
Boulder, CO	5B	Cool, dry
Las Vegas, NV	3B	Hot, dry
Seattle, WA	4C	Cool, marine
San Francisco, CA	3C	Temperate, marine
Baltimore, MD	3A	Temperate, moist

3.3.4 Addressing Comfort

The literature provides little guidance with regards to occupant thermal comfort expectations in MM buildings, and this has a profound impact on the proposed simulation study. It is currently unknown whether occupants adhere to "deterministic" or "static" thermal comfort expectations—such as the PMV-PPD model—or whether adaptive comfort expectations prevail. Since MM buildings are, after all, a hybrid of conventionally conditioned and naturally ventilated buildings, one might expect that the answer lies somewhere between the two extremes. Occupants may, for example, switch between these two sets of expectations depending on the current mode of cooling in the building or the time of year.

Although this dissertation does not aim to answer this question, the simulation study needed to address this knowledge gap because thermal comfort plays a significant role in defining objective function penalties for the optimizer. To this end, a thermal comfort "bracketing" approach was adopted that allowed for consideration of three comfort models: 1) the ASHRAE 55-2004 PMV-PPD model, 2) the ASHRAE 55-2004 adaptive comfort model and 3) the EN-15251 adaptive model. Each building was investigated using three different thermal comfort penalties in addition to a fourth

energy-only objective function that ignored comfort impacts. Optimal policies were then compared to examine the sensitivity of solutions to comfort expectations and were benchmarked against the energy-only case to contrast performance with a purely energy-optimal strategy. One expects that the adaptive comfort models will generally provide more latitude for free cooling, allowing deeper energy savings, but that comfort considerations overall will significantly limit achievable energy savings.

As described in Equation 3.1, the objection function bears a discomfort cost/penalty to discourage solutions that result in poor thermal comfort. This approach is suggested because metaheuristic optimizers like PSO typically cannot enforce rigid inequality constraints and, more importantly, because ideal comfort is not always guaranteed, even in the best designed and controlled facilities.[1] Post-occupancy comfort evaluations in passively cooled buildings often examine exceedance of accepted thermal comfort boundaries. Therefore, thermal comfort penalties have been assigned in a similar manner, based on person-hours of exceedance experienced across the entire building model. Exceedances were heavily penalized only if the comfort was worse than in the base or reference models.[2] The goal is to allow the optimizer to explore solutions with equal or better comfort compared to a conventional building. An example penalty for the PMV-PPD comfort model that penalizes deviations from the 80% acceptability (±0.5 PMV) region would be

$$\begin{aligned} C_c &= k \sum^{\text{time}} \sum^{\text{zones}} n_{\text{occ}} \max\left(v_{\text{opt}} - v_{\text{base}}, 0\right), \\ v_{\text{opt}} &= \max\left(|\text{PMV}_{\text{opt}}| - 0.5, 0\right), \\ v_{\text{base}} &= \max\left(|\text{PMV}_{\text{base}}| - 0.5, 0\right), \end{aligned} \tag{3.8}$$

where n_{occ} is the number of occupants per zone, PMV_{opt} is the PMV value for a given zone,

[1] An alternate approach would be to use a multi-objective optimization to explore the frontier of Pareto-optimal solutions that balance energy and comfort considerations. However, given the computational burden already observed in preliminary offline optimizations, this approach does not seem feasible.

[2] Static comfort penalties compare against the base model, which does not use natural ventilation. Adaptive comfort penalties compare against the reference model, which represents conventional mixed mode operation.

PMV_{base} is the PMV of the same zone under base case conditions, v_{opt} and v_{base} are the number of violations for the optimal and base cases, respectively, and k is an arbitrarily high coefficient used to penalize comfort worse than the base case. Similar comfort penalties were formulated for adaptive comfort models by altering the violation terms, v_{opt} and v_{base}, to examine violations of operative temperature limits rather than PMV. These comfort penalties take the form

$$\begin{aligned} C_c &= k \sum^{\text{time zones}} \sum n_{\text{occ}} \max\left(v_{\text{opt}} - v_{\text{base}}, 0\right), \\ v_{\text{opt}} &= \max\left(\max\left[T_{op,\text{opt}} - T_{\text{upper}}, 0\right], \max\left[T_{\text{lower}} - T_{op,\text{opt}}, 0\right]\right), \\ v_{\text{base}} &= \max\left(\max\left[T_{op,\text{base}} - T_{\text{upper}}, 0\right], \max\left[T_{\text{lower}} - T_{op,\text{base}}, 0\right]\right), \end{aligned} \tag{3.9}$$

where T_{op} is the operative temperature in a given zone and $T_{\text{upper/lower}}$ are the upper and lower operative temperature limits allowed by adaptive comfort standards.

3.4 Rule Extraction

The approach of this research was to derive near-optimal guidelines for supervisory operation of MM systems from the results of offline optimizations using available supervised learning techniques. One can employ traditional parametric regression techniques or non-parametric data mining approaches (decision trees, support vector machines, "boosting" and so on) to approximate strategies discovered by the optimizer and to formulate those patterns in a computationally tractable manner that could be implemented in a BAS. Only selected techniques were explored under the scope of this dissertation, including generalized linear models (GLMs), classification and regression trees (CART), and adaptive boosting (AdaBoost). The methodology for each is clarified below.

3.4.1 Generalized Linear Models

GLMs are a rule extraction approach most closely tied to classical statistics, and are therefore presented first. Unlike multiple linear regression, GLMs do not directly relate regressors to responses. They are often used where least-squares linear regression could not appropriately model the process due to non-normal data (e.g. modeling binomially distributed values). Rather, they model the relationship between regressors and a linear predictor, θ, that is a transformation of the original responses. A link function, g, is chosen to match the distribution of the process being modeled (with binary variables, for example, the logit link function is used).[3] The GLM then relates the linear predictor θ to regressors by

$$\hat{\theta} = \hat{\beta}\mathbf{X}', \tag{3.10}$$

where $\hat{\theta}$ is a vector of the predicted values of θ, $\hat{\beta}$ is a vector of the estimated model parameters, and $\mathbf{X}'$ is an augmented matrix of predictor variables (the leading column contains all ones). Unlike standard multiple linear regression, the model parameters cannot be determined in closed form, and an iterative process must be used instead. The common approach is to choose parameters that maximize the model's likelihood function via an iterative optimization process, a process known as the maximum likelihood method (described in detail in Chapter 3 of [43]). Once $\hat{\theta}$ has been predicted, the values can be transformed back into the desired responses through an inverse link function (for example, the inverse logit function for binary responses).

As with all statistical modeling, GLMs require care on the part of the user to ensure that models are not over-fitted. This research adopted the commonly used stepwise regression approach to find right-sized and parsimonious models. The Akaike Information Criterion (AIC) was adopted to assess goodness of fit at each stage of the stepwise search. The AIC objectively measures the model's ability to reproduce the variance of the observations with the fewest model parameters [103], providing a goodness of fit value that balances model skill with the tendency to over-fit.

[3] Note that in the case that the link function $g(y) = y$ is chosen, GLM simplifies into standard, least-squares multiple linear regression.

Through stepwise regression, we seek to obtain a model with minimal AIC.

Once the rule has been formulated, implementation is fairly trivial and could easily be implemented in any modern BAS. Computing the linear predictor is a matter of simple arithmetic. For a given point in time, the process is simply

$$\hat{\theta} = \beta_0 + \sum_{i=1}^{n} X_i \hat{\beta}_i, \tag{3.11}$$

where n is the total number of predictors in the model. In reality, X would be provided by the building automation system through sensor data, such as insolation, ambient temperature, or zone temperatures. The linear predictor would then be back-transformed using the inverse link function appropriate to the GLM constructed. In the case of binary forecasting, the inverse link is the inverse logit function,

$$g^{-1}(\hat{\theta}) = \hat{p}_z = \frac{e^{\hat{\theta}}}{1 + e^{\hat{\theta}}}. \tag{3.12}$$

If the BAS does not support transcendental functions, the exponential term could be approximated with a Taylor series expansion. The result, $\hat{p}_z$ is the probability of a window opening, which when compared to a threshold value of 0.5 is converted into a binary signal. This binary signal can then be used to drive the availability of automated windows or the red/green light signal for a notification system. These techniques could readily be extended to forecasting of continuous optimizer decisions through the use of alternate GLM link functions.

3.4.2 Classification and Regression Trees

While GLMs can skillfully model a variety of distributions, their form (see equation 3.10) is not as readily understandable as typical control logic, which is often formulated in a series of if/then/else statements. A decision tree is significantly more comprehensible by human beings than a series of regression coefficients and may be a more viable approach in practice, where building operators need to clearly understand the control logic installed on a BAS. Classification

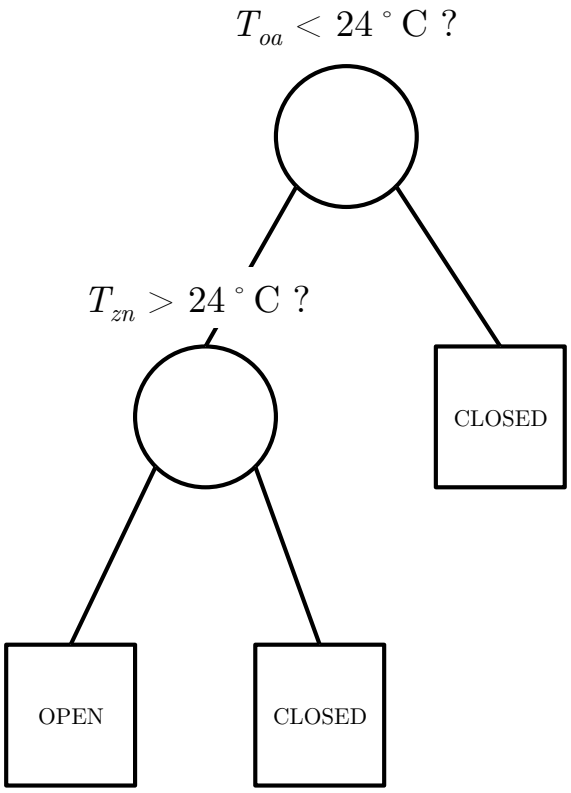
and regression trees (CART) are formulated as a cascade of binary decisions (Figure 3.7) that are directly translatable into nested if/then/else rules. Each node on the tree constitutes a binary decision, and each terminal node on the tree represents the model's response to a set of inputs.

Developing appropriately sized trees occurs in a two-step process. First trees are grown in such a way that they minimize overall classification error of a training/learning dataset of known classification. The learning dataset is akin to $\mathbf{X}$ in the GLM framework, and the known classifications might be the optimizer's chosen binary window states, $\vec{z}$. The process begins by testing a series of "splits" or binary rules on the dataset. Splits sort the dataset based on one feature or predictor at a time, so a simple split might be "$T_{oa} < 20$." Points in the training set that satisfy the split would be filtered to the lefthand side of the node, other points to the right. The best split for a node most effectively separates the training set into its different classes. This process continues until a stopping criterion has been reached or until all current terminal nodes have points of uniform class.

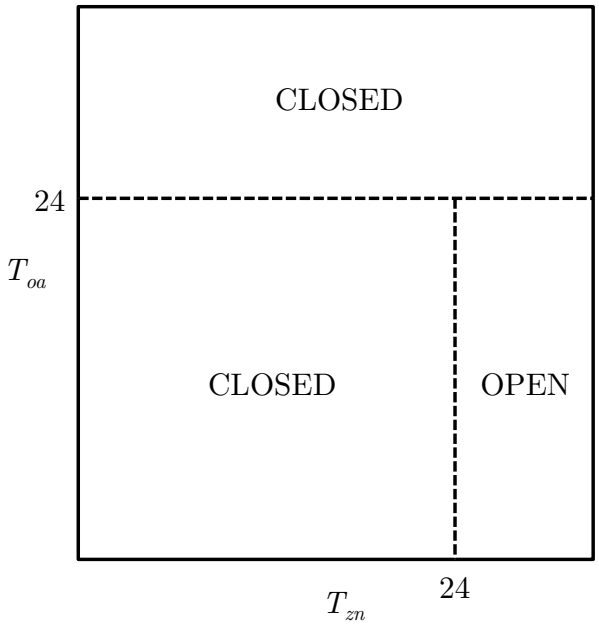
As with classical regression techniques, there is a tradeoff between model complexity and prediction skill. A rudimentary measure of model skill is the misclassification error, $R(\mathrm{T})$, which tallies the total percentage of points misclassified by the tree T. However, if one guides the tree building approach solely to minimize $R(\mathrm{T})$, it will yield extremely large trees that over-fit the data (in fact, this will yield trees that have terminal nodes with only one training point in them, lowering the $R(\mathrm{T})$ to zero). To balance model complexity with performance, Breiman *et al.* introduced a cost complexity parameter, R_α, which combines misclassification cost/error with a cost penalty for complexity. It is given by

$$R_\alpha(\mathrm{T}) = R(\mathrm{T}) + \alpha|\mathrm{T}|, \tag{3.13}$$

where α is a non-zero scalar parameter and $|\mathrm{T}|$ is the "complexity" of the tree T (simply the sum of its terminal nodes). For each step in the tree growing process, a cost-complexity estimate is made and recorded. The cost-complexity parameter allows for the crucial second step of the CART process, the backwards pruning of trees to a minimal cost-complexity point. A rule of thumb



(a) Tree diagram



(b) Feature space segmentation

Figure 3.7: The left diagram shows a simple two-level CART to determine if windows should be opened in a space based on outdoor and indoor temperatures. The resulting classifications of the feature space are shown to the right.

provided by Breiman et al. is to back-prune trees to the first sub-tree that falls within one standard error of the minimum R_α value. The rule is intended to find the simplest tree with performance comparable to a tree with minimum R_α [21]. Figures 3.8 and 3.9 provide a visual representation of back-pruning for a particular tree that has been reduced from 50 to 9 terminal nodes with no meaningful performance degradation.

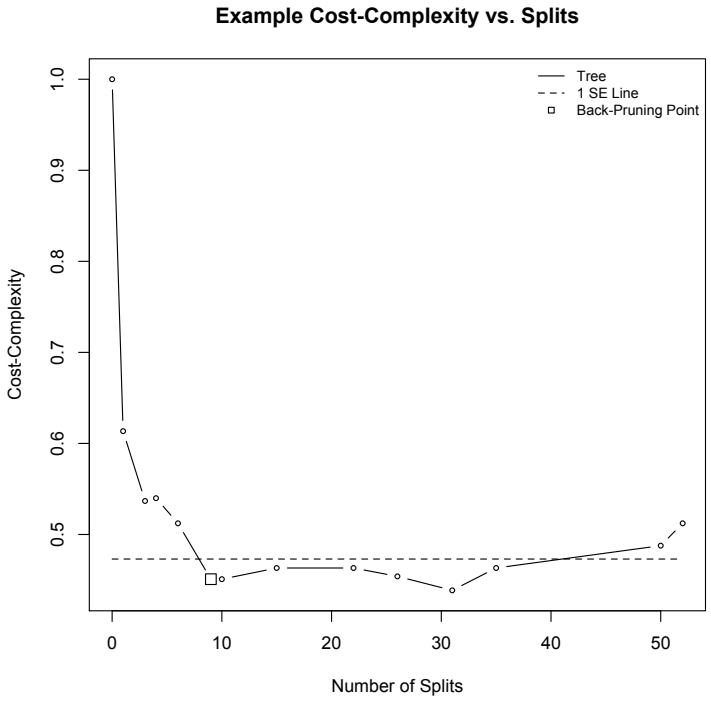
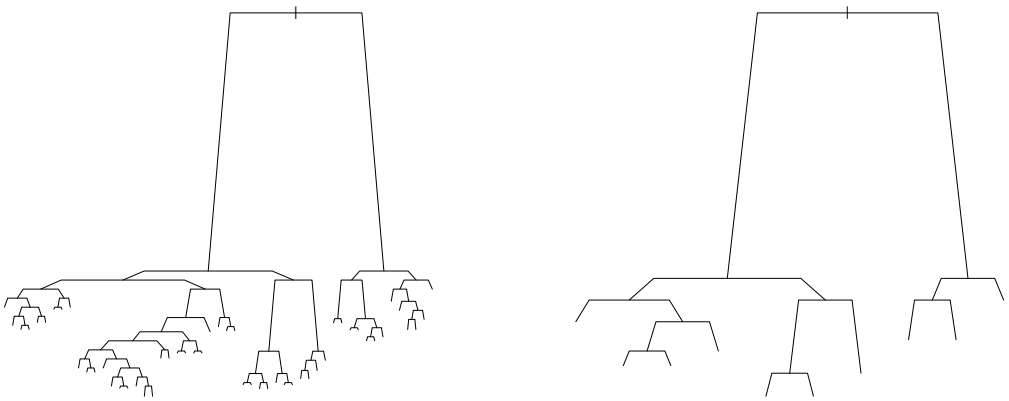


Figure 3.8: Increasing tree size can reduce misclassification error, but raises overall cost-complexity and results in over-fitting. Back-pruning trees to a minimum cost-complexity point—denoted with the square above—yields right-sized trees. The 1 standard error rule is employed, with the standard error threshold indicated by the dashed line.

Only the high-level tree growing methodology is addressed here; however, there is a rich theoretical framework available for CART in [21].



(a) Unpruned tree

(b) Pruned tree

Figure 3.9: A CART model is reduced from over 50 splits down to nine splits through back-pruning.

3.4.3 Adaptive Boosting or AdaBoost

Adaptive boosting was first successfully implemented by Freund and Schapire in the mid-1990s and has evolved into one of the more popular and robust classification algorithms. The core principle of the original discrete "AdaBoost" is that a linear combination of so-called "weak learners" can "vote" together and synergistically produce a "strong classifier" [41]. A weak learner is a classification rule that, in isolation, exhibits very poor performance—often only slightly better than chance. A good example of a weak learner is a "decision stump" or a one-layer classification tree, identical to the CART models described in the previous section. Mathematically, a weak classifier, $h(\vec{x})$, simply evaluates the feature vector, $\vec{x}$, of a given point and produces a +1 or -1 classification. The boosting algorithm identifies a linear combination of the best m weak learners and assigns weights, a, to the classifications produced by each. The sign of the resulting linear combination of "votes" determines which class is "elected" by the boost, given by

$$H(\vec{x}) = \text{sign}\left(\sum_{i=1}^{m} a_i h_i(\vec{x})\right), \tag{3.14}$$

where m is the number of weak learners in the series and a_i is the weight corresponding to weak learner h_i. Weights can usually be determined in closed form.

Boosts inherit some of the desirable properties of CARTs. Despite their complex behavior, they are expressed in if/then/else logic and simple arithmetic, meaning that they are also at least feasible for use in most building automation systems (although a large boosting model can consume hundreds of lines of code). Unfortunately, because boosting models could comprise dozens of CARTs "voting" together, their logic can be a bit more difficult to visualize and understand, placing them somewhere between CARTs and GLMs in terms of comprehensibility.

3.4.4 Skill Evaluation

3.4.4.1 Open and Closed Loop Performance Evaluations

The skill of a given rule in reproducing the original optimizer control sequence can be evaluated in two ways. When statistical or machine learning models are first trained, the rough performance of the rules can be gauged by an **open loop** test. In this paper, the rule is trained on approximately two thirds of the optimizer results and then tested on the remaining third of the dataset, the cross-validation set. The rule forecasts control responses for the entire cross-validation period using predictors from the cross-validation set. Predictors in the cross-validation set contain the states present in the original optimizer solution ($\vec{x}^*$). Evaluation of the rule results in a near-optimal control vector, $\vec{u}'$. Open loop tests are fast to implement and may be useful for comparing several rule formulations, but they do not reflect real world performance.

Closed loop tests more realistically depict actual performance. The rule is embedded in a building energy model and predicts current timestep actions using state information from the previous timestep of the simulation. This is the closest approximation of embedded performance on a "live" BAS. The loop is "closed" because actions taken by the rule in the current timestep directly impact the predictor set used by the rule in its next iteration. Misclassification errors made by the rule can quickly cause thermal states—and thus the predictor set—to diverge from those experienced during the optimal solution.

Closed loop tests can be challenging to implement, as many building energy simulation programs were not designed for detailed control investigations. That said, supervisory control investigations like the ones proposed here are usually not too challenging to evaluate. EnergyPlus, for example, enables some control customization through its Energy Management System objects.[4] To speed the process of translating rules into the EnergyPlus Runtime Language (ERL), several model parsing and translation tools were developed in **R**® [37] to quickly and accurately translate

[4] When those capabilities are inadequate, tools like the Building Control Virtual Test Bed (BCVTB) are available to expand EnergyPlus and can even be used to exchange information in real time between the simulation engine and a technical computing platform like Matlab, which might be used by some users for the rule extraction process.

R model objects into useable ERL code with minimal manual input. These translators enable quick implementation of GLM-, CART-, and boost-based rules extracted using **R**'s `glm`, `rpart`, and `ada` packages, respectively. Sample code for the tools is provided in Appendix C.

3.4.4.2 Statistical Performance Measures

For the present example of binary window control, we adopt two performance measures relevant to categorical forecasting and classification. We can measure rule performance against the original MPC solution through a ranked probability score (RPS), demonstrating the degree to which the model predicted the original optimizer results. However, this information is only partly useful, since the optimizer signal might just as easily be reproduced by a white noise process. We can therefore compare the RPS of the model against the RPS for a random process. This is done through the ranked probability skill score (RPSS), which has been used in various climatological contexts to compare model skill in predicting categorical rainfall and streamflow quantities [82] and is described in detail by Wilks [103]. The RPSS compares the accuracy of model predictions against chance, but rather than simply comparing our model against a 50-50 chance of a window opening, we compare against the probability of window openings observed in the optimizer results, which tends to be much lower. The RPSS is negative if model results are worse than chance, 0 if model results reproduce chance events, and positive if model results are closer to the original observations than chance. A 1 represents a perfect score.

RPS is computed by dividing window opening predictions into j categories, in this case two for binary control states. A vector of forecast probabilities, p_j, is constructed based on the rule predictions; a vector of observed events, z_j, is constructed from the optimal results. We then take the cumulative density function of p_j and z_j, resulting in the two-category vectors, $p_{cdf,j}$ and $z_{cdf,j}$. Note that, in our case, the RPS is computed for each instance that a window opening is predicted by the rule, even if the rule executes more frequently than would be allowed by the MPC time blocking scheme. The RPSS is then the ratio of the mean RPS values:

$$\mathrm{RPS} = \sum_{i=1}^{j} \left(p_{cdf,i} - z_{cdf,i}\right)^2 \tag{3.15}$$

$$\mathrm{RPSS} = 1 - \frac{\overline{\mathrm{RPS}}_{\mathrm{model}}}{\overline{\mathrm{RPS}}_{\mathrm{chance}}}. \tag{3.16}$$

While RPSS provides valuable information on the match between probabilistic forecasts and the original optimizer control sequence, we are also interested in the degree of conformity between the actual binary signal generated by the rule and the corresponding optimizer sequence. For this, we can use the misclassification error, $R(\mathrm{T})$. To further strengthen rule performance, it can be useful to categorize different types of errors (i.e. false positives and false negatives). We adopt the nomenclature $R(i|j)$ to denote classification errors in which the true classification was i, but the forecast classification is j. For example, in the context of binary window control a $R(0|1)$ error would represent errors incurred by opening windows that should have been closed.

Chapter 4

MPC Validation Cases

This chapter presents results of several validation cases, mainly to demonstrate intuitive behavior on the part of the PSO optimizer and examine the validity of the initialization horizon approach presented in Chapter 3. Proof of concept on the rule extraction approach is provided in Chapter 5.

4.1 PSO Tuning and Validation Using Pseudo Problem

The parallel PSO algorithm was tested for performance by conducting a one-day MPC pseudo problem designed to mimic the features of actual optimization cases. A pseudo problem is used because of the high computational cost associated with optimizing actual EnergyPlus models. The pseudo problem, like many of the actual MM cases evaluated in this research, is a function of binary and continuous variable vectors, each of length 12 (this would be akin to breaking a 24-hour planning horizon into 12 two-hour modes). When box constraints and discretization of continuous variables is taken into account, the optimizer explores a decision space on the order of 1×10^7.[1] The function represents a case in which one would simultaneously optimize zone cooling set points and window positions for an entire building. The function evaluator was designed to contain regions of equivalent cost (hyperplanes) and discontinuities, as observed in real MPC cases (discussed later in this chapter). It has a global optimum of 280. Sample code for the function has been provided in Appendix C.

[1] Conducting an exhaustive search on this space using an actual EnergyPlus model would take approximately 9.5 years if run serially!

Tests were conducted comparing the performance of the PSO-C and two modified versions. The CV-RMSE between current swarm objective function values and the global best value was used to establish convergence, per Equation 3.3. Convergence tolerances were evaluated down to 0 to determine whether the optimization would prematurely converge as a result of particle extinction (all particles stop moving, but do not globally converge). Because of the stochastic nature of PSO, 10 optimizations were conducted, and batch statistics on the solutions were generated. A run was conducted using the Nelder-Mead simplex method (Matlab's `fminsearchconstrained`) for comparison. The algorithm timed out after reaching a ceiling of 20,000 objective function calls, with an objective function value of 295, 5% higher than the optimum. Results for PSO runs are compiled in Table 4.1. The addition of penalty-based constraint enforcement and stochastic handling of binary variables, discussed in Chapter 3 improved overall skill in finding the global optimum value, but with significantly more function calls. In effect, we eliminated the premature convergence problems, found global optima with 100% success, but increased function calls by an order of magnitude over the basic PSO-C algorithm of Clerk and Kennedy.

Table 4.1: Optimizer Performance Comparisons: PSO-C

		n Simulations		
Algorithm Description	Success Rate	Mean	Min	Max
PSO-C	10%	473	366	576
w/constraint penalties	90%	1,769	1,200	2,689
w/binary variable handling	100%	12,713	11,610	13,429

With this algorithm in place, the CV-RMSE tolerance was adjusted to investigate whether tolerances could be relaxed somewhat to speed convergence. As Table 4.2 shows, increasing the convergence tolerance up to values of 0.3 had little impact on the overall skill of the optimizer but did manage to roughly quarter the number of simulations required to converge. The pseudo problem showed little sensitivity to this tolerance; performance only started to drop off at a relatively high tolerance of 0.5. Because actual optimizations are expected to have a more noisy objective function

than the one presented by this pseudo problem, tolerance values of at most 0.20 have been employed for the offline MPC study.

Table 4.2: PSO Optimizer Tuning for Convergence Tolerance

		n Simulations			CV-RMSE at Exit		
Algorithm Description	Success Rate	Mean	Min	Max	Mean	Min	Max
Modified PSO	100%	12,713	11,610	13,429	0.034	0.020	0.045
CV-RMSE=0.15	100%	3,921	2,031	6,088	0.126	0.064	0.149
CV-RMSE=0.20	100%	3,557	2,110	5,456	0.175	0.127	0.199
CV-RMSE=0.30	100%	2,998	1,788	7,006	0.249	0.154	0.298
CV-RMSE=0.50	90%	1,763	589	3,566	0.478	0.425	0.499

4.2 Early Conceptual Validation of MPC Environment

Two simple optimization exercises were conducted with an early version of the MPC environment to examine the impacts of optimizing window positions in building models compared to the use of supervisory hybrid ventilation controllers integrated into EnergyPlus. One optimization, denoted as "constant," optimized the binary opening and closing of window positions.[2] Simulations were conducted under Boulder TMY3 weather for a single 24-hour period on two different days, June 1 and the 1% cooling design day. Binary window opening decisions were broken into six 4-hour blocks spaced throughout the day. DX cooling energy was used as the objective function. For the second optimization, we manipulated a temperature setpoint schedule that governs the opening and closing of windows at the zone level. The decision space was discretized such that only integer temperature values were allowed. Furthermore, fairly wide lower and upper box constraints (15 to 40 °C) were used so as not to artificially constrain the solution.

Two additional runs were conducted for comparison purposes, including a case in which all windows were shut (the base case) and another case in which the original hybrid ventilation

[2] The term constant derives from the style of schedule enforcement in EnergyPlus and does not imply constant or static window positions.

controller was allowed to coordinate HVAC and natural ventilation operation.

In the interest of simplicity a one-story building with three interior thermal zones was modeled. The model file is a slightly modified version of the "HybridVentilationControl.idf" example file that is provided with EnergyPlus. The building consists of three conditioned zones, each with operable windows; an unconditioned attic space rests above. Cooling is supplied by a rooftop DX system, serving a single air loop with three terminal boxes. Heating was not considered in the investigation because optimizations were conducted in the cooling season. A simple 3D view of the building geometry is provided in Figure 4.1.

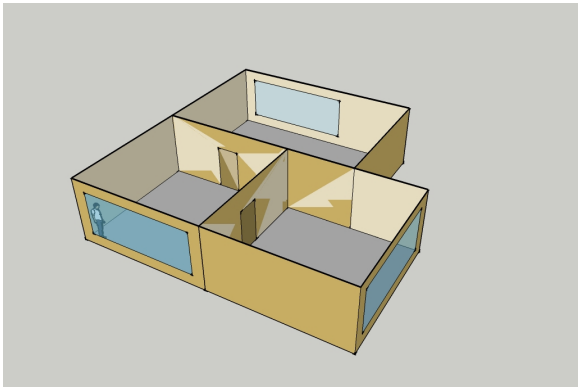
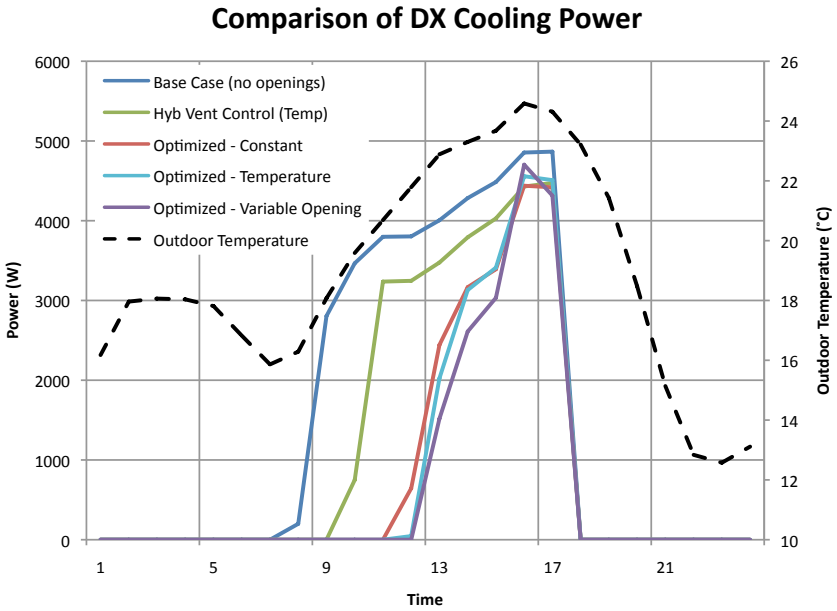


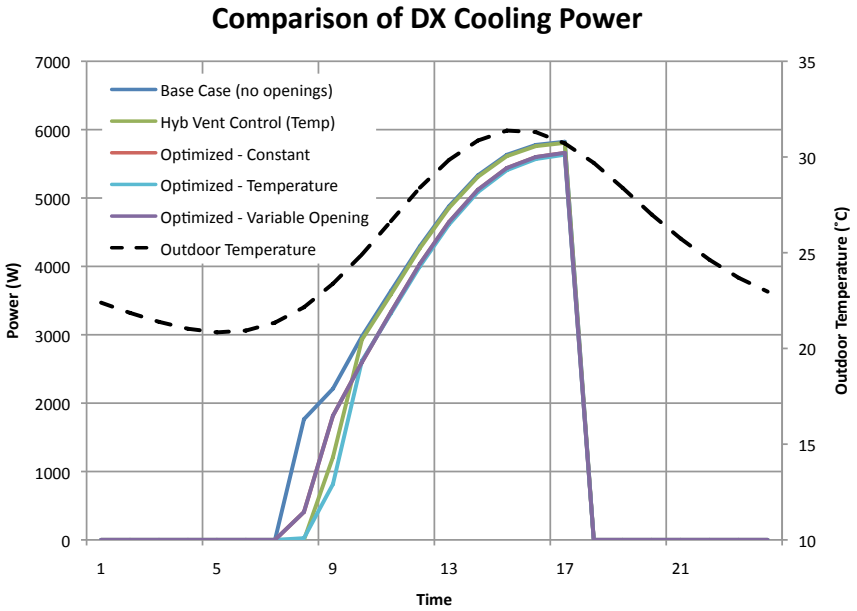
Figure 4.1: 3D view of simplified MM building model.

For the June 1 optimization—a mild, early summer day—the optimal solutions reduced the day's DX cooling energy by nearly 50% compared to the base case, mainly by delaying the onset of mechanical cooling by utilizing cool morning air to ventilate and cool the space. The solution also outperforms the built-in controller. Figure 4.2(a) illustrates the cooling power consumption for the compact DX system for the different optimization and base cases described above. Opportunities for natural ventilation cooling are significantly limited on the cooling design day due to extremely high ambient temperatures. The optimizer effectively has very little latitude to improve on the the hybrid ventilation controller, and thus cooling loads are only slightly delayed (Figure 4.2(b)).

These early results provided some simple validation that MPC is able to provide significant additional savings over conventional MM control heuristics.



(a) Mild day



(b) Design day

Figure 4.2: DX cooling power profiles for a mild swing season day (June 1) and the design day, Boulder, CO.

4.3 Initialization Horizon Approach

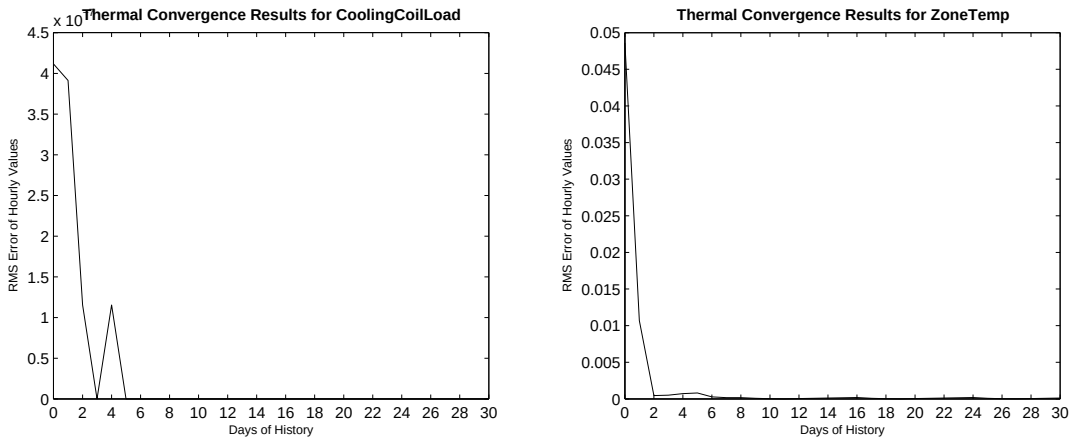
The use of an initialization horizon to reset the thermal state of the building as the optimization advances has not been applied before, and thus a process was developed to ensure that I was of appropriate length to effect a low-error transfer of thermal state in multi-day, receding horizon MPC problems.

To begin, a given building energy model is simulated for a full year on TMY3 weather data, and several outputs are captured, including zone temperatures and cooling equipment loads. These outputs are meant to serve as a proxy for the complete thermal state of the building. The goal is to minimize the root mean square error (RMSE) between the results of the annual simulation and a simulation of shorter duration for a one-day period. As the length of the initialization horizon is increased, there will be a point at which the RMSE plateaus and where the annual and I-length simulation results are in agreement. The RMSE is defined here as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{24}(y_i - y_{a,i})^2}{24}}, \tag{4.1}$$

where $y_{a,i}$ is the annual simulation output value at hour i and y_i is the corresponding value from the shorter simulation.

Sample results can be viewed for SBC (Figure 4.3). The RMSE evaluated for various lengths of I is shown in Figure 4.3 below. Due to the lightweight construction of the base case building, the longest time to converge is about 6 days, thus an initialization horizon of about one week would be used for this particular model. The RMSE values are significantly higher for zone temperatures than for the cooling coil load, which is intuitive given that the cooling coil for an individual AHU sums together the loads from various zones in the building, thus smoothing out disparities between zones. Regardless, errors even for zone temperatures are so small (less than 0.5%) after the one-week initialization horizon that they are well within the noise of simulation error itself.



(a) Cooling coil

(b) Zone temperature

Figure 4.3: Thermal convergence tests based on convergence of cooling coil loads and zone temperatures show that one week provides a sufficient initialization horizon for this energy model.

4.4 The Issue of Solution Equivalence

Many of the examinations into MM building supervisory control presented herein were formulated to explore tradeoffs between mechanical cooling and NV. One of the easiest and most practical ways to do this from a supervisory standpoint is to formulate the optimization problem in terms of decisions on cooling setpoints and NV signals. Optimization of zone cooling setpoints is a common MPC approach presented in numerous prior studies [49, 59]. Interestingly, this problem formulation leads to some issues in interpreting solutions that are not widely discussed in the available literature. Early MPC runs conducted for the offline MPC simulation study resulted in significant "noise" in the setpoint portion of the solution,[3] which is not satisfying or confidence-inspiring when attempting to identify globally optimal solutions. One would ideally like to see "crisp" solutions that can readily be attributed to physical properties of the building, weather sequences, or other logical problem parameters.

Some amount of this noise can be attributed to equivalence in MPC solutions or solutions with identical objective function values. For example, in the binary portion of the solution space, when

[3] The term noise is used here to denote solutions with abrupt discontinuities that are not attributable to any known phenomenon, either physical or numerical, present in the problem.

decisions are made on relatively short time scales (e.g. one decision per hour), the binary window opening sequence 001 might very well yield the same objective function value as 010 if physical states during hour 1 and hour 3 are similar enough. These types of time shift equivalence issues were eliminated by parsing the MPC problem coarsely enough in time (usually in two- or three-hour blocks) that significant objective function values existed between permutations of the binary sequence. The optimizer can also explore tradeoffs between competing elements of the objective function, such as the tradeoff between cooling and heating energy or tradeoffs between comfort and overall HVAC energy use. Comfort-energy tradeoffs were effectively eliminated by applying a large comfort penalty coefficient to the comfort penalty term, putting any comfort violation costs on a different order of magnitude compared to facility energy use.

The setpoint portion of the decision space, however, represents a different challenge because setpoints introduce discontinuities into the objective function resulting from the evaluation of inequalities in controllers and thermostats. Furthermore, any cooling setpoints above the zone air temperature will result in the same step change in cooling power, resulting in large hyper-plateaus in the objective function. This often results in a number of equivalent optimal solutions, in effect a broader "valley" of optimal solutions rather than one deeply buried, globally optimal point. In Figure 4.4, equivalent setpoint vectors are box-plotted for an MM1 case evaluated on San Francisco weather, highlighting areas of insensitivity to setpoint decisions, which intuitively occur early in the morning when zone temperatures are relatively low and internal gains are not present. At these times, setpoints can be chosen throughout the entire range of the box without having a significant impact on the objective function. The simplified case presented was an optimization conducted using an energy-only objective function. The optimizer correctly maximized cooling setpoints, but explored a number of more "jagged"equivalent solutions during each planning horizon that would have been equally viable to implement. The solution shown in black was simply the first policy that resulted in a minimum.

For the remaining offline MPC cases evaluated in this dissertation, the solutions presented represent only one of the equivalent solutions, namely the first identified by the optimizer. In the

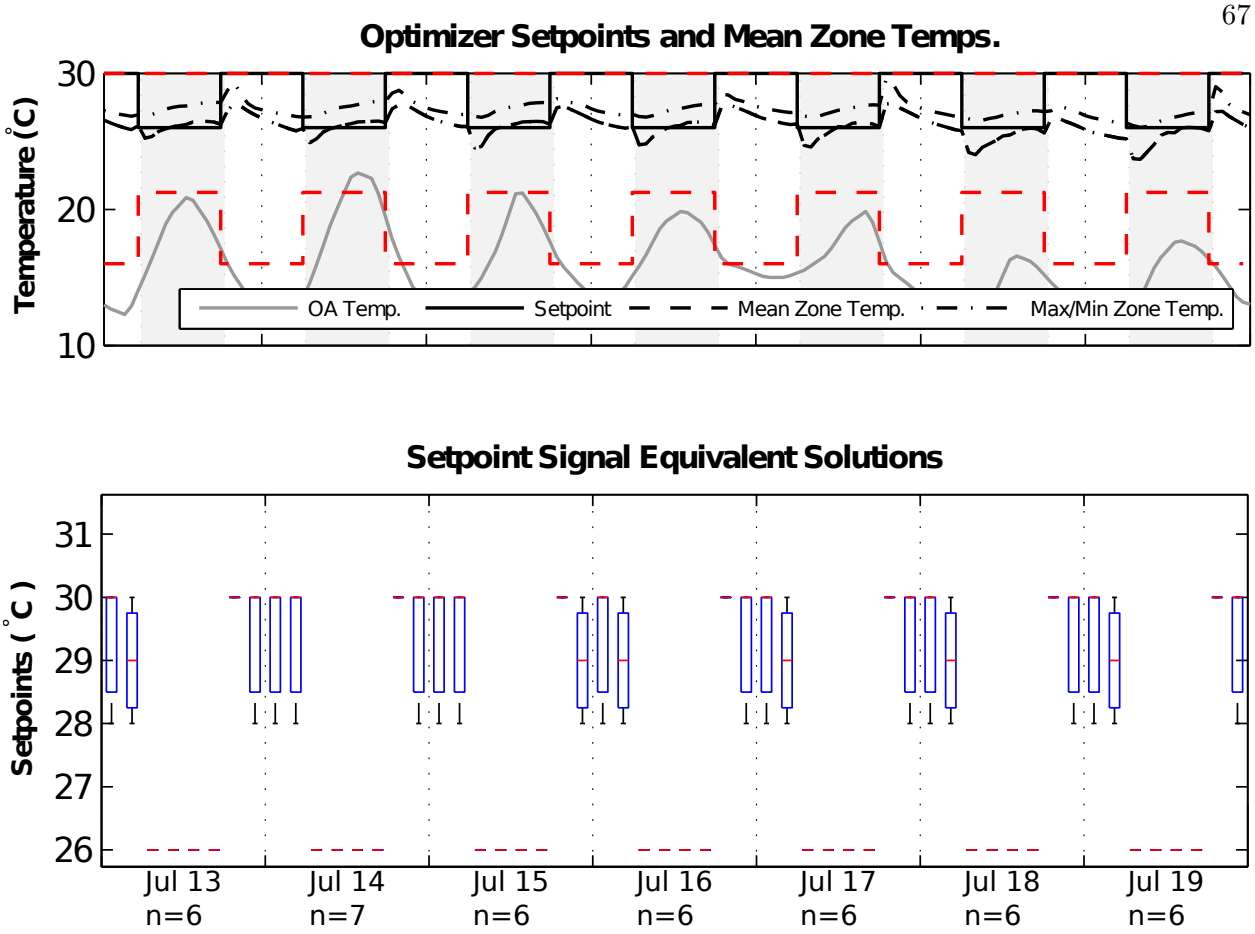


Figure 4.4: Equivalent setpoint solutions for the MM1, cooling season, energy-only objective function, San Francisco weather. The number of equivalent solutions found for each day is labeled below the abscissa.

subsequent rule extraction process, two different approaches were attempted to address equivalent solutions. Initially, training sets contained all the equivalent solutions for each planning horizon, and rules were trained on these "expanded" solution sets. However, this approach proved problematic, because some portions of the training set contained more equivalent solutions than others and therefore had more "weight" in the training process. Therefore, it was desirable for the purposes of rule extraction to identify a single optimum for each planning horizon. To this end, the training set was post-processed to identify the maximum cooling setpoint sequences for each planning horizon, allowing the largest setpoint setups that still yielded an optimum objective function value. These "maximum allowable" solutions then comprised the training set.

Chapter 5

MPC and Rule Extraction for a Binary Window Control Problem

The following chapter expands upon results published in Building and Environment in February 2011 and the Journal of Building Performance Simulation in March 2012 on the application of MPC to simple MM window control problems and the subsequent extraction of rules from the offline MPC results [69, 70]. Note that many of the optimizer refinements discussed above were not incorporated in time for these results, but this is inconsequential since the emphasis here is to demonstrate and compare the effectiveness of several different rule extraction approaches on a large set of optimization results as a validation of the approach.

5.1 MPC Results

Offline MPC was conducted on the MM1 building model using publicly available TMY3 weather data for Boulder, CO. The total window of the optimization spanned June 15 through August 30, exactly 11 weeks. Decisions were carried out and executed over a 24-hour planning/execution/cost horizon ($P = E = C$), and the optimizer manipulated a single binary decision vector for global window open/closed position in 2-hour blocks, for a total of 12 decisions per day. A base and reference case were run over the same period for comparison. The base case is the standard DOE benchmark building (SBC from Table 3.1), without any natural ventilation, whereas the reference case is MM1, but with occupant window control per the mean response of the Humphreys algorithm.

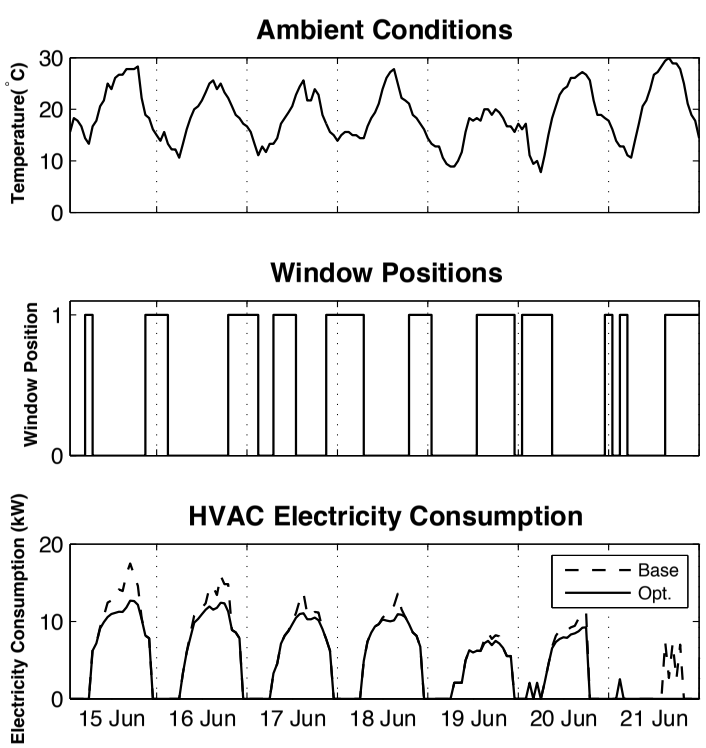
For this early case, the objective function included electric energy use of the cooling equipment

in the building (both fans and DX cooling equipment) and a penalty to discourage frequent state transitions, limited to 5% of the total cooling energy use of the building.

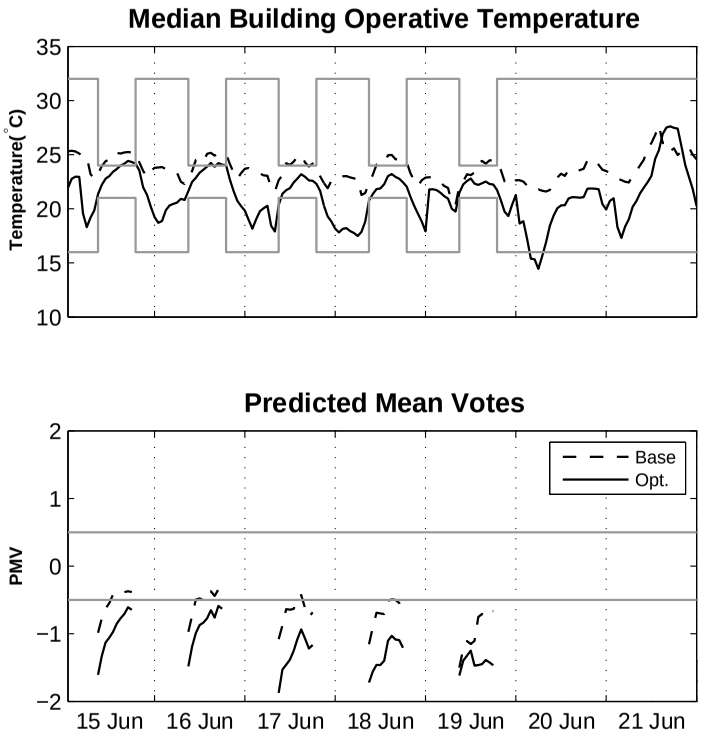
$$C(\vec{x}) = C_e(1 + 0.05C_{\text{switch}}) \tag{5.1}$$

C_e is the cooling electricity use and C_{switch} is the penalty term.

Figures 5.1(a) and 5.1(b) illustrate results from a mild week in June. In 5.1(a), the upper graph shows ambient temperatures over the week; the middle graph shows the optimal solution alongside the mean window opening behavior in the reference model; and the bottom graph illustrates electric power consumption for HVAC equipment for all three cases, showing time periods during which savings accrue. The optimal solution found resembles a night ventilation strategy, with the optimizer opening windows during cooler nighttime periods. This form of passive thermal energy storage utilization allows the building to ride out some of the daytime cooling loads without the need for mechanical cooling. Load reductions are modest for the week shown, but the optimizer was able to provide 1,630 kWh (13%) in electricity savings over the entire cooling season. The electricity savings accrue most noticeably on the cooling peak in the afternoon. As shown in Figure 5.1(b), operative temperatures are uniformly cooler and predicted mean votes lower than in the base case due to the night venting, particularly during early occupancy. Expanding the objective function to include gas energy use or the inclusion of thermal comfort penalties could remedy this, but since results are used here to train and demonstrate simple rules, these steps are ignored.



(a) June Energy Savings



(b) June Comfort

Figure 5.1: 5.1(a) shows the optimal solution and energy use profile for a typical week in mid-June. The optimal solution outperforms the base case through a night ventilation strategy. 5.1(b) provides a summary of comfort conditions in the space. Cooling and heating setpoints are denoted by the gray lines above and below the zone temperatures, respectively. Note that zone temperatures and predicted mean votes are low, particularly at the beginning of occupancy, due to night ventilation. PMV values have been removed for unoccupied periods.

5.2 GLM Rule Extraction

Three GLMs, summarized in Table 5.2, were formulated to compare different parameter sets and parameter pruning approaches. The predictor variables considered are shown in Table 5.1. The goal of the GLM is to replicate optimizer window opening decisions. Since the response to be modeled varies binomially, the logit link function was used to transform responses, z, into the linear predictor θ. Predictions ($\hat{\theta}$) are made and back-transformed with the inverse logit function, yielding probabilities of windows opening or closing. If the probability exceeds a threshold of 0.5, an opening results; otherwise, windows are closed. Models were trained on data from the first seven weeks of the summer, then cross-validated against the last four weeks of data.

Table 5.1: Predictor Variables Considered

Variable	Description
T_{oa}	Outdoor dry bulb temperature
T_{dp}	Outdoor dew point temperature
v_{wind}	Wind speed
θ_{wind}	Wind direction
I_{dn}	Direct normal solar radiation
T_{core}	Core zone temperature (first floor only)
$T_{\text{floor,zone}}$	Mean temperature for a given floor and zone (total of 10)
z	Binary window state at a given point in time

A stepwise regression approach was used to find the parameter set that minimized the model AIC. With each successive model, a larger number of lagged (prior timestep) predictors were included to attempt to capture process memory. Model 1 utilized only current timestep predictors ($\mathbf{X}_t$), whereas model 2 included previous-hour predictors as well ($\mathbf{X}_{t-1}$). Model 3 included the previous hour's optimal window state (z_{t-1}) as a predictor in addition.

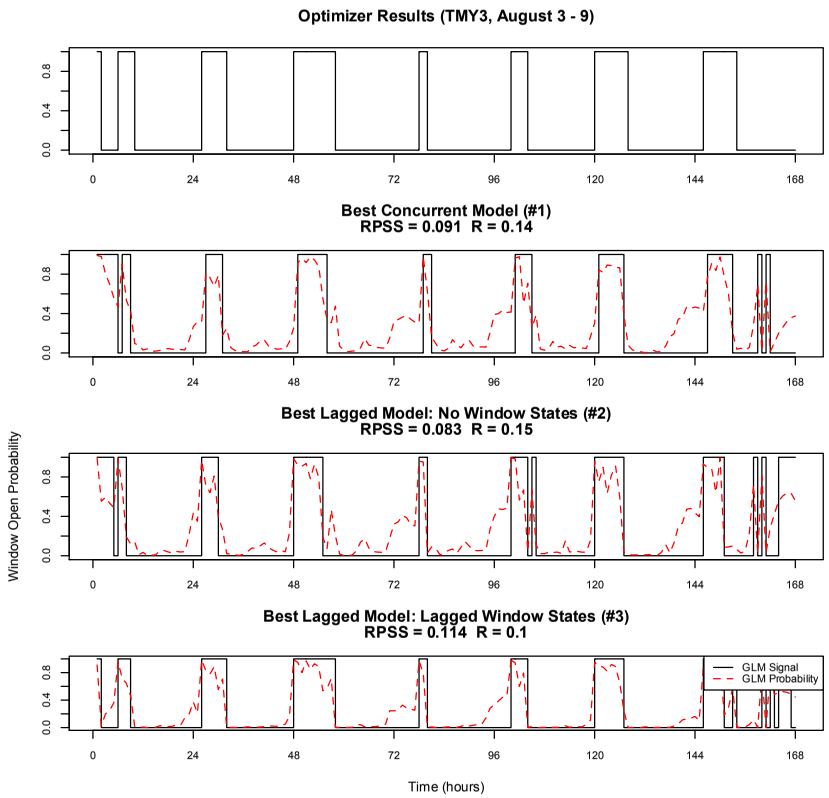
5.2.1 Open Loop Performance

The resulting cross-validation predictions (i.e. predictions made for points not present in the training set) and the original optimizer sequence are presented in Figure 5.2(a) for the week of

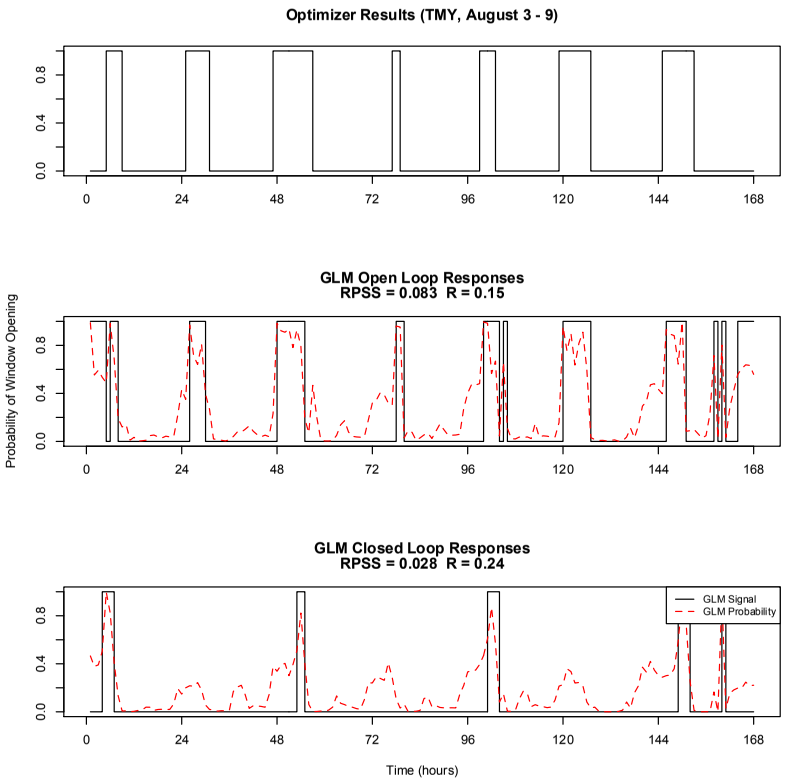
Table 5.2: Model Formulations Considered

	Predictor Types		
Model	$\mathbf{X}_t$	$\mathbf{X}_{t-1}$	z_{t-1}
1	•		
2	•	•	
3	•	•	•

August 3 through 9 (the first week of the cross-validation period), with probabilistic predictions as a dashed line. Model skill is evaluated using the RPSS as well as the misclassification error. All models perform well compared to the original data, tracking periods of opening and closing accurately, even for very brief periods, such as the 2-hour opening occurring on the first night of the week. However, note that models 1 and 2 also miss several long periods during which the optimizer windows were open, namely the beginning of the sixth and seventh nights.



(a) Open loop test comparison



(b) Closed loop test comparison

Figure 5.2: Open and closed loop predictions of window position for models 1 through 3 are presented to the left alongside offline MPC optimal results for the first week of the cross-validation period. The charts above show the increasing disagreement between the extracted rule and the original optimizer signal as one progresses from open loop to closed loop performance tests.

5.2.2 Alternate PCA-Based Model Formulation

Selecting a "best" GLM predictor set in this instance is a difficult task due to the large degree of multicollinearity inherent in the predictor set. A correlogram of all temperature-based predictors (Figure 5.3) illustrates that the building's zone temperatures are highly correlated to each other. Correlation values are in the 0.9 range between all perimeter zones, as these zones all experience the same envelope loads. Consequently it is difficult for a stepwise regression algorithm to discern which combination of zone temperatures might produce the best AIC values. To this point, during the stepwise fitting process, 17 other candidate parameter sets were found with AIC within 0.5% of the best value.

One way to quickly eliminate this ambiguity is to eliminate the multicollinearity issues altogether through PCA. PCA was performed on the GLM 2 predictor set (zone temperatures only) prior to model fitting. Because GLM 2 includes consideration of both concurrent and lagged predictors, two separate PCAs were conducted. Examination of the eigenspectrum revealed that over 90% of the zone temperature variance could be captured in the leading 3 modes of variance. In electrical engineering parlance, this would represent very effective signal-noise separation.

It is instructive to examine the coefficients of the leading eigenvectors to better understand the physical meaning behind the leading PCs. Figure 5.4 plots the eigenvector coefficients for the leading three PCs. The greater the absolute value of a coefficient, the stronger the influence of that term. Note that the eigenvectors each correspond to an easily recognizable physical phenomenon in the building. The leading mode of the decomposition heavily weights perimeter zones that all experience envelope loads and, more importantly, natural ventilation. The second mode more heavily weights the core zone and de-emphasizes the perimeter. The third mode corresponds directly to zone orientation. Perimeter zones with western orientation receive a large positive weighting; those with eastern orientation have a large negative coefficient; and those with north-south orientation receive a coefficient similar to the core zone. Thus we can capture three physical phenomena in a very compact form.

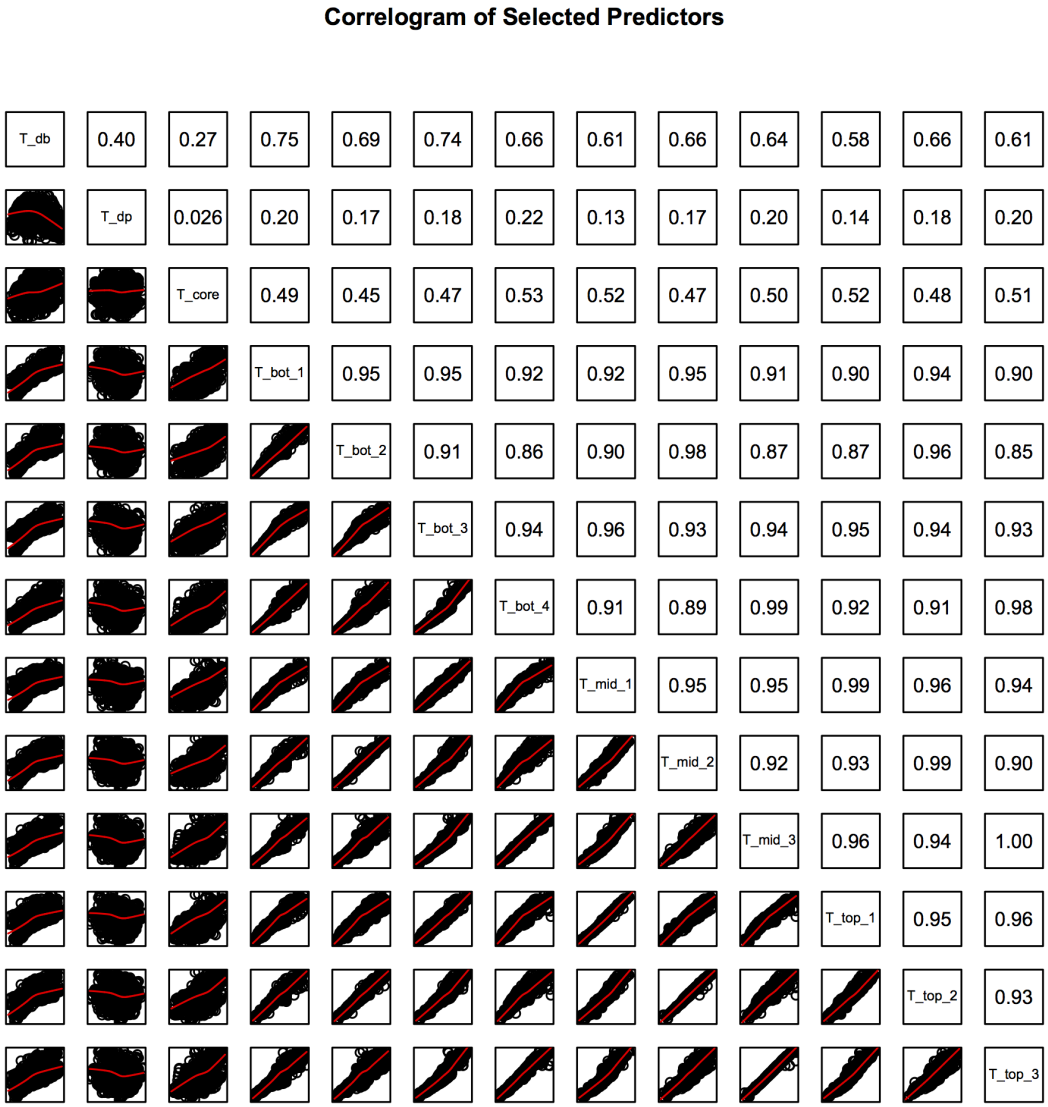


Figure 5.3: The above correlogram shows grid plots of all temperature predictors, with correlation values listed in the upper half of the grid. Variables T_{db} and T_{dp} represent dry bulb and dew point temperatures, respectively. All other variables are zone temperatures. T_{core} is the temperature of the core zone on the first floor of the building, whereas all other zone temperatures have some envelope exposure. The zone temperatures—particularly perimeter zones—are highly multicollinear.

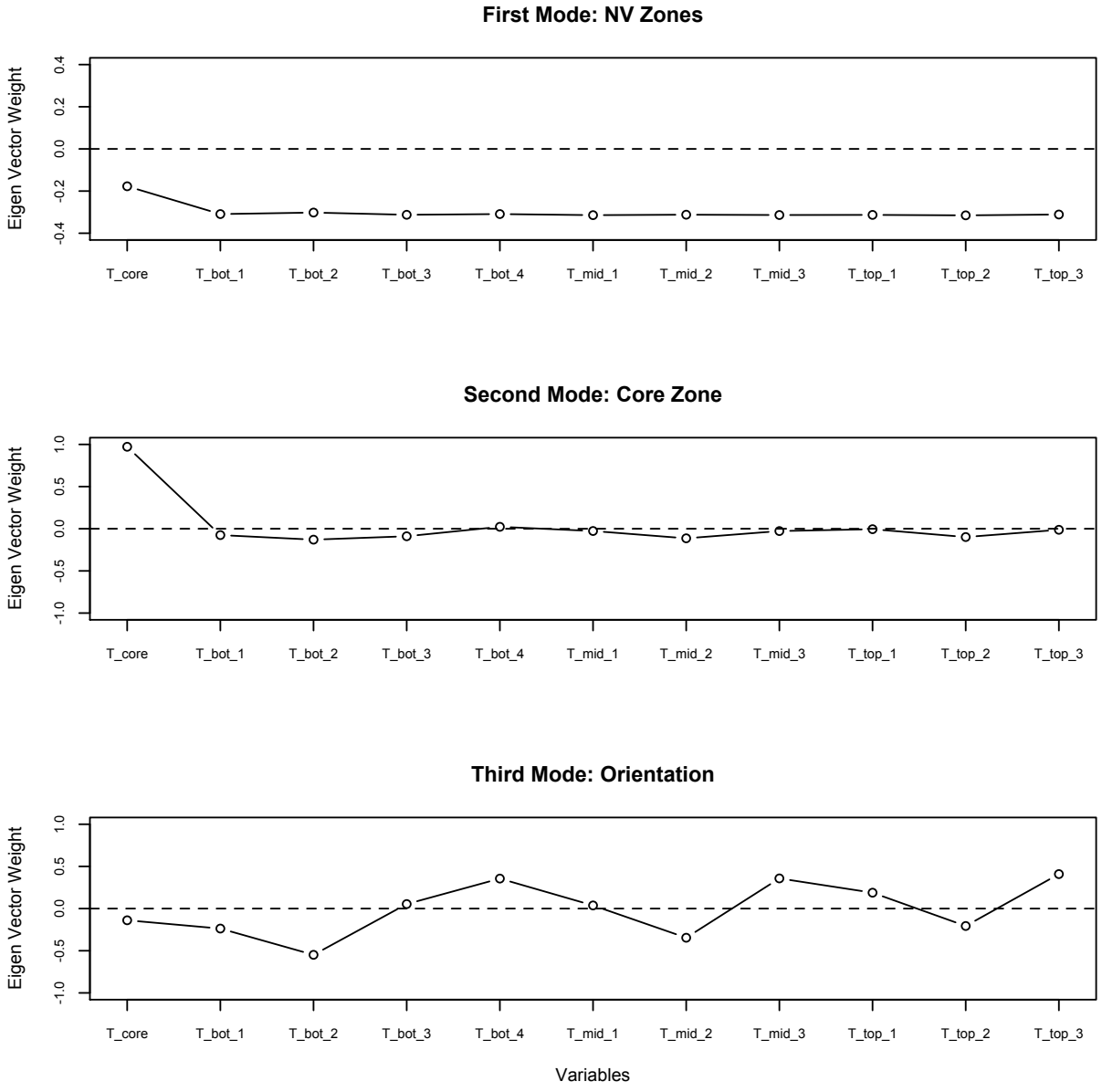


Figure 5.4: The eigenvector weights for the three leading modes in the PCA relate to three recognizable physical phenomena in the building. T_{core} represents the temperature of the core zone on the ground floor. All other temperatures are for perimeter zones.

A GLM can then be constructed from a subset of the original predictors (ambient conditions like dry bulb temperature and direct normal solar irradiance) and the three leading PCs for concurrent and lagged zone temperatures:

$$\hat{\theta} = f(\mathbf{P}_c, \mathbf{P}_l, \mathbf{X}^*). \tag{5.2}$$

$\mathbf{P}_c$ are the leading PCs of the concurrent variables, $\mathbf{P}_l$ those of the lagged variables. $\mathbf{X}^*$ represents the remainder of the original predictor set.

In open loop tests, a PCA-based GLM performed in a practically identical manner to GLM 2 itself, but was obtained directly without using a stepwise regression approach. RPSS and misclassification error were actually slightly less than GLM 2 (0.082 and 14.9%, respectively), although still within a few percent of the original values. The emphasis here is not on any performance improvement but rather the manner in which a parsimonious model can be directly obtained in situations where multicollinear zone temperature states would otherwise make selection of a "best" predictor set challenging. Results throughout the remainder of the paper will, however, focus on the performance of GLM 2, which was derived through the stepwise regression approach.

5.2.3 Closed Loop Performance

GLMs 2 and 3 were also tested for performance using the original building energy model. Closed loop tests clearly demonstrate the pitfalls of generalizing rule performance based on an open loop test, as there was an obvious degeneration in the skill of both models once embedded. Results from GLM 2 (Figure 5.2(b)) are representative, showing long periods of missed window openings (type 1|0 errors). In the case of GLMs 2 and 3, performance degraded significantly when the loop was closed. Misclassification errors compound and result in increasingly wider deviations in predictor variables, like zone temperatures, which in turn lead to further misclassification errors. Divergence of rules from optimal performance is discussed in greater detail in later sections of this chapter. One can observe that the dashed probability response of the model follows the diurnal night venting pattern well, but often does not result in openings because it does not exceed the

threshold value of 0.5. Lowering the threshold for openings could improve performance.

5.3 CART Rule Extraction

CARTs were trained on the same training and cross-validation datasets using the `rpart` package in **R**® [37], which implements the CART methodology of Breiman *et al.* [21]. Three different CARTs were investigated based on different predictor sets (Table 5.2). Each model was trained on seven weeks of data, then cross-validated on the remaining four weeks of optimizer results. As described previously, a minimum cost-complexity parameter was used to examine the goodness of fit for each tree candidate grown. The tree was back-pruned to a point of near-optimal cost-complexity using the "one standard error" rule.

5.3.1 Open Loop Performance

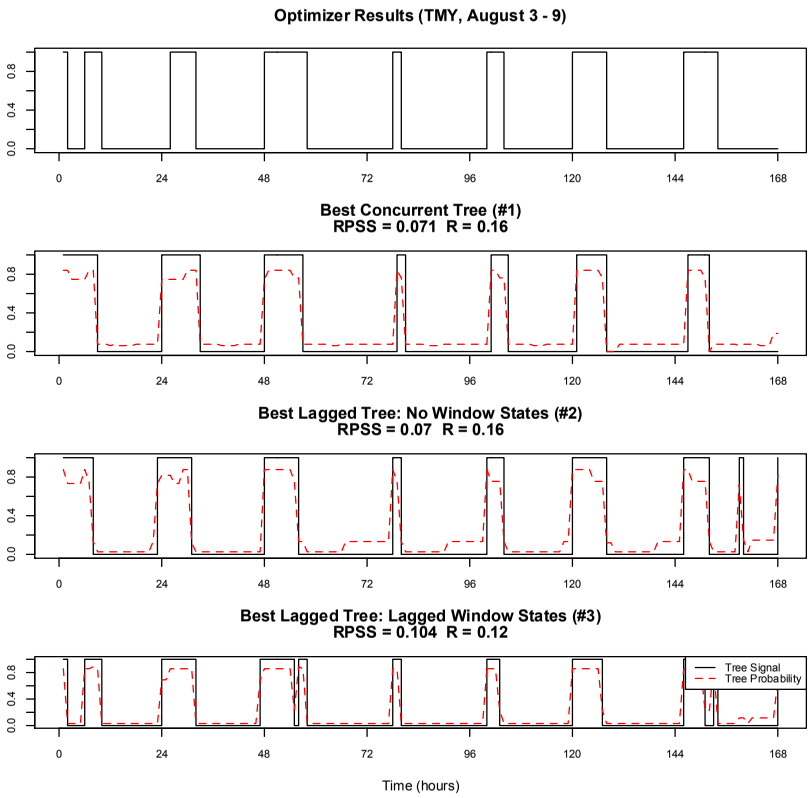
Open loop cross-validation predictions for the three CARTs are illustrated in Figure 5.5(a), with both the binary signal and the probability of a window being opened. The probability at any given terminal node is simply the proportion of points in the training set classified as a 1 for that node (i.e. the prior probability). Probabilities are only included to be able to generate RPSS values and for comparisons with GLMs.

The CARTs performed similarly to the three GLMs from the previous section, although RPSS values were uniformly lower, misclassification error rates uniformly higher. The third predictor set with the inclusion of the z_{t-1} lagged window state consistently yielded the best performance in open loop tests; however, the z_{t-1} term almost completely dominates the rule once it is allowed as a predictor. This caused undesirable oscillations in window state, so models with the z_{t-1} lagged window state were excluded from closed loop testing.

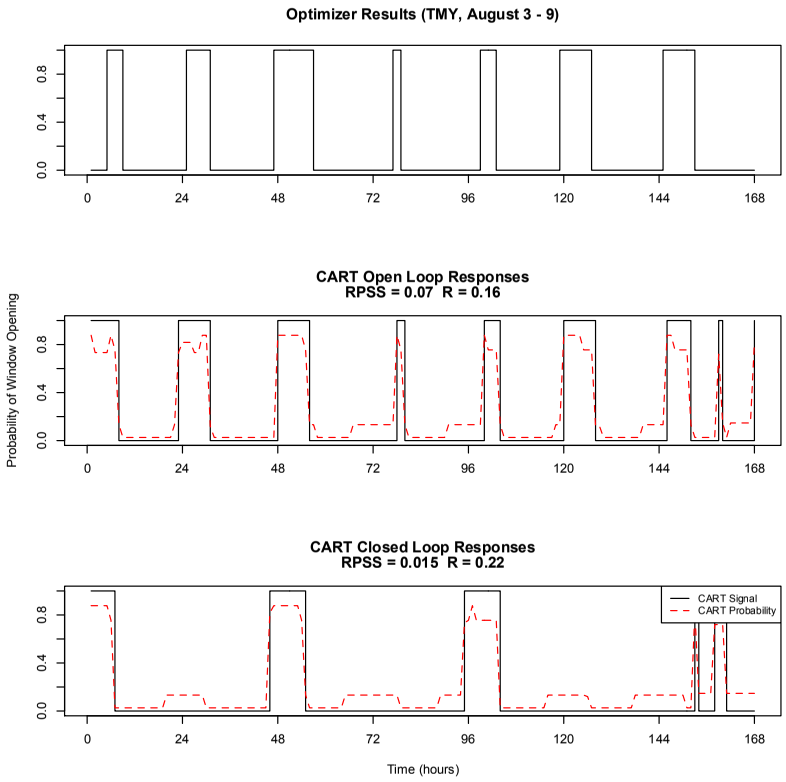
5.3.2 Closed Loop Performance

As was seen with GLM-based controllers, model performance can severely degrade in the closed loop test. CART 2 was embedded in the original energy model and simulated, and its

performance is illustrated in Figure 5.5(b) below. The CART did not miss as many window openings as GLM 2 and shows improved RPSS and misclassification error as a result.



(a) Open loop test comparison



(b) Closed loop test comparison

Figure 5.5: Closed loop sequences for CART-based controllers (left) were very similar to those derived from GLMs. RPSS values for the CART-based models were uniformly lower in open loop implementation. CARTs performed somewhat better in closed loop tests (right) than corresponding GLMs.

5.4 AdaBoost Rule Extraction

5.4.1 Consideration of Weak Learner Complexity

As with GLMs and CARTs, adaptive boosting models were trained on the same optimizer dataset (seven weeks), then cross-validated on the remaining four weeks of data. The boosting algorithm was capped at 50 iterations, meaning that the resulting boosting model would contain a chain of 50 weak learners. The particular **R** implementation of the AdaBoost algorithm used to develop these models (the `ada` package) allows boosting models to be developed from a variety of different weak learners, including simple decision stumps up to complete multi-level decision trees. Due to the binary/logistic nature of the model, we can take the probability as

$$p(\vec{x}) = \frac{e^{2H(\vec{x})}}{1 + e^{2H(\vec{x})}}, \tag{5.3}$$

allowing again for RPSS comparisons between the different models [42].

Several variations of weak learners were explored using the concurrent predictor dataset. The first used simple decision stumps (one split per learner), the second used two-layer CARTs (three splits per learner), and the third used "full" decision trees (trees are allowed to grow as long as each terminal node has seven or greater points). Clearly, a series of 50 CARTs, each potentially three or four layers deep, yields a boosting model of significant complexity and can even be impractical for implementation in a building simulation model; however, this model is shown here to illustrate performance differences. As shown in Figure 5.6, the boosting models vary somewhat in performance, although there is clearly not a linear relationship between the increase in performance and the complexity of the model. The third model achieves RPSS values comparable to those of the best GLM using only concurrent predictor information, but its RPSS is only about 20% better than the boost based on decision stumps and with orders of magnitude greater complexity than the two simple boosting models. The two-layer tree learner model strikes a reasonable balance between prediction skill and complexity and was chosen for all boosting investigations that follow.

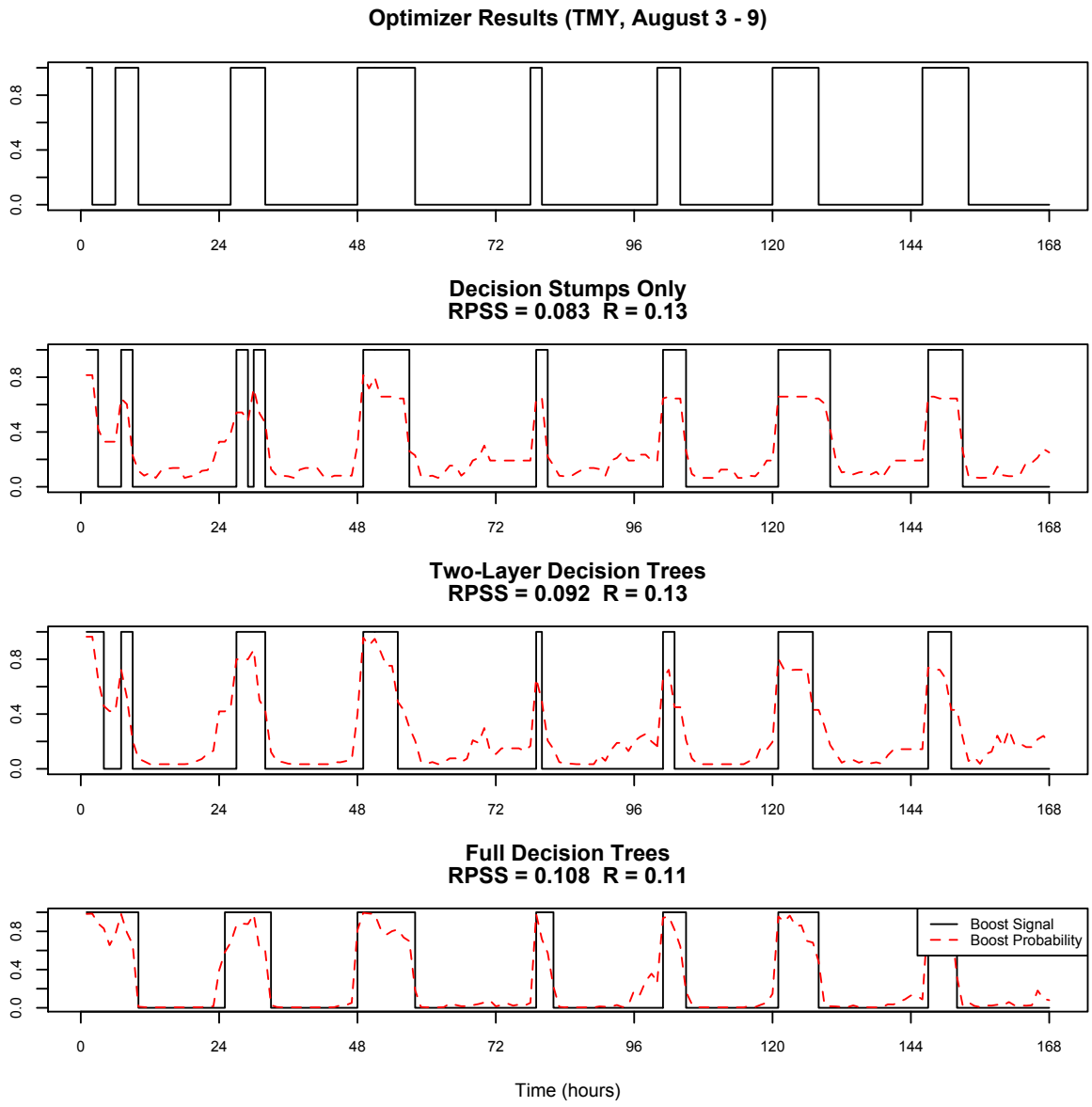


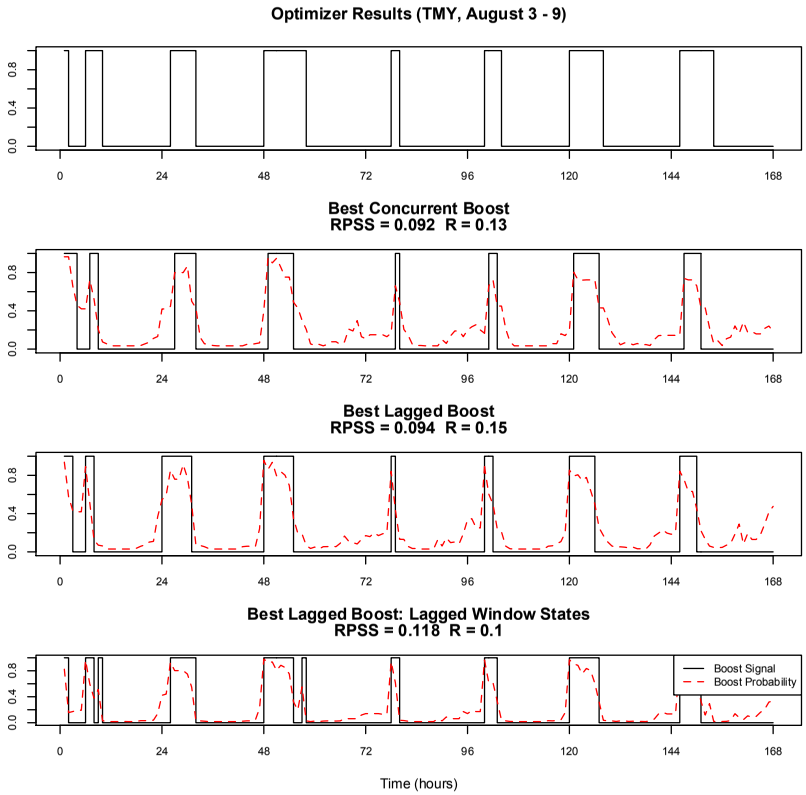
Figure 5.6: Boost models of varying complexity were compared for open loop performance. Increases in complexity beyond the two-layer tree learners yielded minimal performance benefits.

5.4.2 Open Loop Performance

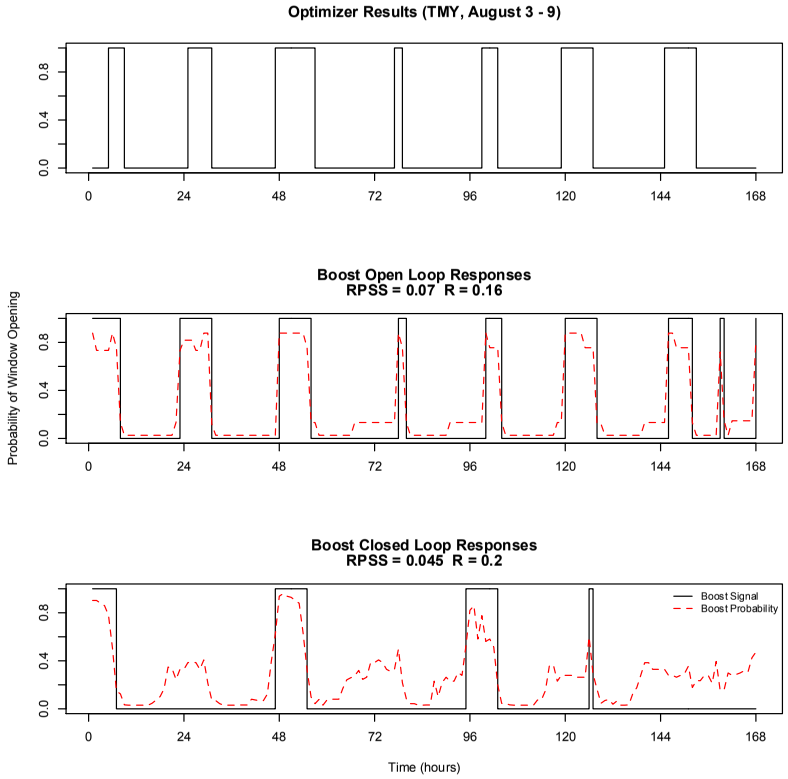
Two-layer boosts were examined using the various predictor datasets described in Table 5.2. Each model was first cross-validated in open loop fashion. Results are illustrated in Figure 5.7(a). Although RPSS values and misclassification errors are similar to the open loop performance of the GLM formulations, they possess the same advantage as CARTS in that they are easily convertible to if/then/else logic and, thus, can be likened to a rule-based controller.

5.4.3 Closed Loop Performance

As with all other models, closed loop performance tests were conducted on the boosting models. The closed loop performance of the boosting model outperformed both GLMs and CARTs, achieving best overall RPSS and misclassification error (Figure 5.7(b)).



(a) Open loop test comparison



(b) Closed loop test comparison

Figure 5.7: Open loop results for the two-layer boosting tree models developed on different predictor datasets (left) showed performance comparable to the GLMs and CARTs presented in Figures 5.2(a) and 5.5(a). Closed loop performance (right) was slightly better than other rule formulations.

5.5 Cross-Model Comparison of Energy Performance

Table 5.3 provides a summary of the energy savings achieved during the cross-validation/testing period with offline MPC and using the best-performing models from all three categories.[1] For the GLMs, the open loop performance of the PCA-based GLM is also presented for comparison with GLM 2. To demonstrate that the extracted rules achieve better-than-random performance, a version of the building model was run using a stream of random numbers to guide the opening and closing of windows.[2] A rule-based controller based on a natural ventilation heuristic was also included for comparison. The opening and closing of windows are controlled at the zone level with the following logic:

```
If (Tambient < Tsetpoint < Tzone) AND (12C < Tambient < 30C)
  OPEN WINDOWS
Else
  CLOSE WINDOWS
End
```

Openings most often occur for some portion of the night. The zone must require cooling according to the setpoint and the outdoor air must be able to satisfy some of the cooling load. The outdoor air must also fall within a range of 12 to 30 °C. The results presented for the heuristic case represent some manual tuning of the setpoint to encourage sufficient night ventilation.

In open loop tests, most of the models were able to maintain greater than 90% of optimizer savings, with total classification errors in the 15% range and RPSS values of 0.07 to 0.09. Note that the RPSS and misclassification errors tend to loosely follow the same trend as energy savings; however, even modest increases in misclassification can undermine energy savings and result in rules that underperform the base case, as with GLM 2 in its closed loop test. In closed loop operation, most models suffer some additional degradation in optimality. The CART and boost rules seem to

[1] Energy savings are only counted during the cross-validation period, not the entire simulation period.

[2] Random numbers were binomially distributed with a mean equal to the probability of open windows from the MPC solution (27%).

suffer the least from this phenomenon, whereas the GLM-based rules see more dramatic drawdowns, at least without manual tuning.

What is of greater practical importance is the fact that the statistical performance of the rule in an open loop test is not necessarily indicative of its closed loop performance. This is problematic because it means that the analyst is forced to embed the model and perform a closed loop simulation in order to obtain an accurate picture of energy performance. Closed loop testing can be time-consuming, but at present this appears to be the only practical means of estimating real-world control performance.

Table 5.3: Open and Closed Loop Energy Savings Comparisons for Cross-Validation Period

	HVAC Electricity			**Statistical Performance**			
	Use (kWh)	Savings (kWh)	Savings (%)	RPSS	R(T)	$R(0\|1)$	$R(1\|0)$
Base Case	4,663	-	-	-	-	-	-
Optimizer	3,989	674	-	-	-	-	-
NV Heuristic	4,144	519	77%	-	-	-	-
Bernoulli Random	5,501	-838	-124%	-	-	-	-
GLM 2							
Open Loop	4,050	613	91%	0.083	15%	8.5%	6.7%
Open Loop–PCA	4,056	607	90%	0.082	15%	7.6%	7.3%
Closed Loop	4,571	92	11%	0.021	24%	1.6%	21.9%
CART 2							
Open Loop	4,070	593	88%	0.070	16%	10.1%	5.7%
Closed Loop	4,094	569	84%	0.015	22%	10.1%	11.9%
Boost 2							
Open Loop	4,026	638	95%	0.091	15%	7.6%	7.8%
Closed Loop	4,033	630	93%	0.045	20%	9.7%	10.6%

A quick glance at Table 5.3 also at least empirically confirms what one might intuit about open loop versus closed loop tests: the open loop test represents the ceiling for a rule's performance, setting the maximum achievable energy savings for the closed loop test. Although we do not attempt to rigorously prove this point here from a theoretical standpoint, this observation stands to reason. Recall that the open loop test first takes the response of the model based on the predictor set established during optimization. The rule then forecasts what the optimizer would have chosen

under the exact same conditions. This is as close as the rule can come to approximating the original optimizer control sequence, which we assume to be a globally optimal sequence. Once the model is run in closed loop fashion, feedback occurs, errors compound, and the states of the building—mainly zone air temperatures—begin to diverge from the optimal. For each subsequent timestep, the rule finds itself at a greater disadvantage by starting from increasingly less optimal states, and performance falls further behind. The process of divergence is illustrated in Figure 5.8, in which the cumulative savings for CART 2 are plotted—both closed and open loop—alongside the optimizer's original solution. For the first few days, the extracted rule actually follows the optimizer sequence almost exactly, trending alongside the savings profile of the optimizer to within a few kWh. After this period, savings begin to diverge as error compounding occurs. Type 0|1 and 1|0 errors are noted on the chart, and error rates are displayed in the legend. Note that the closed loop solution closely tracks the open loop solution (asymptotically, for the first few days), but the cumulative savings of the open loop case are never exceeded by the closed loop case.

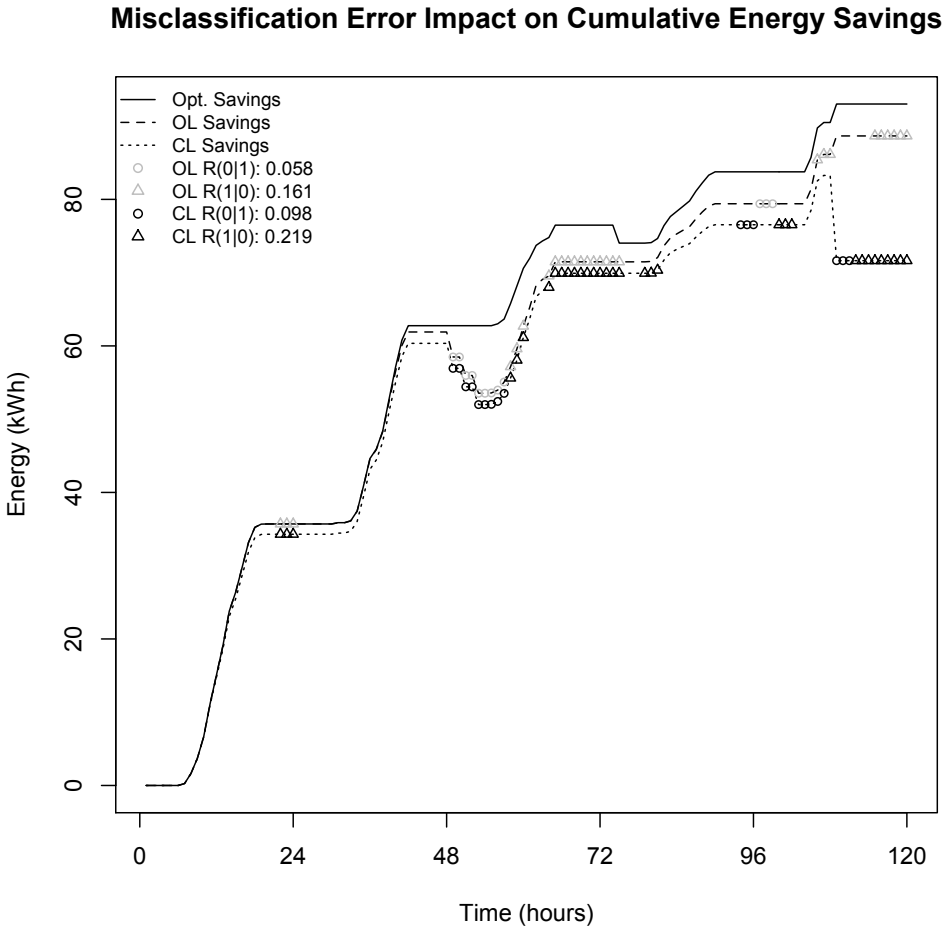


Figure 5.8: The cumulative savings of the open and closed loop solutions quickly diverge from the optimal, but remain relatively closely coupled to within a few percent.

5.6 Strengthening Interpretation and Performance Through Weighted Treatment of Error Types

As in medical diagnosis or forecasting credit defaults, certain types of error can carry greater weight than others. Note in Figure 5.8 that the first large departure from optimizer performance occurs at the beginning of the third day of operation during which a sequence of 0|1 type errors occur. Could this mean that maintaining closed windows at certain periods is more important than opening windows? Similarly, a further examination of Table 5.3 shows that the top-performing

adaptive boosting model consistently maintained lower 1|0 errors than any other model during closed loop tests. Perhaps this means that taking full advantage of natural ventilation at all times is the more important goal. In fact, it is difficult to pinpoint exactly which type of error is more important in these kinds of tests because of time lag effects. The consequences of one's decisions may not be felt until a later time at which point, for example, a compressor might turn on prematurely. It is practically impossible to analyze the amount of energy savings foregone as a result of individual misclassification types; only the cumulative effect of all misclassification can be easily known.

What one can easily observe for the presented case is that type 1|0 classifications seem to increase disproportionately compared to 0|1 errors when each of the rules is placed in a closed loop test. Once embedded the rules are more likely to close windows that should otherwise be open. In particular, the CART and boost rule seem to achieve better performance based on their ability to maintain longer periods of open windows compared to the GLMs. This suggests that 1|0 errors should be treated preferentially to preserve congruence with the cooling season case evaluated by MPC.

We can use this information to our advantage by preferentially treating different types of errors in the rule formulation process. Recall that CART fitting for categorical forecasts is guided by minimizing a cost-complexity metric, R_α, which includes the total misclassification error for the tree, $R(\mathrm{T})$. The misclassification error can be broken down into two terms, $R(0|1)$ and $R(1|0)$. Each of these terms can then be multiplied by a different loss term, l, placing greater or less emphasis on certain types of errors.

In the case of the rules in this paper, greater penalty must be placed on 1|0 errors, so the $l(1|0)$ coefficient should be increased over the $l(0|1)$ coefficient. Decreasing the $l(0|1)$:$l(1|0)$ ratio in CART 2 from 1:1 had modest performance impacts, increasing the optimal savings recovered up to 94% in open loop operation and 89% in closed loop (Table 5.4), even though overall error rates remained about the same.[3] Note that 1|0 errors still rose dramatically in the closed loop test, and

[3] Ultimately a $l(0|1)$:$l(1|0)$ ratio of 1:1.75 was arrived at after a series of manual trials. This does not represent an "optimal" value, nor would this type of weighting apply to all cases. We merely use an asymmetrical weighting to demonstrate that small performance improvements can be achieved.

even the weighted CART 2 could not outdo the performance of the seemingly more robust boost model.

Table 5.4: Open and Closed Loop Energy Performance of CART with Asymmetric Loss Coefficients

	HVAC Electricity			**Statistical Performance**			
	Use (kWh)	Savings (kWh)	Savings (%)	RPSS	R(T)	R(0|1)	R(1|0)
CART 2							
Open Loop	4,070	593	88%	0.070	16%	10.1%	5.7%
Closed Loop	4,094	569	84%	0.015	22%	10.1%	11.9%
Weighted CART 2							
Open Loop	4,032	632	94%	0.065	17%	11.6%	5.2%
Closed Loop	4,064	600	89%	0.025	21%	11.2%	10.3%
Boost 2							
Open Loop	4,026	638	95%	0.091	15%	7.6%	7.8%
Closed Loop	4,033	630	93%	0.045	20%	9.7%	10.6%

5.7 Cross-Model Comparison of Rule Structure

One of the key assumptions of the rule extraction exercise is that the rules developed unveil some of the underlying structure in the data. In the case of rule extraction for offline MPC, we aim to gain a better understanding of the logic that may have been used by the optimizer to guide building operation so that we can replicate it. Since in this paper we have used three different styles of models for rule extraction, it is interesting to examine their structure for similarities. We can examine the choice of variables for each model and the relative "strength" or importance of the variable in the model.

Each type of model possesses a different method for assessing variable importance or significance. In the case of the GLM, traditional hypothesis testing can be used to evaluate the significance of individual variable coefficients chosen. In the case of binomial variables, a χ^2 distribution test is used to assess the probability of rejecting the null hypothesis (p). We can then take the complementary value, α, (i.e. $1-p$) to be the relative strength of a variable in the model.

In CART models, variable "importance" is assessed by summing the number of times a variable was evaluated as a potential split in the tree; splits considered during the iterative tree growing process but not implemented in the final tree are considered.[4] Similarly, variable importance can be assessed in boosting models by summing the number of times that a variable appears in the weak learners that comprise the boost. In the boosting models used in this paper, this would be the equivalent of summing the number of tree nodes containing a split with a certain variable.

If we take each of these significance/importance measures and normalize them such that the sum across all variables is 1, values can be plotted across the parameter set to assess commonalities between the rules (Figure 5.9). It should be noted that the absolute values of the bars are not important, because the figure is intended to provide a qualitative impression of model structure. For this case, we have examined the structure of rules based on the second parameter set, which included concurrent and lagged predictors. A close resemblance between the relative importance profiles of the CART and boosting rules is readily apparent. These two models place the greatest importance on concurrent zone temperatures, even emphasizing some of the same specific zones. Similar commonalities between the CART and boost can be seen with regards to other variable classes as well (e.g. lagged zone temperatures).

The GLM—which lagged both in stability and performance compared to the CART and boosting rules—only vaguely aligned with the other models in its variable significance profile. It placed almost no significance on concurrent ambient conditions, and placed equal weighting on many of the other variables used. Some of the discrepancy between models might be explained by the stepwise model fitting process and issues that this method can encounter in trying to distinguish between highly multicollinear predictors.

One could attribute at least some of the similarities between the last two rules to the fact that the fundamental component of each is a CART, and thus both utilize the some of the same tree growing principles in selecting variables. However, boosting models can exhibit dramatically

[4] In this case, both competing and surrogate splits were included in the importance measure. See [21] for further clarification on CART importance measures.

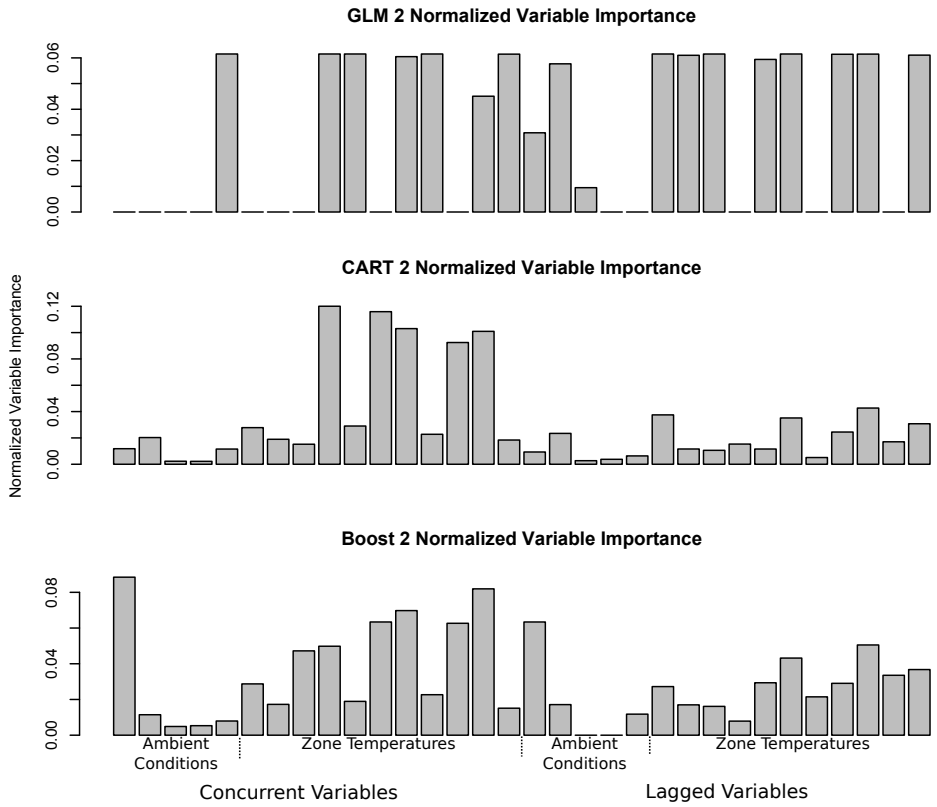


Figure 5.9: Variable importance/significance measures are presented for each style of rule, with values normalized to allow for comparison. The CART and boost rules show consistent agreement across the predictor set, with somewhat less agreement with the GLM.

different behavior compared to simpler tree-based classifiers when trained on the same dataset [10], so one could assume that some similarity observed is due to common underlying logic, which is discussed below.

For ease of interpretation, we can examine the dendrogram of CART 2 (Figure 5.10) and assess the logic employed. All of the variables employed in the splits are zone temperatures, with the exception of the dry bulb and dew point temperatures (T_{db} and T_{dp}, respectively). The subscripts "bot", "mid", and "top" refer to the three floors of the building, whereas the number refers to the zone's position on the floor. A "-1" in the subscript denotes that the variable is lagged. When a statement is evaluated as true, evaluation proceeds down the left branch. The classification

at the terminal nodes is binary, with 0 for closed windows and 1 for open. As the rule is dominated by zone temperatures, this CART has mostly "learned" how to map zone temperature conditions to window states based on the results of the offline MPC problem, achieving the best possible classification without the inclusion of forecast information available to the optimizer. Even in this simplistic formulation, the rule outperformed the natural ventilation heuristic by a reasonable margin in closed loop cross-validation tests. Care should be taken in generalizing rules extracted in this manner, as the following section will address.

5.8 Robustness

As with other inverse modeling techniques, care must be taken to ensure that the rule is trained on a dataset broad enough to encompass typical operating conditions. The models presented, for example, were only trained on MPC solutions for a cooling season case. When CART 2 is used to control the test building during the swing or heating season, it not only underperforms the natural ventilation heuristic but the base case itself. In the month of October, the base case HVAC system consumes about 1,913 kWh of electricity. The night venting heuristic presented in earlier sections provides relatively minor savings (less than 1%), but the extracted rule actually increases electric consumption by over 9%. There is no guarantee that the rule will function outside the conditions of the original MPC case. Heuristics extracted in this manner appear to be only as robust as the data used to train them.

We propose two different approaches to improve the robustness of extracted rules. The first and simplest approach is to develop multiple rules for different sets of boundary conditions. Offline MPC runs could be conducted parametrically for occupied and unoccupied conditions, swing and cooling seasons, and so on. Separate rules could then be developed for each case and applied as appropriate by the building automation system. For example, rule A might apply during weekdays during the cooling season, and rule B would apply to weekdays during the swing season. One would naturally need to analyze how finely to parse the boundary conditions by determining where patterns in the MPC solution differ.

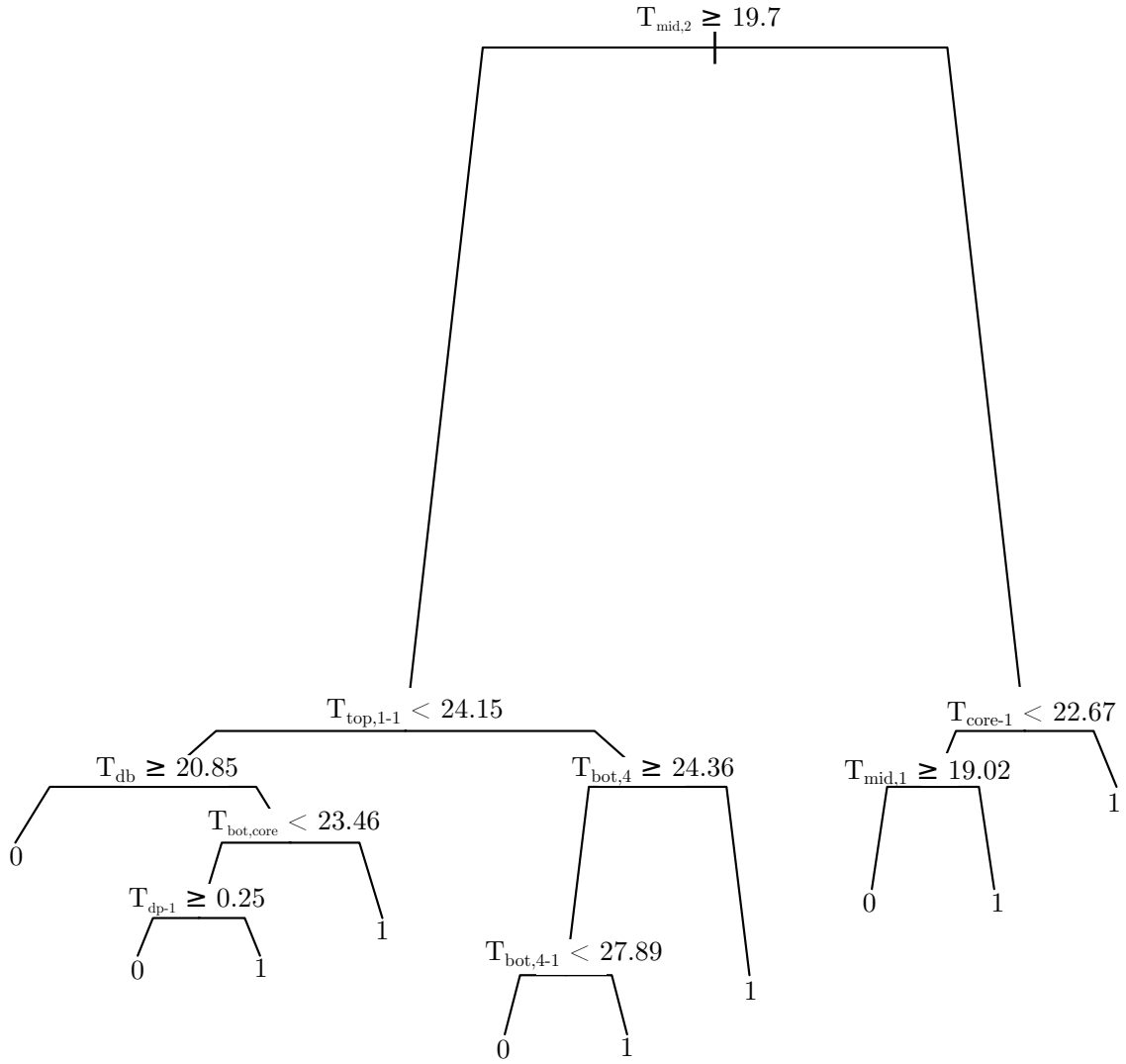


Figure 5.10: CARTs benefit from their ease of interpretability because their logic can easily be depicted in tree form. CART 2 uses zone temperatures almost exclusively, with the exception of two splits on dry bulb and dew point temperatures (T_{db} and T_{dp}, respectively).

A more rigorous but computationally burdensome approach would be to base rules on the results of stochastic MPC, optimizing control decisions in the presence of uncertain weather and occupant behavior. Stochastic MPC could expose the optimizer to an ensemble of different boundary conditions, yielding a corresponding ensemble of MPC solutions and associated model states. The training set for the rule could either be based on the entire ensemble of MPC solutions or could be drawn from some subset of the entire distribution (e.g. all solutions falling within the one standard deviation of the expected value).

5.9 Conclusions

Three rule extraction techniques have been examined and compared for performance, including GLMs, CARTs, and adaptive boosting. A combination of open and closed loop tests were used to assess the statistical skill and energy performance of the models during a cross-validation period. Although all rules were able to recover approximately 90% of the original optimizer energy savings under open loop tests, only the CART and boosting rules were able to maintain reasonable performance during closed loop testing. The GLM-based rules saw significant performance degradation during closed loop tests; CART and boosting rules only degraded in performance by a few percentage points, still retaining the vast majority of optimizer savings (84% and 93% for the CART and boost rules, respectively). If GLM rules were to be pursued in the future, a supplemental PCA technique has been demonstrated that can easily help filter out multicollinearity from predictor variable sets and yield parsimonious, right-sized models. Tests indicate that open loop tests can provide a quick glimpse of best-case rule performance, but that these results can differ—in some cases drastically—from closed loop tests that more accurately reflect real-world operation.

A more detailed examination of misclassification error types indicated that type 1|0 errors seemed to play a larger role in overall model energy performance for the presented case. That is to say, it was more important to harvest savings from appropriately timed window openings than to potentially add loads by leaving windows open at inappropriate times. By disproportionately penalizing 1|0 errors in the CART formulation process, we were able to recapture a few percent

more of the optimizer energy savings, demonstrating a process by which to further strengthen rule performance.

Even with the aforementioned modifications to CART 2, the boosting rule was still the top performer from an energy perspective; however, the boosting rule involved a combination of 50 weak learners (each of them a two-layer CART itself), resulting in over 800 lines of EnergyPlus Runtime Language code, all of which would need to be translated into a BAS script to control a real building. Translation and implementation of the code are not a problem, but comprehensibility and acceptability by facility management staff are real concerns. In the end, real human beings are responsible for the control of a building, not a BAS, and those individuals must understand and trust the control algorithms at work inside their systems. An 800-line boosting model may as well be a black box. A 50-line CART whose code and graphical decision tree can all be printed on a single piece of paper has great practical advantages. Clearly, a rule's degree of optimality should not be the only consideration in choosing a best performer.

At the core of this research is the idea that the original offline MPC optimization yielded some underlying truth about the operation of this particular building that could be mined from the results. If universal rules were truly present, one should observe some similarity in rule structure across different extraction techniques. Indeed, our CART and boosting rules showed remarkable similarity in their overall structure, suggesting that, at least for this simple case, the rule extraction process had effectively learned some of the relationships mapping building states and weather conditions to optimal decisions. In later rule extraction cases presented in Chapters 7 and 8, we include simple forecast information (e.g. day-ahead max/min outdoor temperature forecasts) and more historical state information (e.g. previous-day max/min zone temperatures or window states) to improve performance. This provides the rule with a more "omniscient" view of the building's trajectory (closer to what the optimizer sees), while still formulating the problem in terms of values that are readily available in a real world setting.

The results indicate that rule extraction is a promising technique for developing and fine-tuning supervisory operation strategies in buildings. It could allow a conventional BAS to replicate

optimal operational strategies without the need for an online MPC system. Additional cases are presented in Chapters 7 and 8 to extend this concept to more realistic and complex cases, culminating in a physical test.

Chapter 6

Offline MPC Simulation Study

Through the preceding chapters, an offline MPC scheme has been demonstrated that can handle the form of MINLP problems presented by supervisory control optimization in MM buildings. A variety of rule extraction techniques have also been applied to simplified MPC cases to provide proof of concept and examine the level of performance one might expect when extracting heuristics from MPC results. This chapter examines the question of optimal supervisory control in MM buildings more broadly, through a simulation study across several MM building types and a number of climates. As described in Chapter 3, a detailed sensitivity analysis or parametric simulation study (e.g. examining impacts of internal gains, window placement/size, or weather forecast uncertainty on offline MPC solutions) was simply not practicable due to the very long runtimes associated with the chosen offline MPC framework. Prior research has examined such sensitivities in great detail (e.g. [49]). A coarser parametric study was instead pursued, with priority placed on examining solution differences between different "typical" MM building types, multiple objective functions, and a variety of climate conditions. These three macro-scale factors have far greater impact on solutions and provide the appropriate level of insight for this research, which is the first to examine optimal control in typical MM buildings. This research will instead help illuminate where detailed sensitivity analyses are needed, which can then be conducted in future work.

The primary goal of this phase of the research was to provide a benchmark for supervisory control schemes in a variety of MM building types. We then have a "yard stick" against which to measure existing supervisory control heuristics and determine their degree of sub-optimality.

In certain building types and climates—often in the more advanced reference case buildings where systems are already somewhat optimized and more sophisticated control schemes are used—existing supervisory heuristics provide energy savings and comfort that are very close to our solutions. In many of the simpler buildings like MM1 and 2, MPC capitalizes on thermal storage and adaptive comfort opportunities that simple heuristics cannot reproduce. The simulation study also provided several inspirations for very simple supervisory control improvements that should enable simple heuristics to achieve better energy and comfort performance.

As is noted throughout this chapter, a collection of results for relevant cases from the offline study (Boulder results only) are provided in Appendix D. It was not practical to include many of the results inline, but the appendix may help illuminate solution nuances discussed in this chapter.

6.1 Methodology

The overarching methodology behind the design of the simulation study is provided in Chapter 3. Additional details on model development, specific MPC problem formulations, and selection of parametric runs is provided below.

6.1.1 Model Development

Models for four prototypical MM buildings (MM1 through MM4) were developed in EnergyPlus using the mix of features presented in Table 3.1. The "seed" for the models was the medium office building model from DOE's reference building models [36], with a narrower floor plate (60 ft) typical of naturally ventilated buildings [24, 23, 72]. Occupant densities, internal gains (both equipment and lighting), and equipment operational schedules were maintained from the reference models. Zone cooling setpoints were maintained at 24 °C during occupied periods with a 30 °C setup; heating temperatures were 21 °C with a 16 °C setback. Due to the interest in finding optimal strategies under occupied conditions, where comfort boundaries must be observed, daily occupancy of 6am to 6pm was assumed. Weekends and holidays were ignored as the study was mainly concerned with daily profiles during occupied periods, during which comfort must be

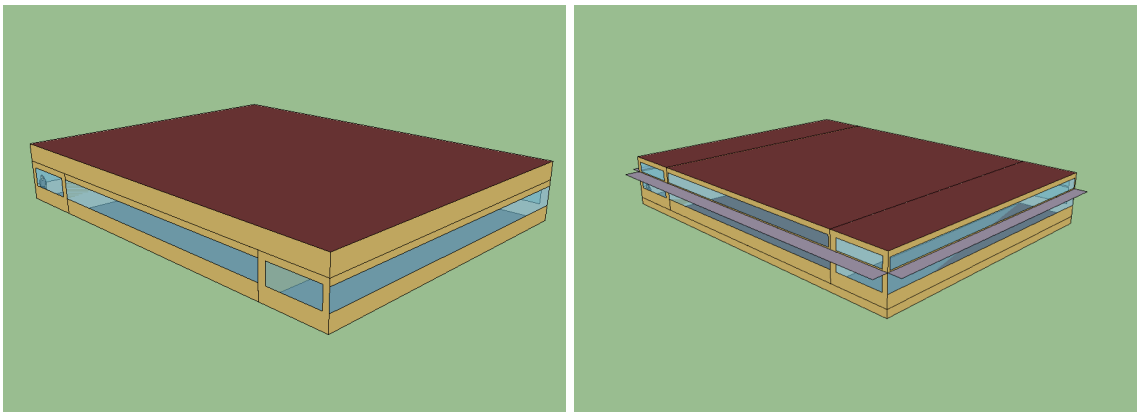
preserved.

To reduce model runtimes, only the three zones of the second floor of the building were modeled (adiabatic boundary conditions were placed on the top and bottom of the collective zones). EnergyPlus' `AirflowNetwork` nodal airflow model was employed to simulate ventilation flows in the three zones. An array of wind pressure coefficients aligned on center with the building's operable windows was developed using a database of CFD-derived wind pressure coefficients, provided through a web tool by the Netherlands Organization for Applied Scientific Research (TNO) [93]. Occupant window openings were governed by the modified Humphreys behavioral model, as described in Chapter 3.

The MM1 model utilizes ASHRAE 90.1-2007 standard steel-framed exterior wall constructions and windows. It employs an all-air HVAC system with a rooftop AHU, DX cooling, and VAV terminal devices (hot water reheat). In MM2, wall constructions were modified to conform with the mass wall constructions listed in 90.1. Exterior shading devices with an approximately 0.9m projection were added on all but the north-facing façades to reduce solar gains. HVAC systems were resized accordingly.

MM3 represents a fundamental change in systems in several respects. First, the building now supplies ventilation and cooling air through an underfloor air distribution (UFAD) system, delivered through a 46cm underfloor plenum. The stratified University of California San Diego UFAD room air model is now used to determine temperature in the occupied and unoccupied regions of the zone, with an assumed transition height of 1.7m. The airside systems are sized per guidelines published in the ASHRAE Underfloor Air Distribution Design Guide [11]. Primary cooling is still provided through a packaged rooftop AHU with DX coils. Since UFAD supply air temperatures are typically much closer to room neutral conditions, heating is provided through perimeter hydronic baseboard convector/radiators fed by a gas boiler. MM3 also incorporates an **interlock** between manually operated windows and the HVAC system, meaning mechanical ventilation and cooling is disabled when occupants open windows. This feature is accomplished in practice through window contact sensors.

MM3's envelope now also incorporates automated night ventilation windows directly above the vision glazing and shading devices (see Figure 6.1), a feature common in more advanced MM facilities. Under standard heuristics (called the reference case), the night ventilation windows are allowed to open anytime during the unoccupied period to satisfy a zone setpoint of 22 °C, pre-cooling zones to the lower end of the ASHRAE 55 summer comfort region.



(a) MM1 & 2: Operable Vision Glazing

(b) MM3 & 4: Automated Night Ventilation Transoms and Operable Vision Glazing

Figure 6.1: Two glazing schemes for operable windows in MM building models.

Finally, in MM4 the cooling concept is completely overhauled. All heating and cooling is provided through radiant ceilings via concrete core conditioning or thermo-active building systems (TABS). When in cooling mode (the thrust of our investigations), the chilled ceiling is pulsed or charged throughout the unoccupied hours, with control systems attempting to achieve a zone temperature setpoint of 22 °C. The core is then pre-cooled and allowed to float through the occupied period. Supply water temperatures are controlled according to the reset schedule of Olesen, which adjusts supply water temperature as a function of outdoor temperatures [78, 77]:

$$T_{chw,s} = 0.35(18 - T_{oa}) - 18, \tag{6.1}$$

where T_{oa} is the outside dry bulb temperature in degrees Celsius. Ventilation is still provided through a UFAD system, but since zone sensible loads are met by the radiant systems, only hygienic ventilation is required. A dedicated outdoor air system, or DOAS, provides ventilation rates per

ASHRAE 62.1-2007. Automated exterior solar shades are enabled on the vision glazing, activating at 200 W/m^2 of beam insolation incident on the window surface, dramatically cutting solar gains.

The central plant is now a ground source heat pump (GSHP) with the ability to operate in "free cooling" mode (i.e. bypassing the compressor to chill water based purely on natural heat removal from the ground loop boreholes). Radiant systems are commonly paired with GSHPs in practice because of high supply chilled water temperatures, low supply hot water temperatures, and generally low design ΔT requirements. It was impractical to incorporate detailed models for the ground loop system, mainly due to the significant increase in simulation time that results. For example, EnergyPlus provides the option to model the condenser side of the loop, the ground borehole heat exchanger field, with a dynamic ground heat transfer coupling method incorporating g-functions, but past experience with these models suggested that they could more than double runtimes.

A simplified approach was implemented via an `EnergyManagementSystem` program that provides a coarser estimate of GSHP energy consumption and supplied cooling/heating, as illustrated in Figure 6.2, in which the entire GSHP plant system is substituted with district heating/cooling objects and assumed COPs. The purpose of this simplification is to decrease runtimes by ignoring some of the longer timescale interactions between the condenser and the deep ground. Water heating and cooling coils for the air handler are fed by a common GSHP, radiant heating and cooling loops by a second GSHP. Rather than varying COP based on performance curves, constant COPs of 2.5 and 3.5 are assumed for heating and cooling modes, respectively. These are relatively conservative values. The GSHP coupled to the radiant ceilings has the option to bypass compressor operation when ground temperatures are sufficiently below the chilled water supply setpoint. Deep ground temperatures are established based on monthly ground temperatures from the weather file at a depth of 5m.

A quantitative summary of model loads and basic constructions are provided in Table 6.1.

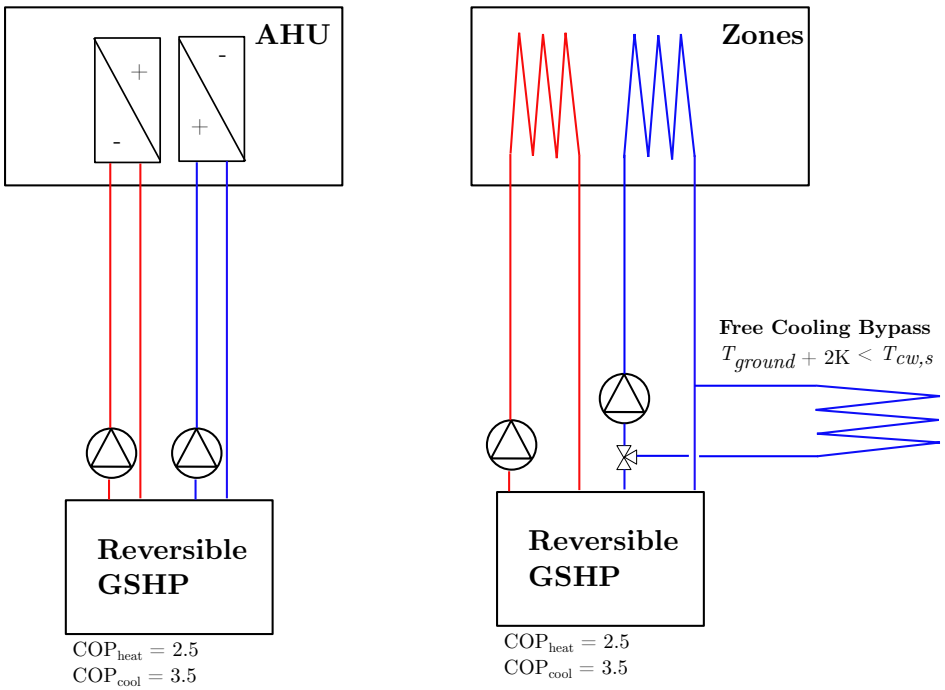


Figure 6.2: Schematic of the simplified modeling approach taken for GSHP with free cooling.

Table 6.1: Summary of Model Design Values and Loads

	MM1	MM2	MM3	MM4
Shell Properties				
Opaque Wall U-Value (W/m^2-K)	0.48	0.68	0.58	0.58
Window U-Value (W/m^2-K)	3.2	3.2	1.3	1.3
WWR	0.24	0.24	0.36	0.36
Loads				
Occupant Density (m^2/person)	18.6	18.6	18.6	18.6
Electrical Power Density (W/m^2)	8	8	8	8
Lighting Power Density (W/m^2)	10	10	10	10
Total Zone Design Loads (W/m^2)	65.3	61.8	71.5	36.8
System Parameters				
OA Design Flow (L/h)	392	392	211	211
Cooling Design Airflow (L/h)	2400	2200	3800	211
Supply Air Temperature (°C)	13	13	18	18

6.1.2 MPC Problem Formulations

Each building represented a slightly different MPC problem formulation (see Table 3.2 for an outline of the decision variables considered). All MPC problems are similar to the general case presented in equation 3.1 in that they are MINLP, with some nuances. In MM1 and MM2, the optimizer is examining strategies to improve upon the manual operation of windows by manipulating window positions and zone temperature setpoints. The inclusion of zone temperature setpoints provides the HVAC system with the ability to "trim" temperatures as needed to fulfill comfort requirements, while guaranteeing minimum hygienic ventilation values. This can be considered a hybrid between concurrent and changeover operation: the cooling function of the HVAC system can effectively be disabled by setting up setpoints, but the ventilation function cannot, due to minimum damper positions on the VAV terminal boxes. As such, this strategy will be referred to as **partial changeover** to distinguish it from a complete changeover strategy in which all mechanical ventilation ceases when natural ventilation is used.

Since MM1 and MM2 only contain occupant-operated windows and the investigation is to benchmark those decisions against optimal control, optimizer decisions on window openings are restricted to the occupied period. The actions of the optimizer can be viewed as a potential red/green light signal that could be used to inform occupants of desirable natural ventilation periods. Setpoints, on the other hand, can be manipulated by the optimizer at all times of day, allowing for nighttime pre-cooling, for example. They are constrained between values of 16 and 30 during unoccupied periods, and 21 and 30 during occupied periods in increments of 2K (the lower end of these constraints prevents cooling setpoints lower than prevailing heating setpoints). Decisions for both variables are made in 12 two-hour blocks over a 24-hour planning horizon ($P = 24$) and implemented on a 24-hour execution horizon ($E = 24$); costs are evaluated over a 72-hour cost horizon ($C = 72$); and thermal states are initialized over a two-week initialization horizon ($I = 336$). The problem is mathematically formulated as

$$
\begin{aligned}
\text{Minimize } C_{tot} &= f(\vec{x}, \vec{y}) = C_e + C_c \\
\text{Subject to} &: \\
&\vec{x} \in \{0,1\}^{12} \\
&\vec{x}_{t \notin t_{occ}} = 0 \\
&\vec{y} \in \mathbb{R}^{12} \\
&16\,^{\circ}\text{C} \leq \vec{y}_{t \notin t_{occ}} \leq 30\,^{\circ}\text{C} \\
&21\,^{\circ}\text{C} \leq \vec{y}_{t \in t_{occ}} \leq 30\,^{\circ}\text{C},
\end{aligned}
\tag{6.2}
$$

where $\vec{x}$ is the window position, $\vec{y}$ is the zone temperature setpoint, and t_{occ} are the occupied periods of the planning horizon.

In MM3 automated windows are added to the building model. Whereas the optimizer manipulates manual windows only during occupied periods to emulate optimal occupant behavior, it manipulates automated windows during unoccupied periods to achieve optimal pre-cooling. Another binary decision vector, $\vec{z}$, is therefore included in the problem formulation:

$$
\begin{aligned}
\text{Minimize } C_{tot} &= f(\vec{x}, \vec{y}, \vec{z}) = C_e + C_c \\
\text{Subject to} &: \\
& \vec{x} \in \{0,1\}^{12} \\
& \vec{x}_{t \notin t_{occ}} = 0 \\
& \vec{y} \in \mathbb{R}^{12} \\
& 16\,^{\circ}\text{C} \leq \vec{y}_{t \notin t_{occ}} \leq 30\,^{\circ}\text{C} \\
& 21\,^{\circ}\text{C} \leq \vec{y}_{t \in t_{occ}} \leq 30\,^{\circ}\text{C} \\
& \vec{z} \in \{0,1\}^{12} \\
& \vec{z}_{t \in t_{occ}} = 0
\end{aligned}
\tag{6.3}
$$

Finally, in the case of MM4, decisions are no longer made on zone temperature setpoints since a DOAS is in use. Instead, the schedule of circulation pump for the radiant ceilings is controlled in pulse-width modulation (PWM) fashion, specifying the fraction of the hour that systems should be allowed to operate. Thus $\vec{y}$ becomes a PWM signal in units of minutes/hour. Natural ventilation decisions remain unchanged:

$$
\begin{aligned}
\text{Minimize } C_{tot} &= f(\vec{x}, \vec{y}, \vec{z}) = C_e + C_c \\
\text{Subject to} &: \\
& \vec{x} \in \{0,1\}^{12} \\
& \vec{x}_{t \notin t_{occ}} = 0 \\
& \vec{y} \in \mathbb{R}^{12} \\
& 0 \text{ min/hr} \leq \vec{y} \leq 60 \text{ min/hr} \\
& \vec{z} \in \{0,1\}^{12} \\
& \vec{z}_{t \in t_{occ}} = 0
\end{aligned}
\tag{6.4}
$$

6.1.3 Parametric Investigations

A set of initial investigations was conducted on Boulder-based building models to assess the impacts of different objective functions and comfort interpretations. Three of these included comfort penalties—ASHRAE 55 "static", ASHRAE 55 adaptive, and EN 15251 adaptive—and the fourth included energy only (no comfort penalty). Each of these four cases was evaluated on "swing season" weather (the mid-May to mid-June) and cooling season weather (mid-July to mid-August), for a total of eight cases per building.

To supplement the existing runs and benchmark full changeover operation—recall that only partial changeover was allowed up to this point—a second suite of optimizations was conducted on building models that employ interlock between manually operable windows and the VAV terminal boxes. When windows are opened, VAV dampers are forced shut, and the system fan powers down.

After an initial extensive investigation in the Boulder climate, it was possible to take the most promising problem formulations (e.g. objective functions offering the best combination of

savings and comfort) and examine them under other climate conditions, namely San Francisco, CA (3C), Las Vegas, NV (3B), Seattle, WA (4C), and Baltimore, MD (4A). Climate investigations were not conducted on MM4 due to the extensive effort required to properly size the systems and the significantly higher runtimes associated with optimizations.

6.2 Results: MM1 and MM2

Results for MM1 and MM2 are presented simultaneously due to the high degree of model similarity and identical MINLP problem formulations for these cases. Each case was optimized preliminarily for a period of two weeks. The first week of results is considered a "warmup" period during which the optimal solution pattern begins to establish itself. It is ignored in our analysis.

All MPC results are compared to two different types of default buildings. The base case model reflects operation of the building in the absence of natural ventilation. The reference case model depicts a typical MM configuration (e.g. manual operation of windows by occupants).

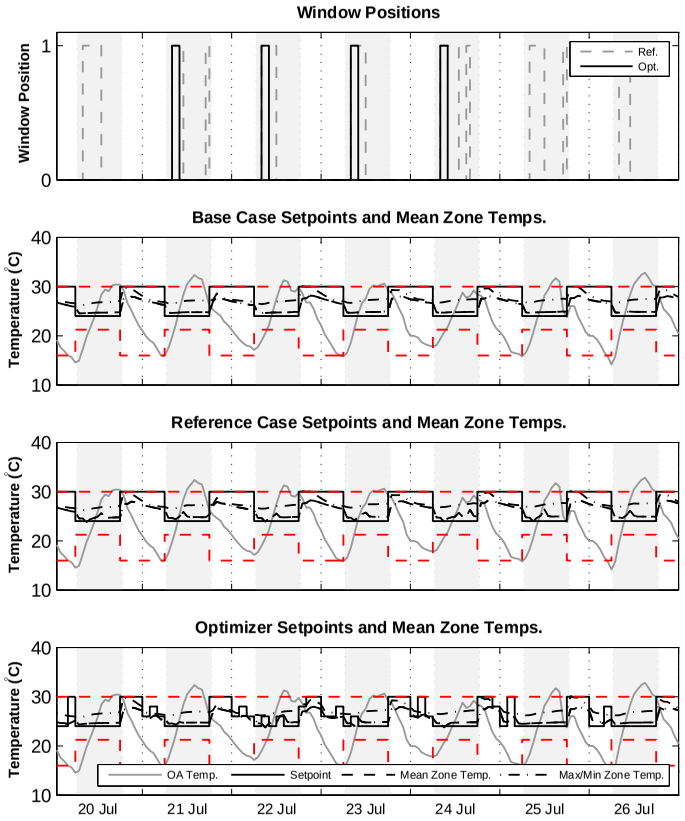
6.2.1 Representative Results for Boulder, CO Climate

Although they are the simplest models in this study, MM1 and 2 provide insights into trends that persist across models and climates throughout the remaining cases. For example, these cases highlight the restrictiveness and relatively low MM savings opportunities afforded when the ASHRAE 55 PMV-PPD comfort model is enforced and the higher energy savings potential achievable under adaptive comfort standards. These cases also demonstrate the expectedly large gap between solutions that enforce comfort through the objective function comfort penalty and those that only seek to minimize energy use. Finally, MM1 and MM2 identify times at which expected occupant window opening behavior is close to optimal (mild weather) and periods when occupants may be overly zealous with window opening (cool or hot weather).

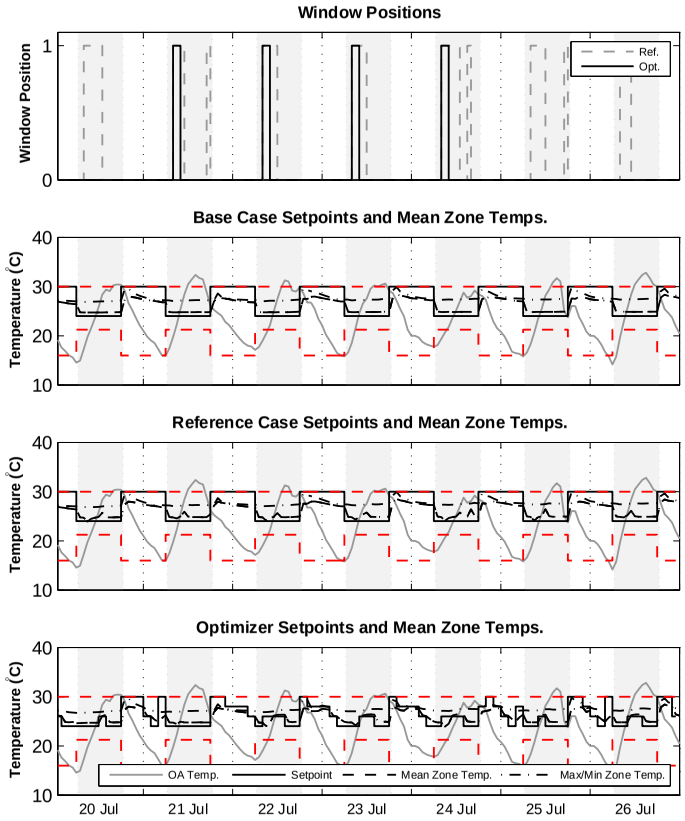
To begin, we can examine a typical solution pattern for MM1 and MM2 during the cooling season with comfort enforced according to the ASHRAE 55 "static" PMV-PPD comfort model (Figure 6.3). Each group of charts provides time series of window positions as well as base, reference

and optimal case setpoints schedules and zone conditions. The gray shaded regions represent occupied periods. One notes immediately that the optimizer opens windows only for brief periods of time at the onset of occupancy when ambient temperatures are in the upper teens Celsius. These openings can coincide with occupant openings, but most often occupants are much more aggressive with window operation and utilizing NV until at least midday. Again, this is not a surprising result since occupant window opening models are governed by adaptive comfort temperatures rather than the static PMV boundaries enforced in the optimizer solutions.

In the temperature setpoint space, a general nighttime setup can be seen, with setpoints maintained between 24 to 26 °C—the upper edge of the ASHRAE 55 summer comfort window—during occupied periods. Setpoints are abruptly set up after occupancy, with some gradual pre-cooling occurring throughout the night (usually economizer pre-cooling). As discussed in Chapter 4, equivalent solutions are present and cause a number of discontinuities in the setpoint portion of the solution, particularly during early morning hours. Some very mild daytime pre-cooling through NV can also be seen, often followed by a brief period of setpoint setups. The setpoint setups are somewhat more prolonged in MM2 due to its more massive construction and increased damping of the zone temperature response.



(a) MM1



(b) MM2

Figure 6.3: Solution patterns for a cooling season week in Boulder with ASHRAE 55 static comfort guidelines enforced. Dashed red lines indicate the box constraints imposed for setpoint decision vectors.

Under the static comfort interpretation, MM1 and MM2 receive their energy savings benefits from a combination of mechanical free cooling and coil relief associated with early occupancy NV. In MM1, for example, one can see the brief spikes in fan and DX cooling savings that occur concurrent with window openings, followed by the gradual attenuation of the cooling energy profile associated with the release of thermal storage, both from mechanical pre-cooling and NV (Figure 6.4). In MM2, the savings profile changes due to higher thermal mass and the inclusion of solar shading devices, but overall savings are relatively similar at about 60 kWh over the week.

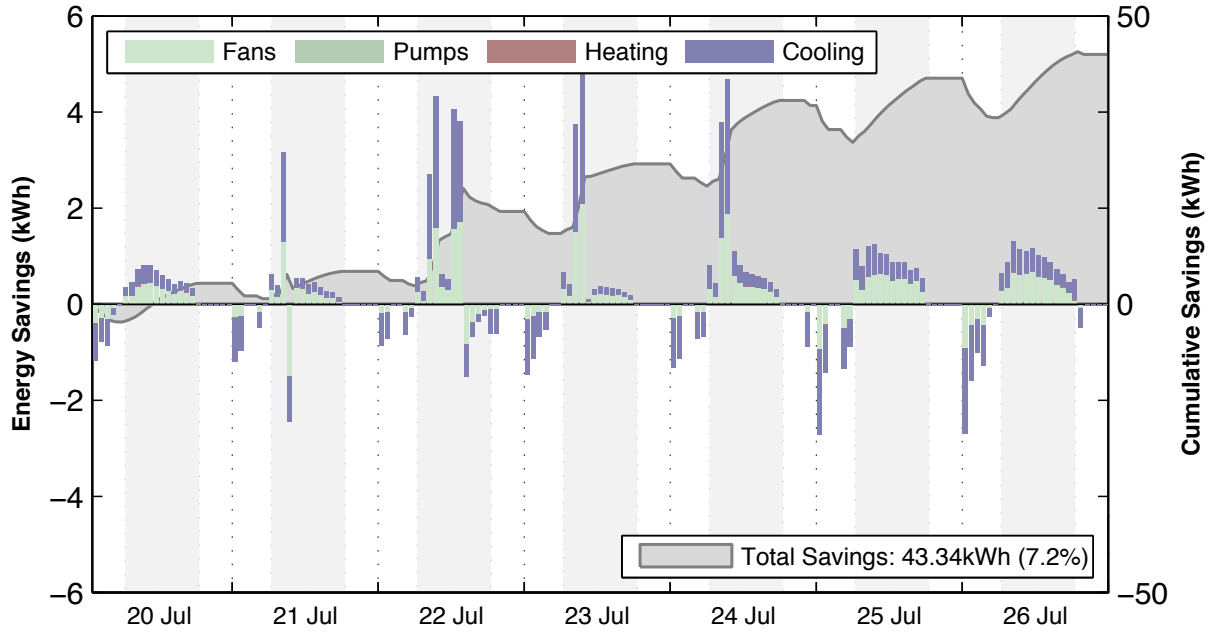


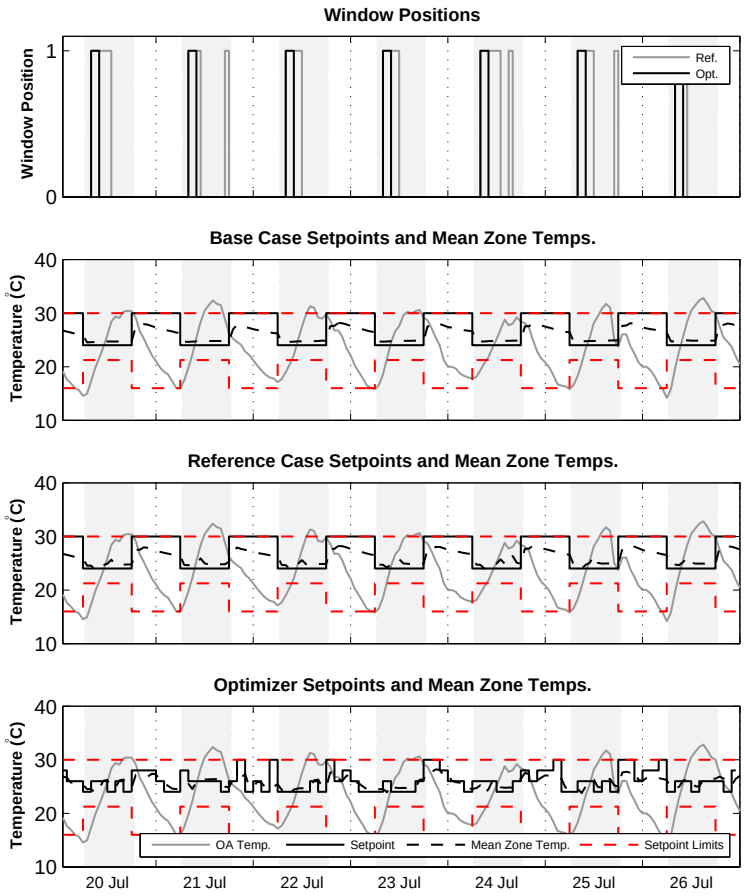
Figure 6.4: Energy savings breakdown between optimal and base case. Positive values indicate savings, whereas negative values indicate loss. Savings associated with NV appear as brief positive spikes, whereas mechanical pre-cooling first appears as negative spikes during unoccupied hours followed by more gradual daytime periods during which savings accrue.

The optimizer is, of course, restricted in its actions due to the tight comfort requirements of the static ASHRAE 55 comfort penalty. When the adaptive portion of ASHRAE 55 is chosen, significantly larger temperature ranges are permitted, and windows can be utilized daily, as shown in Figure 6.5. Mechanical pre-cooling and NV still feature heavily in the solution, but zone temperatures are allowed to fluctuate over a larger range—significantly wider than what is allowed in the

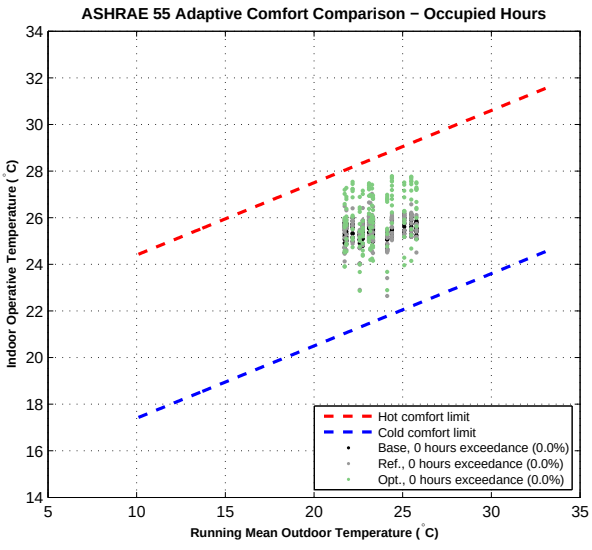
base or reference case—while still adhering to the 80% comfort limits established by the standard. MM1 more than quadruples its savings over the base case building (284 kWh or 19% saved), and manages to save over 160 kWh (25%) compared to the reference case in which occupants control windows, while maintaining zone operative temperatures entirely within the 80% comfort bounds.

Opportunities for savings are modest in the swing season, even when using the adaptive comfort penalties, and they can be difficult to obtain. The optimizer cannot be as aggressive with pre-cooling strategies due to the risk of overcooling (i.e. costs associated with heating energy and with discomfort during occupied periods). Milder ambient conditions during occupied hours means that windows can be utilized over greater periods during the day. This can generate some savings compared to occupant window control; however, when constrained to adaptive comfort criteria, the optimizer often underperforms the base case, which possesses no NV. The result is counterintuitive until one examines the 80% comfort bands imposed by the various comfort standards (Figure 6.6). During swing season periods when running mean temperatures are in the low end of the range for adaptive comfort standards, static comfort standards actually allow for warmer zone operative temperatures and therefore less cooling than the adaptive comfort standards (unless, of course, the building is entirely free running). A MM building could actually have to expend more energy to meet adaptive comfort criteria under these conditions than more typical sealed building conditions. It should not be surprising that the major savings opportunities are located on the righthand side of the graph during cooling season periods.

Overall energy savings for all Boulder MM1 and MM2 cases using the partial changeover scheme are presented in Figure 6.7. Detailed tables (including comfort statistics) and time series of relevant cases can be found in Appendix D. Solutions governed by adaptive comfort considerations yield the deepest energy savings in the cooling season and can return very similar overall savings particularly in cases where massive construction is used (MM2). Cooling season savings with adaptive comfort penalties can yield up to 50% of the maximum achievable (energy-optimal) savings. Optimal results for the swing season indicate effectively no savings opportunities over the base case. For these two building types, the optimal swing season operation in Boulder is basically to



(a) Solution pattern



(b) Thermal comfort

Figure 6.5: Cooling season solution pattern and thermal comfort plot for building MM1, using the ASHRAE 55 adaptive thermal comfort penalty.

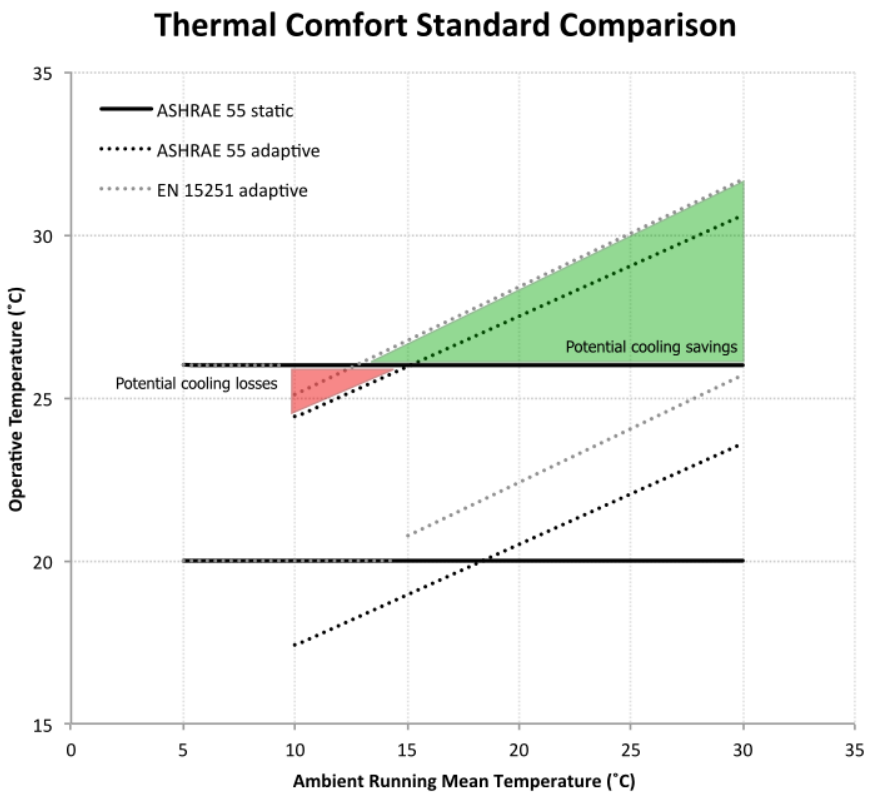


Figure 6.6: Comparison of thermal comfort standards demonstrating expected areas of loss/savings for adaptive standards compared to static standards. Upper and lower boundaries for ASHRAE 55 static fixed at approximate boundaries of summer and winter ±0.5 PMV comfort windows, respectively.

replicate base case operation, with few if any window openings until ambient temperatures warm.

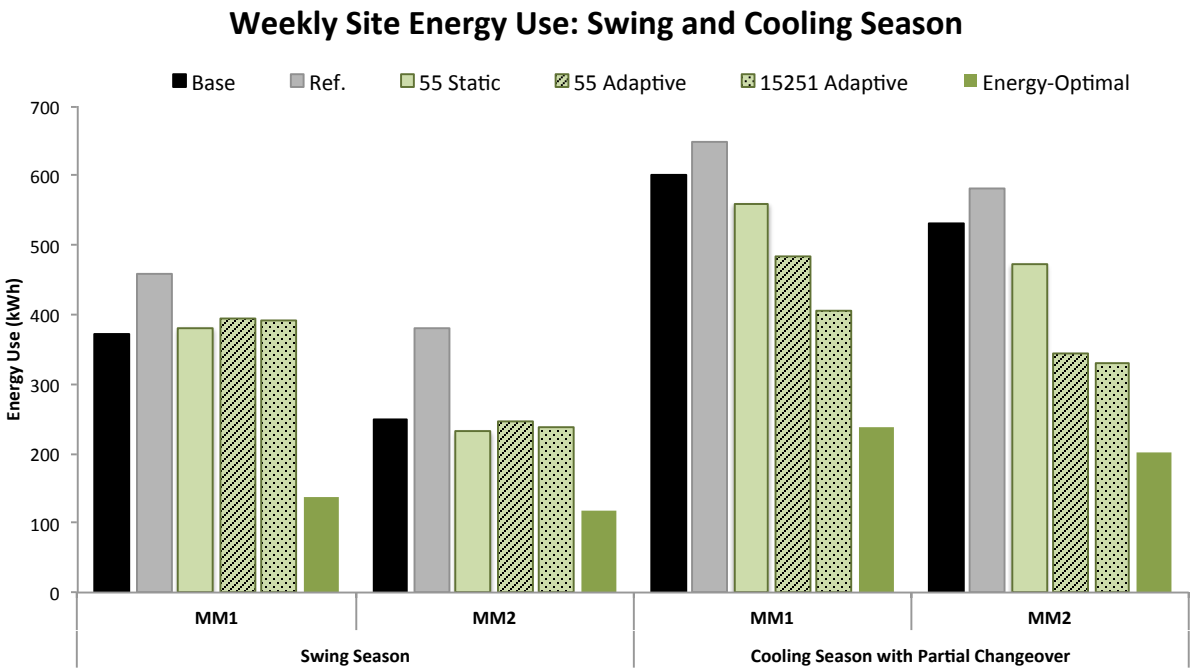


Figure 6.7: Summary of MM1 and 2 weekly energy use/savings using the partial changeover MPC formulation.

6.2.2 Optimized HVAC with Presence of Mean Occupant Behavior

Interestingly, the reference case buildings with occupant window control consistently underperform the base case, because most occupants are only aware of which decisions will afford adaptive comfort, not those that afford energy savings. Really what we see is a collision between the occupants' adaptive comfort expectations and the relatively rigid setpoint schedules expected in many buildings. In later sections, we will demonstrate how cooling setpoint reset heuristics can help to avoid this, but first it is instructive to examine the optimal setpoints that would be achieved by MPC. We can optimize global zone temperature setpoints in the presence of expected mean occupant window opening behavior, yielding the more compact nonlinear programming problem:

$$\begin{aligned} \text{Minimize } C_{tot} &= f(\vec{y}) = C_e + C_c \\ \text{Subject to} &: \\ & \vec{y} \in \mathbb{R}^{12} \\ & 16\,^{\circ}\text{C} \leq \vec{y}_{t \notin t_{occ}} \leq 30\,^{\circ}\text{C} \\ & 21\,^{\circ}\text{C} \leq \vec{y}_{t \in t_{occ}} \leq 30\,^{\circ}\text{C}. \end{aligned} \tag{6.5}$$

Setpoints are still controlled via 12 two-hour time blocks, using the same planning, execution, cost and initialization horizons.

Results show that with appropriate setpoint sequences, a 20–36% savings over the base case can still be achieved (for MM1 and MM2, respectively) even in the presence of mean occupant behavior. For the cases examined (MM1 and 2 during the cooling season, using the ASHRAE 55 adaptive comfort penalty), more than 95% optimizer savings shown in Figure 6.7 can be achieved without enforcing optimal window positions. In other words, the savings in the partial changeover solutions can often be realized without consideration for optimal window position; mean occupant window opening preferences will suffice as long as setpoints are appropriately adjusted.

As the swing season cases have shown, however, occupants might still be prone to open windows at times when outdoor temperatures are overly cool and risk activating heating. For cooler times of year, there are effectively two options to avoid this. The first is to further set back heating setpoints; the second and perhaps most practical is for facility managers to determine a seasonal changeover point that signifies to occupants when it is appropriate to begin opening their windows.

Based on a rigid interpretation of the comfort standards, Figure 6.6 suggests that this transition should occur when ambient running mean temperatures are in the range of 13–15 °C, as this is where the adaptive comfort standards begin allowing for warmer operative temperatures than

PMV-based standards. This is a very practical rule of thumb to use, as it can be applied across any climate. We can further test the validity of this assumption by examining energy signatures from the optimized results using different comfort penalties. The point at which the adaptive comfort penalties allow for lower-energy operation than the static comfort penalties is a reasonable point to establish seasonal changeover. Based on extended optimizations for MM2 in Boulder using the 55 static and adaptive penalties, we see that the adaptive solution began outperforming the static solution at average daily temperatures of just above 15 °C (Figure 6.8).[1] This also happens to be the approximate point where the upper acceptance limits of adaptive comfort standards intersect static comfort standards (Figure 6.6).

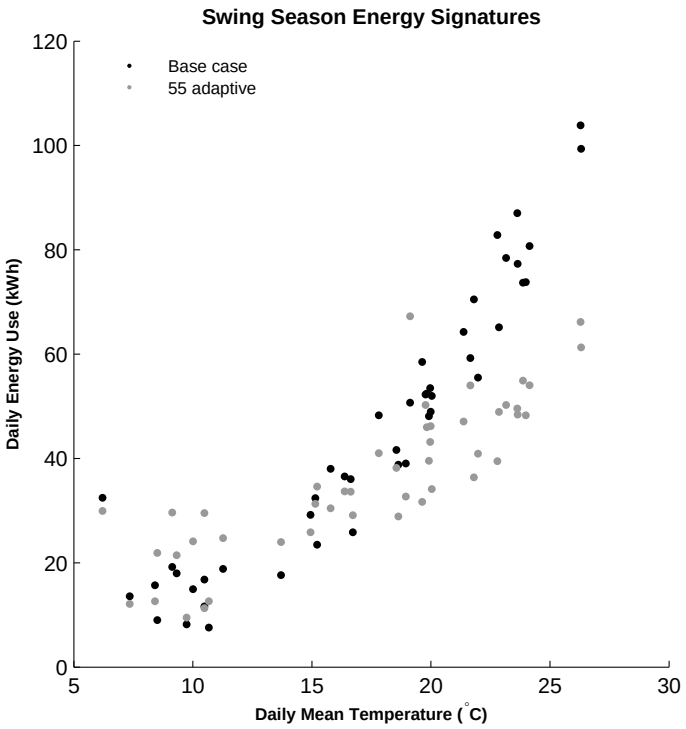


Figure 6.8: Energy signatures for the base case (no natural ventilation) and adaptive objective functions for MM2. Window openings and an adaptive comfort treatment only begin outperforming the base case at daily mean temperatures above 15 °C.

[1] Analysis based on mean daily temperatures rather than running mean temperatures provided a crisper view of this changeover point.

6.2.3 Enhancing Savings with HVAC Interlock and Changeover

Changeover is a popular control topology in MM buildings that disables mechanical ventilation and cooling on seasonal or sometimes daily frequencies when natural ventilation is in use. HVAC interlock can be enabled through window contact sensors to enforce changeover only when windows are open. An interlock-enforced changeover strategy should be the most energy efficient mixed mode topology available for a given building, but also incurs additional instrumentation, complexity, and expense. The MM1 and 2 buildings were re-optimized with such a scheme in place to examine additional savings opportunities beyond the scheme described in the section above. In both the reference and optimal cases, the entire AHU serving the three zones was disabled when operable windows were open.

The optimizer is able to achieve small improvements over the previous cases—for example, a further 10% reduction in overall HVAC energy use for the MM1 55 adaptive penalized case. The main insight gained, however, is that reference case operation is able to approach optimal performance much more closely through interlock, sometimes coming within 10% of an optimal solution (Figure 6.9). The great exception is with mass construction (MM2), in which the optimizer better utilizes thermal mass storage to effect savings.

Because there is no chance of concurrent mechanical cooling under this scheme, comfort issues become more acute, particularly in the reference case, which experiences a number of larger zone temperature spikes and a total of 13 comfort exceedance hours. The optimized MM1 case, in addition to identifying slightly greater energy savings, only sees 2 hours of very minimal comfort exceedance (Figure 6.10(a)). Results are somewhat less dramatic for buildings with improved solar gains control and heavier massing, like MM2, as they can better attenuate peak afternoon temperatures (Figure 6.10(b)).

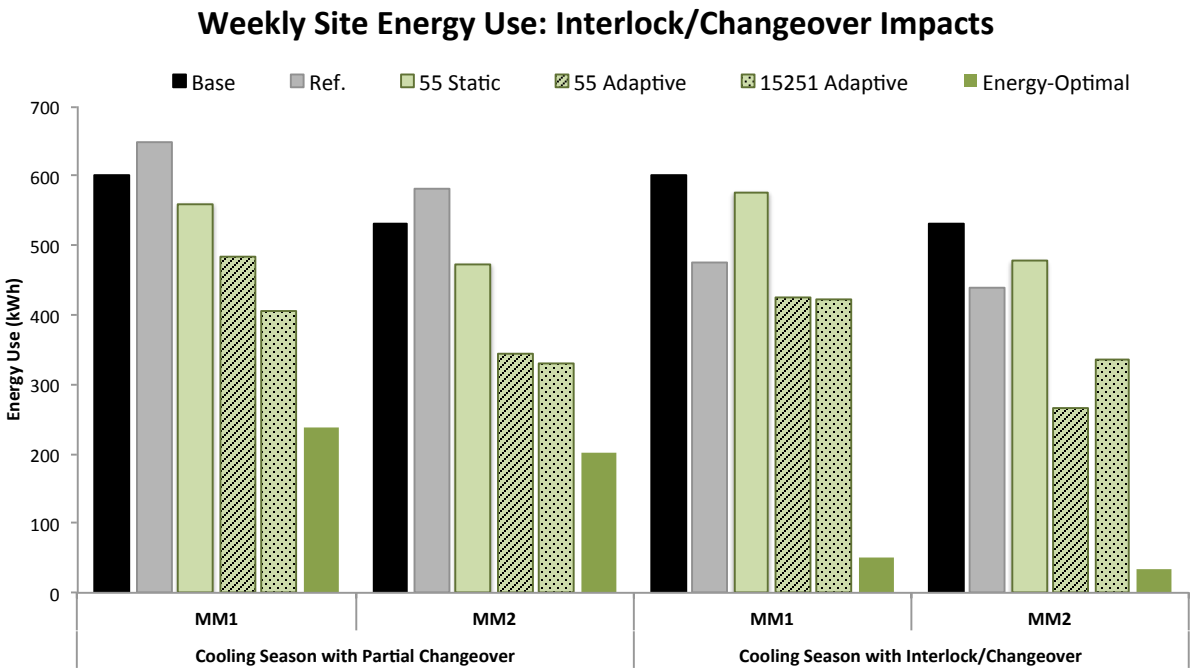
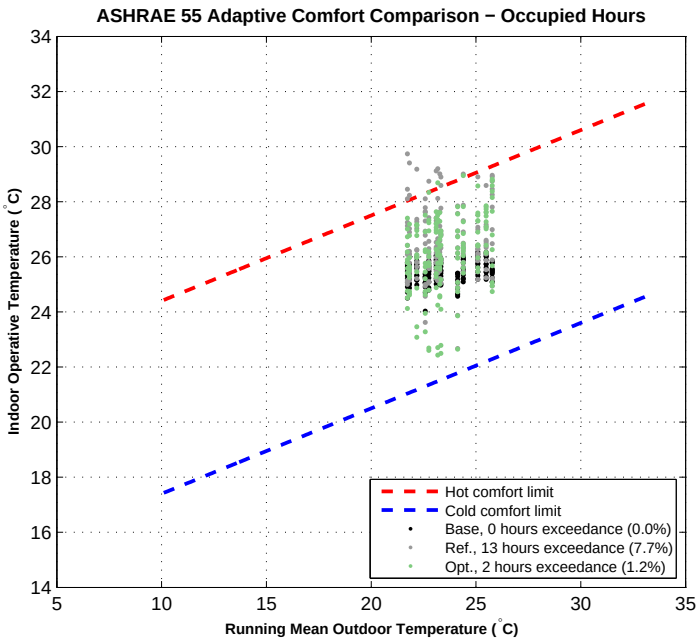
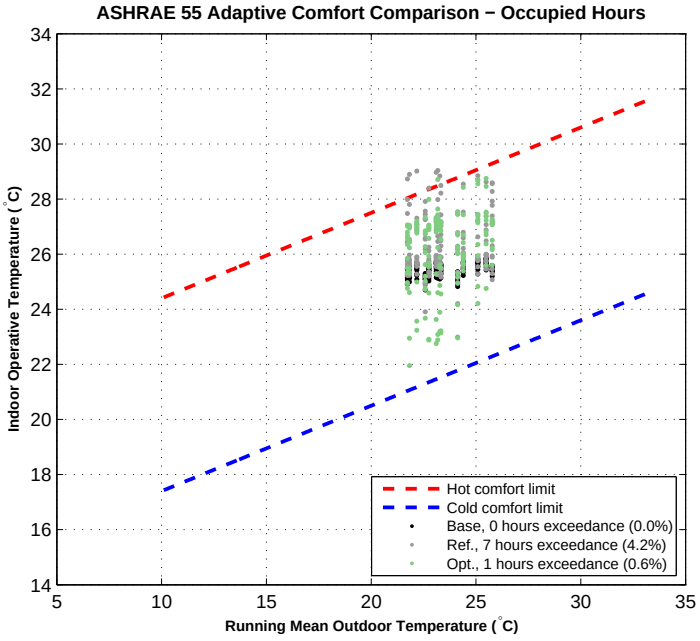


Figure 6.9: Summary of weekly energy use/savings between partial and interlock-enabled changeover operation for buildings MM1 and 2.



(a) MM1



(b) MM2

Figure 6.10: ASHRAE 55 adaptive thermal comfort plot during the cooling season, with changeover operation.

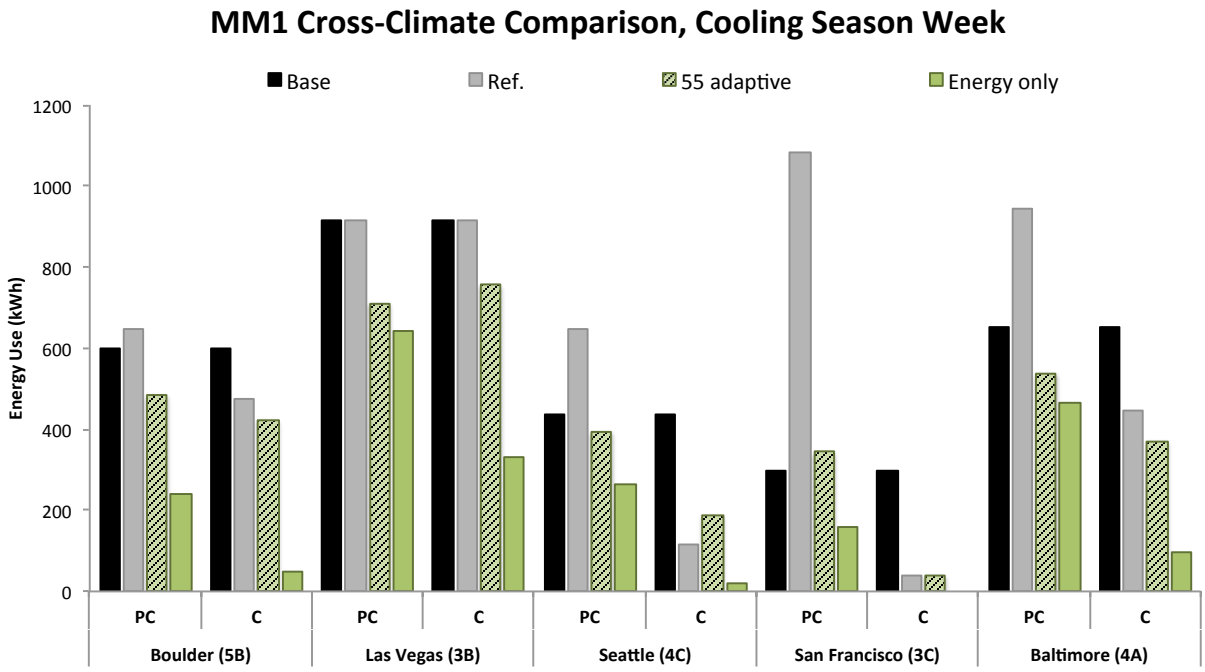
6.2.4 Cross-Climate Comparisons

A small parametric study was performed to examine control performance across several climates. Naturally the baseline energy use and relative savings potential of each building varied by climate. Figures 6.11(a) and 6.11(b) provide some interesting insights. At a high level, we see that very hot climates (Las Vegas), have very low savings potential due to high ambient temperatures (too high for NV or economizer pre-cooling), while other milder climates, not surprisingly, afford deeper savings opportunities (e.g. Seattle and San Francisco). What small savings can be gained in climates like Las Vegas are purely the result of allowing indoor temperatures to float within the adaptive comfort envelope, and it would be difficult to even justify the application of adaptive comfort in such cases since the optimizer exercises no NV.

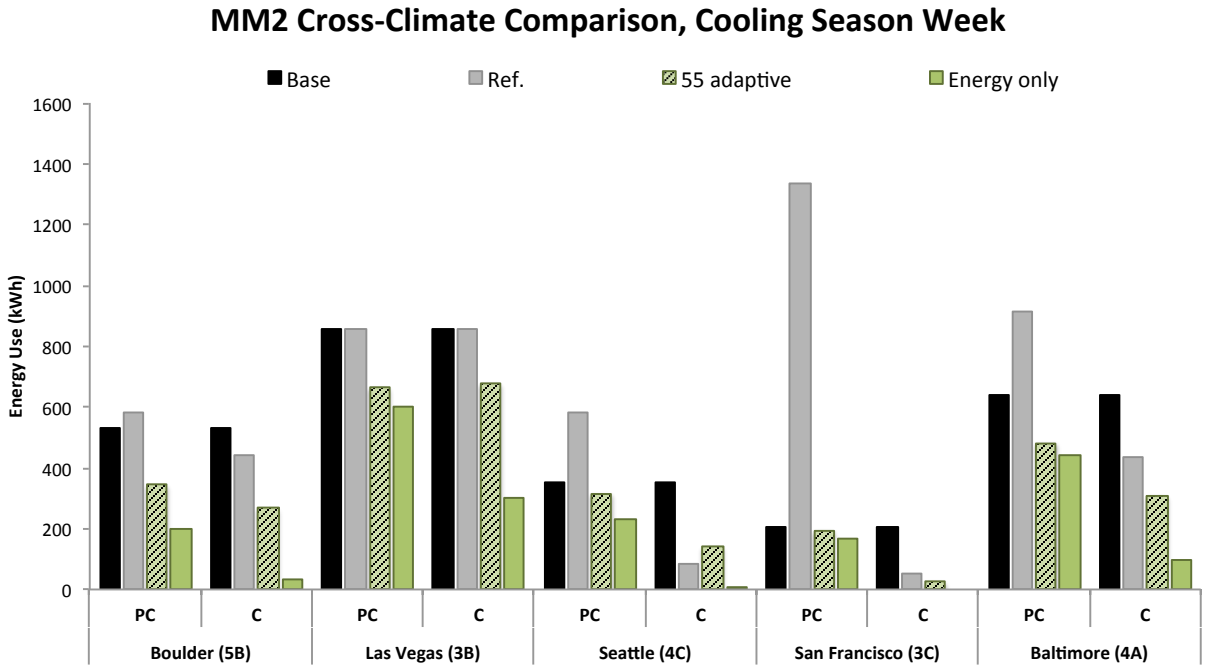
Also of interest are several cases in which typical occupant behavior results in dramatically higher energy use than the base case. This is particularly surprising for the case of San Francisco, which is known as an ideal MM climate. In the cases of San Francisco and Seattle, the differences are due to excessive heating in the occupant-controlled cases, usually during the early morning hours. In Baltimore, the difference is due to openings during overly warm or humid periods. In all of these cases, the timing and duration of window openings under MPC is significantly different than occupant control.

As with the Boulder cases, it is not that the optimizer is able to harness significant free cooling potential through daytime NV. It manages to reap savings by only opening windows for very limited cool periods and mostly by elevating setpoints to take advantage of adaptive comfort. Occupied period cooling setpoints generally trend with the ASHRAE 55 adaptive comfort region (Figure 6.12).

Across all climates except Las Vegas, incorporating interlock-enabled changeover extends savings, usually by eliminating the need for the AHU to maintain minimum ventilation requirements and associated heating/cooling while windows are open. In climates with greater diurnal temperature swings like Baltimore and Boulder, these savings are limited to about an additional



(a) MM1



(b) MM2

Figure 6.11: Summary of weekly energy use results for MM1 and 2 across several climates. MPC schemes include partial changeover (PC) and changeover (C).

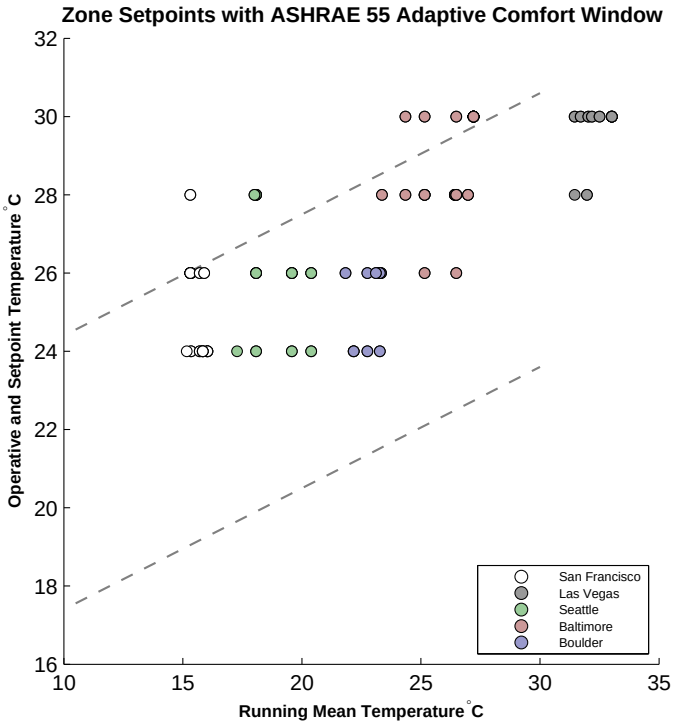
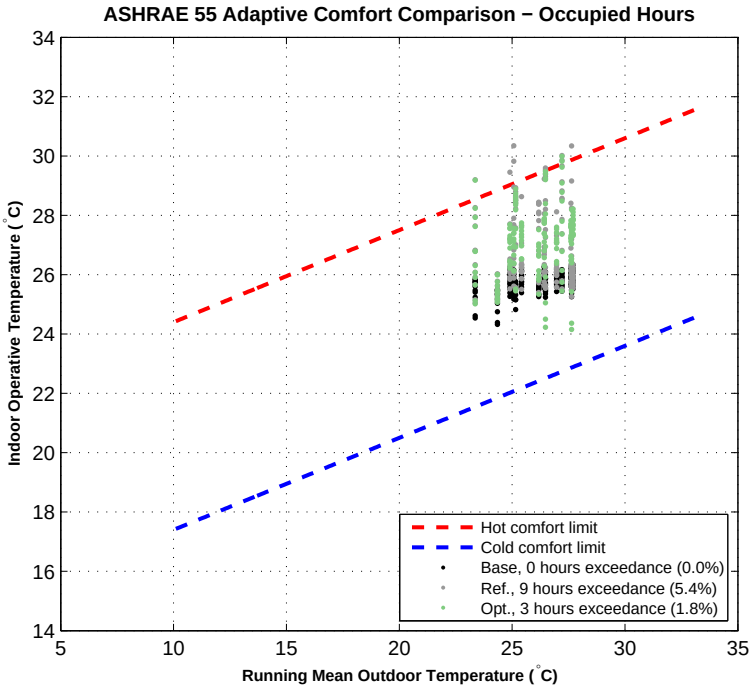


Figure 6.12: MPC-derived setpoints during occupied periods using the ASHRAE 55 adaptive comfort penalty. Setpoints broadly trend with the adaptive comfort window.

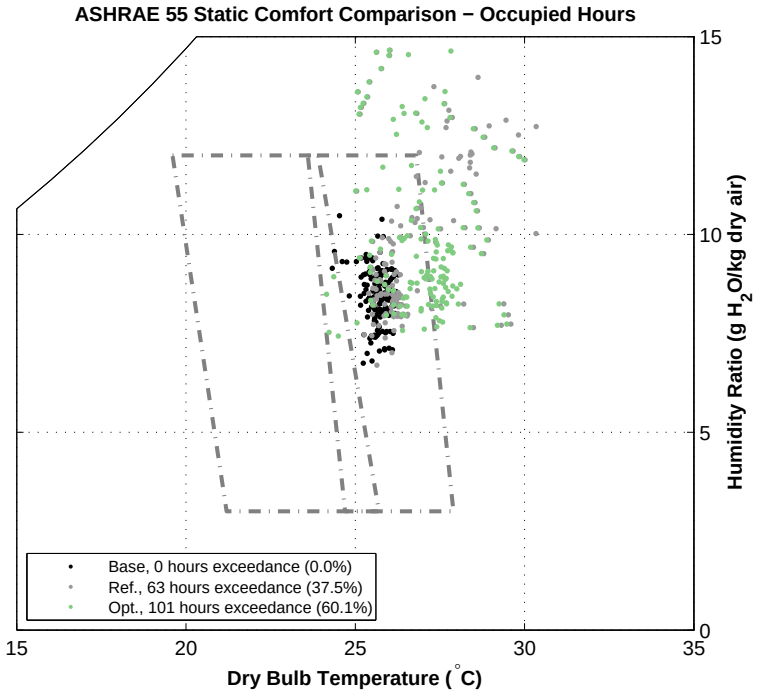
10% due to comfort constraints. However, in milder climates like San Francisco and Seattle where diurnal temperature swings are not as great and ambient temperatures are often in the range of typical supply air temperatures (15–20 °C), natural ventilation can frequently displace the AHU. Recall that this was not possible in the partial changeover cases, where the AHU continued to provide minimal ventilation air during natural ventilation periods. When hygienic ventilation continues to be supplied in these climates at SAT values typical of fully-mixed air systems, additional ventilation air overcools the space. However, when changeover is enabled, outdoor air is frequently capable of balancing zone loads in these marine climates. In the case of San Francisco, one is able to almost completely eliminate AHU operation altogether, suggesting that a seasonal changeover strategy (i.e. allowing the building to "float" during the mildest periods of the year) should be feasible. In these milder climates, the adaptively penalized optimizations often approach the level

of energy savings achieved using the energy-only objective function. The energy-only optimizations represent the maximum possible savings—or free-floating operation—achievable. Therefore, as a general rule, climates where the comfort-penalized solution approaches the energy-optimal solution can be considered good candidates for seasonal changeover strategies, where the cooling system will be taken offline for a portion of the year or may not be required in the first place.

As Baltimore is the only moist climate examined, it bears some special mention. Optimizations were conducted with a sensible adaptive comfort penalty, based on dry bulb temperatures only; humidity is not reflected in the penalty term. In a moist climate like Baltimore, this will have important implications. As shown in Figures 6.13(a) and 6.13(b), thermal comfort under adaptive criteria could be excellent, but when one examines moisture content, we find a number of hours with humidity ratios exceeding 12 $g_{water}/kg_{dry\ air}$, above the accepted upper limit for humidity in ASHRAE 55. The problem is most acute in cases where changeover is enabled (as shown) because no mechanical ventilation is present to dilute the moist outside air resulting from window openings with dehumidified air from the AHU. Under partial changeover, the number of hours exceeding 12 $g_{water}/kg_{dry\ air}$ drops from 27 to 9. Thus in moist climates, partial changeover operation may be preferred even if changeover appears to provide a greater savings opportunity.



(a) ASHRAE 55 adaptive comfort



(b) ASHRAE 55 static comfort

Figure 6.13: Thermal comfort plots for MM1 in Baltimore using the ASHRAE 55 adaptive comfort penalty, with changeover enabled. Moisture content becomes very high during a subset of the occupied hours.

6.3 Results: MM3

MM3 represents a significant increase in building system sophistication from MM1 and 2. Low-energy UFAD systems assuming temperature stratification are used, with perimeter hydronic baseboard radiators instead of reheat. Actuated transom windows above the vision glazing and shading devices allow for automated night flush ventilation.[2] Finally, interlock is enabled on the operable vision glazing, disabling cooling and ventilation to a zone when occupants open windows.

6.3.1 Representative Results for Boulder, CO Climate

MM3 presents a more interesting MPC problem from the standpoint of free cooling because night ventilation is possible. This predisposes the optimizer to exercise night ventilation rather than mechanical pre-cooling, and one expects to see far less cooling setpoint suppression prior to occupancy (unlike results from the previous section). Supply air is also delivered at a significantly higher 18 °C per standard UFAD design, providing coil relief and also enabling slightly greater use of NV during occupied periods, even in partial changeover operation.

The MM3 MPC solutions show significant savings over the base case, but are unable to outperform the reference case. Recall that the reference case employs interlock-enforced changeover, so the partial changeover scheme investigated by the optimizer was physically incapable of outperforming this scheme. We will see in upcoming sections that, even when the optimizer solution allows changeover operation, it is still very difficult to do much better than the reference case control scheme. MM3 solutions utilize a combination of appropriate daytime window openings, cooling setpoint setups (to allow float within the prevailing comfort boundaries), and additional nighttime pre-cooling through night flush ventilation. Nighttime economizer pre-cooling is less common, with night ventilation displacing it whenever possible (pre-cooling with the UFAD system is problematic due to high SATs). The night flush ventilation opportunities in Boulder are bountiful, due to the dry climate and associated large diurnal temperature variation. Representative solutions for the

[2] Transom windows here are located above vision glazing and exterior passive shading, enabling night ventilation to the zones, as shown in Figure 6.1.

swing and cooling seasons using the ASHRAE 55 adaptive comfort penalty are shown in Figures 6.15(a) and 6.15(b). The MPC solution chooses to engage night ventilation much more frequently than the reference case, which uses a proportional controller to modulate night flush ventilation. This allows the MPC controller to more deeply pre-cool the thermal mass while still observing comfort boundaries at the onset of occupancy. Night ventilation is utilized in both the swing and cooling seasons, although to a lesser degree in cooler weather, as one would expect. Even with this greater harvesting of free cooling, the reference case's interlock/changeover system manages better energy performance by simply disabling ventilation and cooling during daytime NV periods.

The introduction of night ventilation in MM3 allows the building to more fully exploit the 6K-wide 80% satisfaction limits in the adaptive comfort standard (Figure 6.14). Effective pre-cooling is achieved without violating the lower comfort boundaries during early occupancy. The building can then float through the 6K band to the greatest degree possible, with mechanical cooling usually only activated in the late afternoon to avoid overheating.

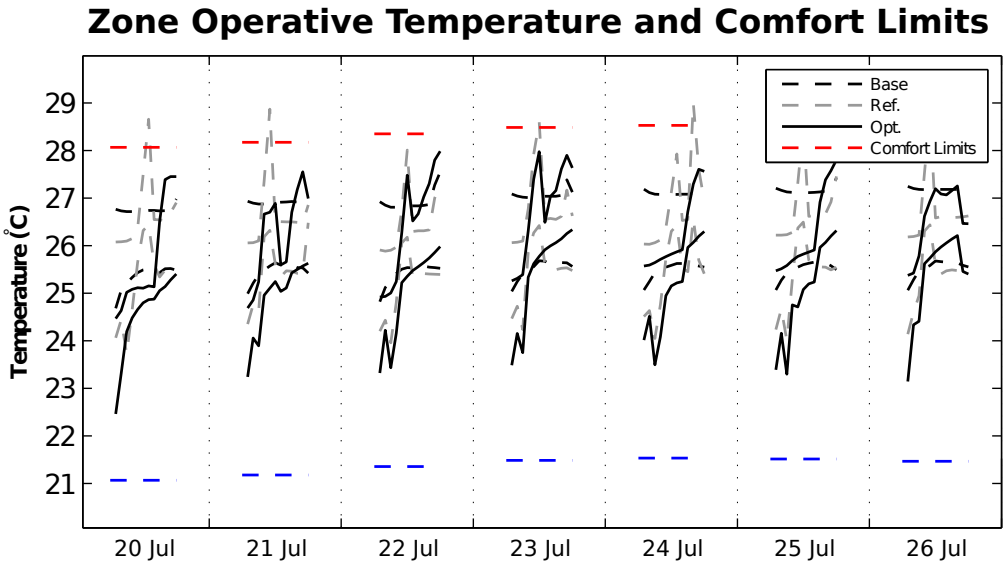
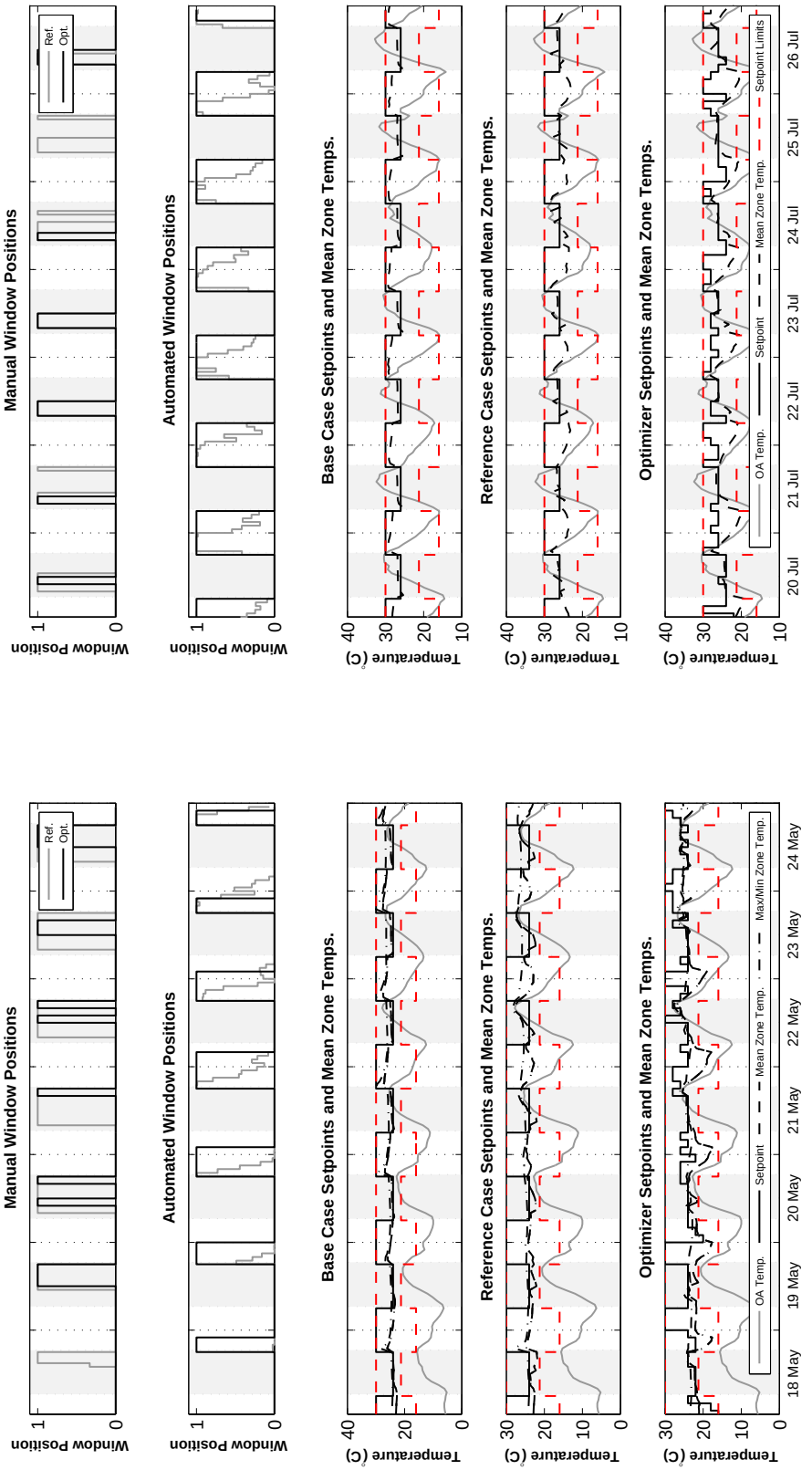


Figure 6.14: Plot of zone operative temperatures for MM3 with the ASHRAE 55 adaptive comfort region. Both the reference case and the MPC solution make use of nearly the entire 6K comfort band on most days through pre-cooling.

The solutions also yield other insights with interesting implications for MM system design.

Recall in the MM1 and 2 swing season cases (Figure 6.7), the MPC controller found few opportunities for natural ventilation, particularly during the swing season. However, in MM3 the optimizer exercises occupant-controlled windows more frequently under similar conditions and still maintains acceptable thermal comfort and energy savings. The change in airside systems can explain this difference. The higher SAT for UFAD systems opens possibilities for NV under partial changeover operation. We can examine the case when the optimizer is attempting to provide free cooling with natural ventilation and has set up the cooling setpoint, allowing VAV dampers to close to their minimum position. Under these conditions in MM1 and MM2, the VAV box continues to deliver 13 °C supply air into the zone at minimum outdoor airflow rates; in the MM3 case, VAV boxes would deliver 18 °C air at significantly lower flow rates. This means that natural ventilation is able to displace mechanical cooling under a wider range of conditions. When ambient temperatures are cool, higher SATs result in a lower risk of overcooling the zone; when ambient temperatures are warmer, the near-neutral SATs provide coil relief compared to fully-mixed systems.



(a) Swing season

(b) Cooling season

Figure 6.15: MM3 typical swing and cooling season solutions for the ASHRAE 55 adaptive comfort penalty case.

A summary of results across all models presented thus far is depicted in Figure 6.16. The MM3 reference case, utilizing night ventilation and changeover, cuts HVAC energy use compared to the base case by nearly 50%, achieving effectively the same level of operational efficiency capable of the optimizer under partial changeover operation. As with MM1 and 2, we see that, at best, only about half of the energy-optimal savings can ever be recovered, due to comfort considerations imposed in the MPC problem.

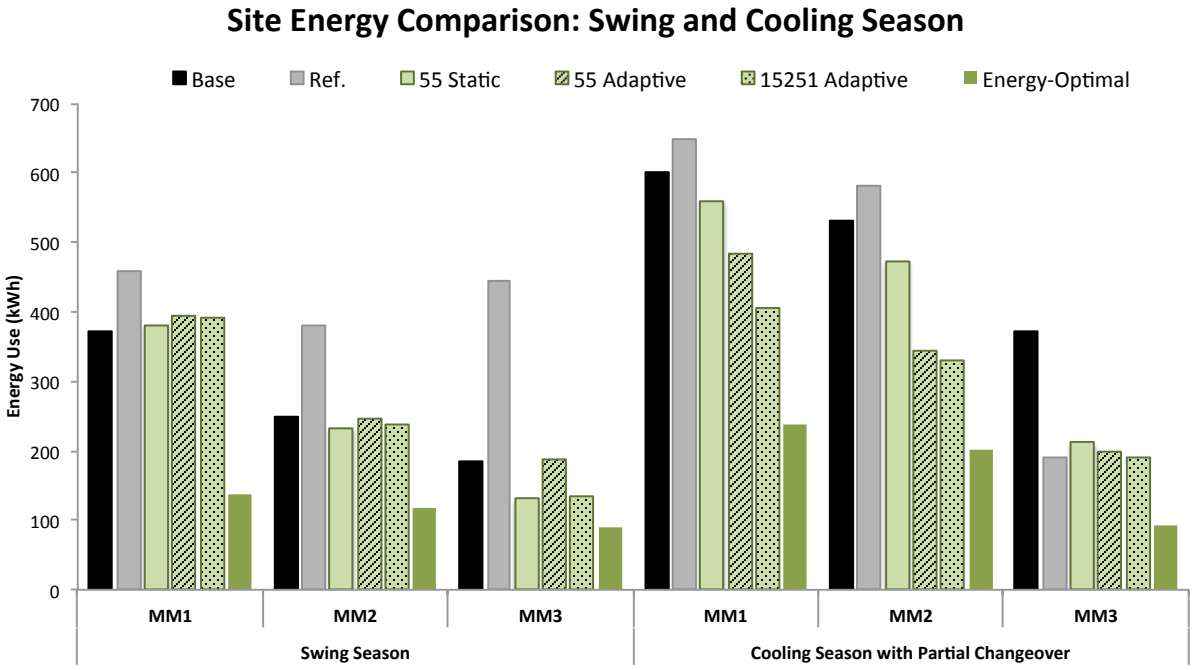


Figure 6.16: Weekly energy use for models MM1 through 3 in swing and cooling seasons.

6.3.2 Optimized HVAC with Mean Occupant Behavior

As with MM1 and MM2, it is practical to examine the performance of the MPC controller in the presence of mean occupant window opening behavior, as occupants are the most common means for controlling office windows during the day. Interlock is enabled on operable windows as in the reference model. From a thermal comfort standpoint, the solution is effectively identical to

the partial changeover and changeover cases above. In terms of actual decision vectors, there are noticeable differences in cooling setpoints (Figure 6.17). A greater attempt at pre-cooling is made prior to the occupied period, charging zone thermal mass so that it can float through periods when windows are opened and airside systems disabled. The periods of occupant-controlled openings last longer than those chosen by the optimizer, hence the additional thermal preparations. When windows are closed during warmer daytime periods and cooling systems are activated, zone setpoints hover in the mid- to upper-20s, which maintains zone temperatures below the adaptive comfort upper limit. Night ventilation patterns remain almost identical for this cooling season case.

Under this scheme, energy performance is similar to the analogous changeover case shown in Figure 6.19, with weekly energy use around 160 kWh. This is actually slightly lower than the energy use of the changeover case, but with higher comfort violations (8 hours vs. none). The lower energy use results from longer daytime NV periods—and therefore interlock—under occupant control compared to MPC. As with MM1 and MM2, we find that as long as outdoor temperatures fall within a reasonably mild range, we do not sacrifice significant energy savings or comfort by allowing occupant control of windows. However, there are large benefits to be gained by more effectively controlling thermal mass through either free or mechanically driven nighttime pre-cooling and by maintaining daytime cooling setpoints that conform better to the adaptive comfort criteria in use.

Another observation from the MM1 and 2 cases holds for MM3: occupants may begin opening their windows too early in the season when outdoor temperatures are too cool, if the behavioral models are to be believed. The Humphreys model used throughout this work suggests that occupants may be inclined to open windows even when outdoor temperatures dip well below established heating setpoints (see Figure 3.6 for measured data on this). In free-running buildings, this poses no problem, but in a building that operates part of the time as a "sealed" building and other times as a free-running NV building, there are consequences, namely that occupants unintentionally introduce heating loads. Allowing perimeter heating systems to be included in the interlock signal would help prevent this problem in MM3.

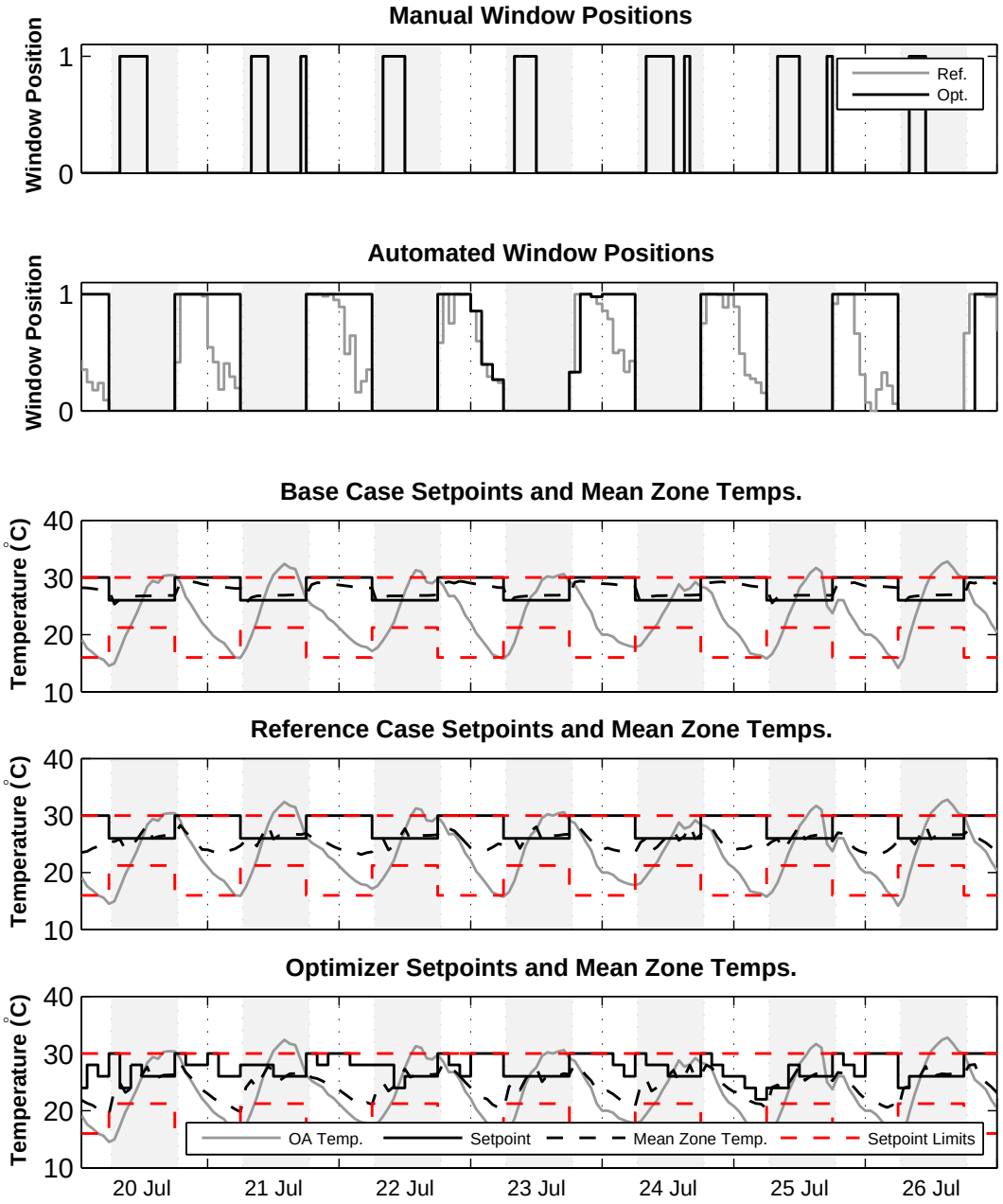


Figure 6.17: MM3 solution in the presence of mean occupant window control and using the ASHRAE 55 adaptive comfort penalty.

The examination of the various static and adaptive comfort standards used today demonstrates at a basic level that adaptive comfort standards should only be expected to provide cooling benefits at outdoor running mean temperatures in excess of 13–15 °C (see Figure 6.6). Further examination of MPC results show that adaptively penalized MPC solutions (i.e. those most likely to allow window opening behavior) only began to accrue savings at mean daily temperatures around 15 °C. Even though MM3's energy signature differs greatly from MM1 and 2 (Figure 6.18), we see that this seasonal changeover point still falls slightly above 15 °C, as it did in the previous cases.

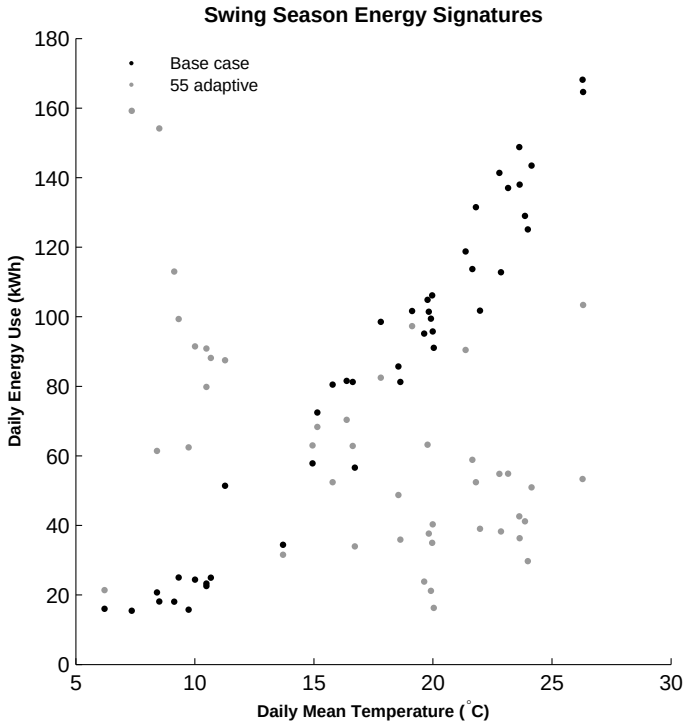


Figure 6.18: Comparison of energy signatures from the sealed base case and the ASHRAE 55 adaptive comfort penalty solutions show that window operation should be encouraged only after average daily temperatures rise above about 15 °C

6.3.3 Enhancing Savings with HVAC Interlock and Changeover

Occupant window openings engage an HVAC interlock that disables the AHU by default in the MM3 reference case. If we extend this capability to the optimized cases, savings naturally

improve as in the MM1 and 2 cases. As Figure 6.19 shows, about 8% additional savings can be achieved by using changeover when adaptive comfort is enforced. In the case of ASHRAE 55 static comfort enforcement, interlock prevents the optimizer from pursuing more efficient strategies without incurring steep comfort penalties. Because the mechanical system is unable to provide concurrent cooling and because the static comfort region is significantly more restrictive, the static comfort cases actually require fewer window openings under changeover to avoid zone overheating. Energy performance results in the two adaptive comfort cases are similar to each other. Optimizer decision vectors are relatively similar as well, with savings derived not so much from a different control sequence but from the interlock itself.

However, one observes that even under MPC, the optimizer is only able to manage a small gain in energy savings over the reference case. In fact, the MM3 reference control heuristics come closest of all models studied to replicating optimizer results. The combination of night ventilation coupled with HVAC interlock provided a highly efficient operational pattern with good comfort, and the optimizer was only able to identify single-digit energy savings and marginal comfort improvements (i.e. 6 fewer hours of comfort violations). It is possible that with further granularity of the MPC time blocking scheme, a marginally better solution could be found. Alternately, the decision vector for window positions could be more finely parsed, allowing fractional window positions rather than binary ones. Regardless, for the purposes of this simulation study it suffices to simply state that the MM3 reference strategy provides about the best balance of comfort and energy use of all buildings investigated, one allowing for only marginal improvements in sight of comfort considerations.

6.3.4 Cross-Climate Comparisons

A cross-climate study was performed on MM3, using cooling season weather and the ASHRAE 55 adaptive comfort penalty. Optimizations were also conducted on the energy-only objective function to determine an absolute minimum energy consumption value that best represented free floating operation. Partial changeover (PC) and changeover (C) operation were both considered. Reference case energy use does not change between PC and C cases because interlock/changeover

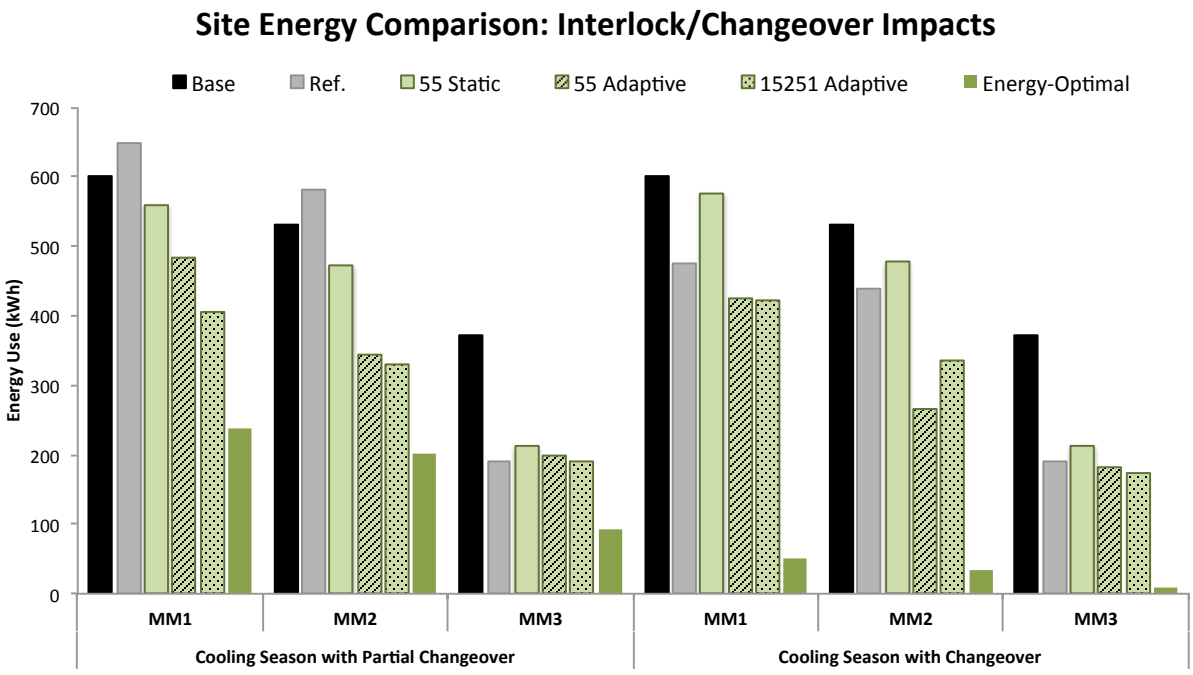


Figure 6.19: Weekly energy use comparison of partial changeover and changeover optimizations for models MM1 through 3. Note that the MM3 reference cases include interlock-enabled changeover by default.

is enabled by default.

Some of the same oddities seen in the MM1 and 2 climate study persist in MM3 (Figure 6.20). For example, a large increase in energy use in the San Francisco reference cases is seen due increases in heating energy use. Also recall that in the MM1 and 2 Las Vegas cases, there was very little potential for savings through window operation. With MM3, Las Vegas cases see some benefit from night ventilation, but the optimizer maintains closed windows during the day. With operable windows closed during the day, it is hard to justify the use of an adaptive comfort penalty, so the savings shown in Figure 6.20 would not entirely be realized in practice (even though as the Las Vegas reference case shows, there is opportunity for some night ventilation savings in the 3B climate).

As in previous cases, savings are most promising in the mild marine climates (3 and 4C), although night ventilation can still prove highly valuable in more temperate climates with greater diurnal temperature swings (5B and 4A). In the MM1 and 2 cases, the marine climates were capable of almost completely eliminating all HVAC energy use during the cooling season through use of interlock/changeover. This is not possible in the case of MM3 due to the configuration of the heating system. In MM1 and 2, all heating and cooling were disabled by changeover operation because both operated through the air system. In MM3, supplementary heating is provided by perimeter baseboard heaters which were not disabled by interlock. San Francisco's high reference case energy use is evidence of this. Inclusion of perimeter heating systems in the window interlock/changeover system is ultimately more true to the spirit of the adaptive comfort standards and would enable greater savings. Follow-on optimizations show that it is possible to bring HVAC energy use down to near zero (9 kWh during the cooling season week) while still observing the ASHRAE 55 adaptive comfort standard, simply by including perimeter heat in the interlock signal. As a result, the optimizer makes almost no use of night ventilation to avoid overcooling the zone in the morning.

The observation that MM3's reference heuristic provides near-optimal performance held true for a number of climates. The combined natural ventilation scheme and HVAC interlock yielded energy performance within 10% of the optimal for several climates, including Boulder, Las Vegas,

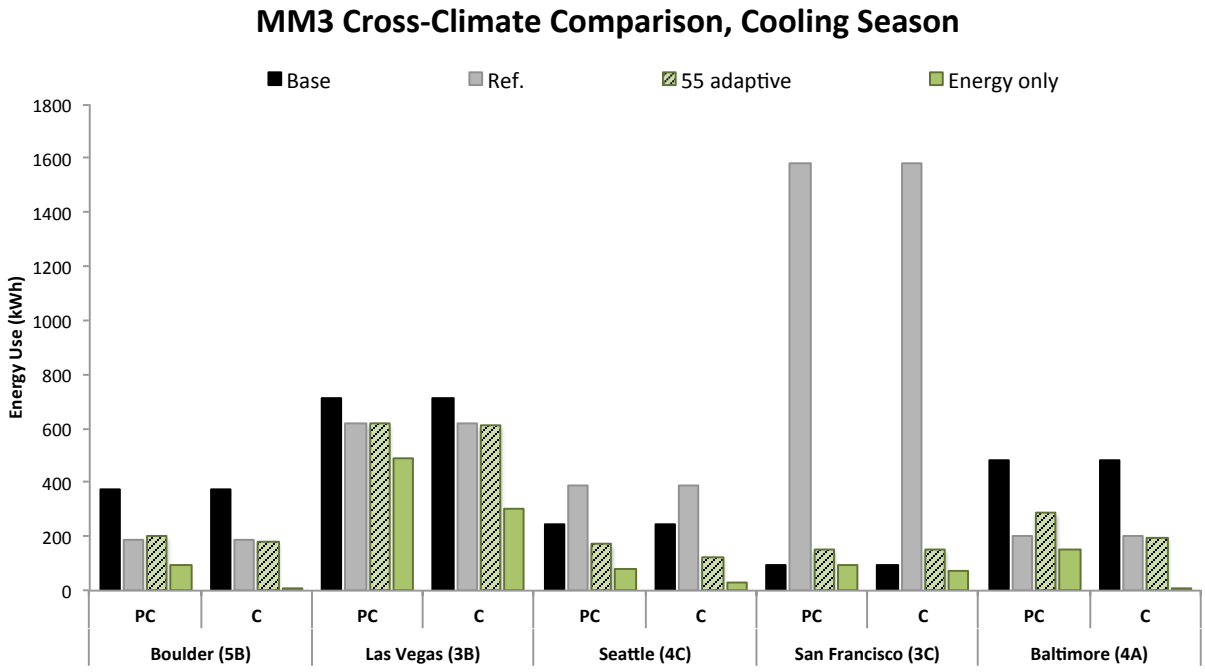


Figure 6.20: Weekly energy use results for MM3 across several climates under cooling season conditions.

and Baltimore. In the cool marine climates (Seattle and San Francisco) where occupant window control introduced heating loads, this behavior was not observed, and the optimizer was able to identify more efficient policies that avoided excessive heating operation.

6.4 Results: MM4

MM4 is arguably the most sophisticated and difficult to control of all the four typical models examined. It utilizes a DOAS in UFAD configuration, as well as automated exterior shades to control solar gains. Most importantly, zone conditioning is provided by chilled/heated ceilings in the zones. The only dehumidification source is the chilled water coil in the AHU for ventilation air. The central plant consists of several GSHPs, one of which can provide free cooling from the ground loop. This model was examined only on the Boulder climate. It is given a brief treatment here due to the more detailed investigation of a similar case used in the field experiment in Chapter 8.

6.4.1 Concurrent Operation

Under concurrent operation, operable windows are allowed to be engaged simultaneously with radiant cooling and DOAS operation. Manually operable windows are optimized during the daytime, night ventilation windows during unoccupied periods, and a PWM signal for the central and local zone circulation pumps is optimized for all times of day. A common cooling season solution pattern is shown in Figure 6.21. One of the most obvious differences between the MPC-controlled case and the others is the significant reduction in TABS cooling and circulation. Once the optimizer pattern emerges, pumps are allowed to operate on a 50% duty cycle, with the end result of reducing both pump and cooling energy. This is analogous to the frequent elevated setpoints seen in the MM1–3 cases above and are enabled partly as a result of NV pre-cooling and also due to the wider adaptive comfort boundaries. TABS circulation still occurs on a regular diurnal pattern, with most operation generally occurring during unoccupied hours; however, we do see some amount of operation during occupancy, which differs from the “night pulsing” strategy employed in the base

and reference cases.[3]

Daytime operable window openings are much more closely aligned with the mean occupant behavior, as opposed to earlier cases—particularly in the more conventional MM1 and 2—in which daytime openings were limited to very brief early morning periods. The exposed, chilled mass has the desired effect of suppressing operative temperatures and helps expand viable window opening periods, even when ambient temperatures are in the 25–30 °C range. The use of low-exergy, water-side cooling sources like GSHPs also helps to expand the daytime window opening periods, because the energy term of objective function is now significantly less sensitive to brief zone temperature spikes that would have generated immediate coil loads in earlier cases.

The overall energy consumption impacts of the various MPC controllers versus base and reference cases are presented in Figure 6.22. Not surprisingly, the overall energy consumption for MM4 trends lower than all other cases due to the use of ground heat sinks/sources coupled to radiant cooling/heating systems. Under cooling season conditions, the optimizer is able to shave another 30–40% off the HVAC energy consumption compared to the base and reference buildings. In the swing season, results are not as impressive and are, in fact, slightly higher than the reference case. The difference, as in MM3, is the presence of interlock/changeover capabilities in the reference case, which provide some fan and cooling energy savings during occupied hours in the swing season. Despite the higher energy use, the optimizer solutions provide improved comfort. For example, in the ASHRAE 55 adaptive cooling season case, the optimizer allows a very mild comfort exceedance for one hour, whereas the reference case six hours of significantly deeper exceedances due to overcooling.

[3] Night slab charging is also a common operational strategy in existing facilities with TABS.

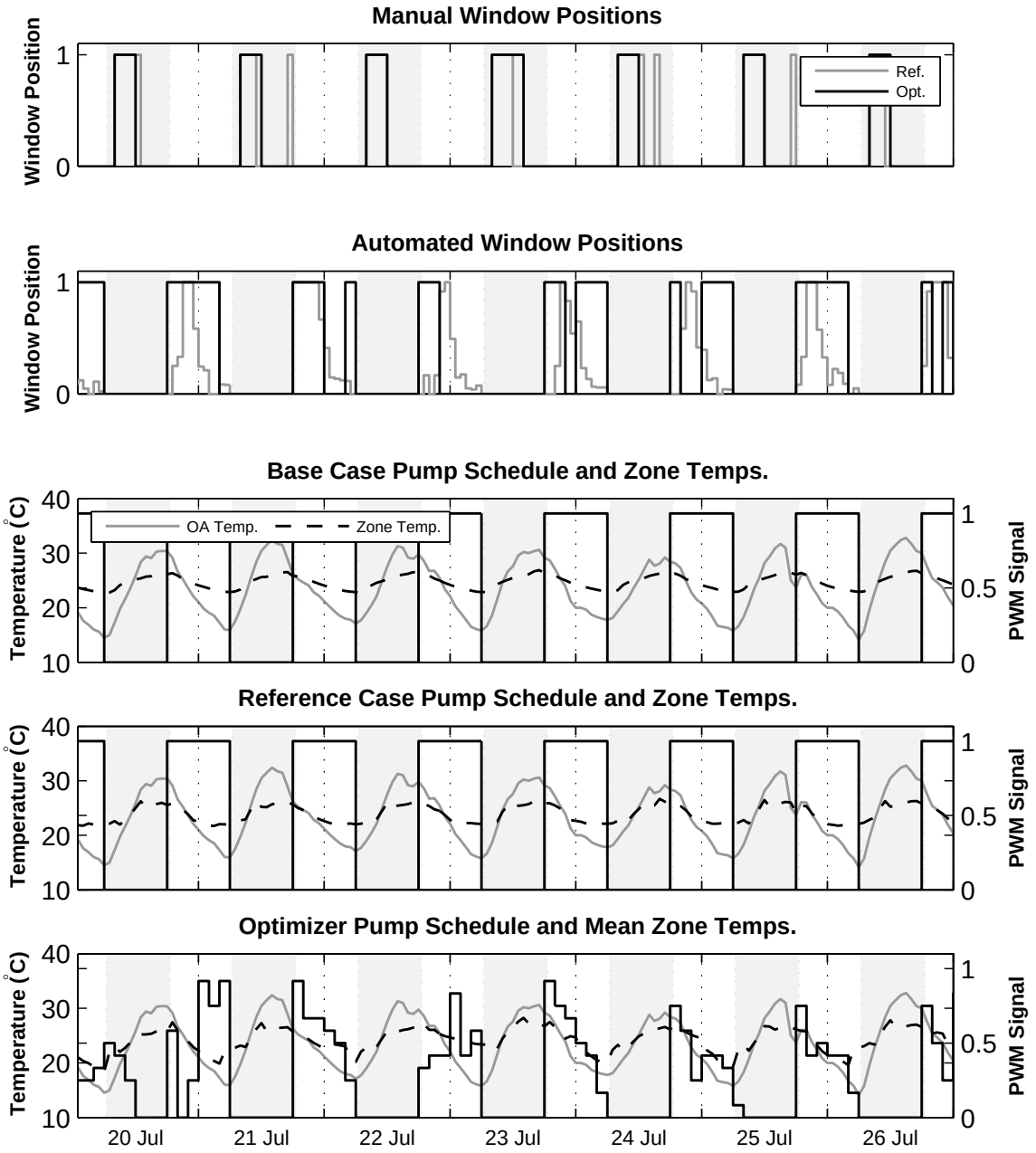


Figure 6.21: MM4 solution for the cooling season case using ASHRAE 55 adaptive comfort penalty.

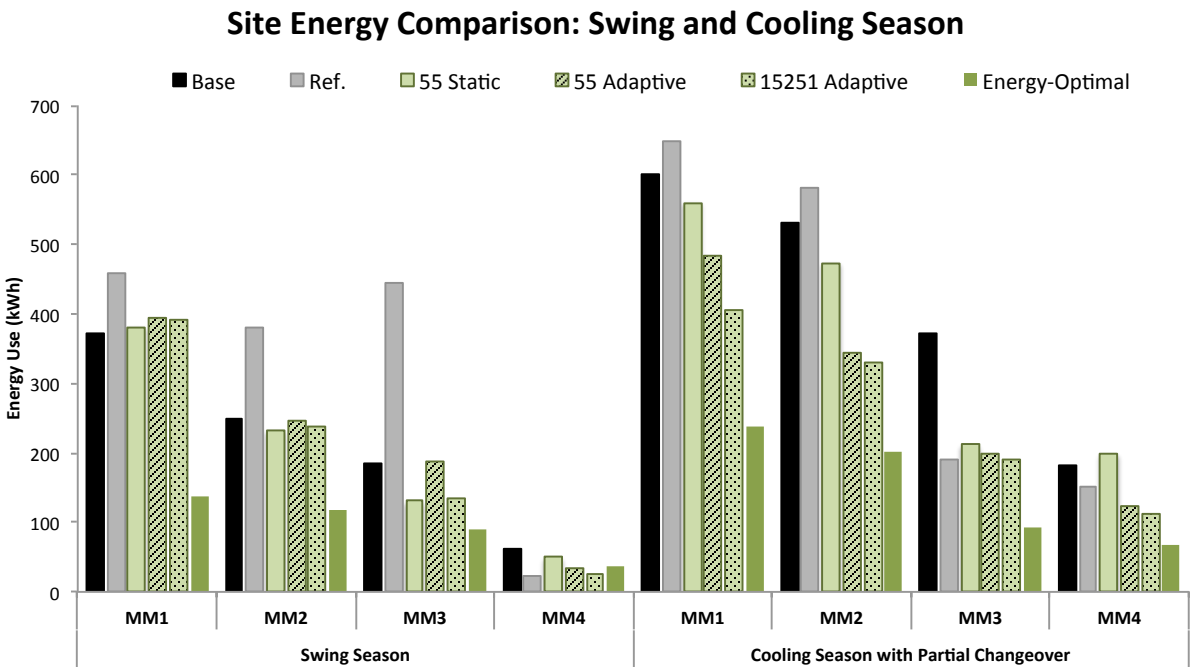


Figure 6.22: Summary of MPC solutions for MM1–4 during the swing and cooling season weeks.

6.4.2 Changeover Operation

Since cooling is provided through concrete-coupled hydronic loops, the time constants associated with the radiant cooling system are an order of magnitude greater than those associated with the zone air node, to which the NV and the air side of the HVAC system are directly coupled. This mismatch in time constants means that imposing interlock/changeover on the radiant cooling system may not make sense due to long time lags. Instead, changeover can be applied to the DOAS system, disabling flow when windows are open. The energy savings impacts are relatively minor on an absolute basis (an additional 23–46 kWh for the adaptive cases), but nevertheless eliminate some of the remaining HVAC energy use (20–40%). A summary of changeover operation for MM1–4 is provided in Figure 6.23.

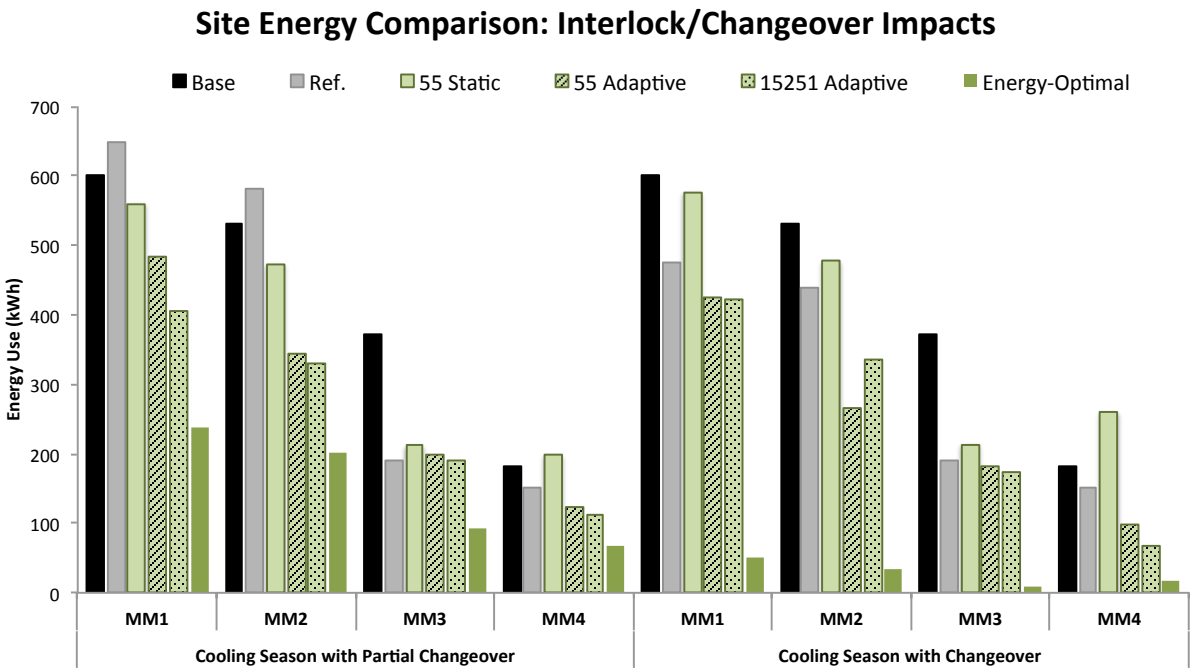


Figure 6.23: Comparison of partial and full changeover operation for buildings MM1–4 during the cooling season week.

6.5 Results Summary

6.5.1 MM1 and MM2

- The greatest savings opportunities for a partial changeover system occur when enforcing adaptive comfort, particularly in the cooling season when setpoint schedules enable building to float before or after periods of window opening.

- In buildings without night ventilation, mechanical economizer pre-cooling is the main savings mechanism.

- As such, it is not as critical that automated controls be used to manage daytime openings. What is crucial is that mechanical systems are controlled in a way that anticipates adaptive behavior, either through partial changeover or changeover. Optimizing cooling setpoints in the presence of expected mean occupant behavior reduced energy savings by only a few percent.

- Changeover offers the best savings opportunities, with additional savings of up to 25% possible through interlock systems.

- During swing season periods in Boulder, we see little to no energy savings to be harvested, even under MPC. The results suggest that facilities should allow static comfort to prevail until daytime temperatures rise above about 15 °C, then allow building occupants to operate windows.

- Temperate climates like Boulder and Baltimore proved the best match with partial changeover strategies. In moist climates like Baltimore, partial changeover is particularly encouraged to allow some concurrent introduction of dehumidified air.

- Milder marine climates like Seattle and San Francisco seem particularly well suited to changeover operation (either seasonal or interlock-enabled) due to more stable temperatures and longer periods of time during which the building can float.

6.5.2 MM3

- Adaptive comfort-penalized solutions still continued to provide the best energy savings opportunities in the cooling season.

- Whereas mechanical pre-cooling tended to dominate the energy savings seen in MM1 and 2, night ventilation enabled through automated transom windows provided most of the benefit.

- The general design principle of UFAD and other systems that deliver air into the zone at room neutral conditions (e.g. displacement ventilation) hinders the applicability of mechanical pre-cooling in this case, particularly when the optimizer has the choice of obtaining pre-cooling directly from NV instead.

- One advantage of the higher SAT used in UFAD designs appears to be that it affords a wider range of optimal daytime window openings. The warmer minimum ventilation air delivered during these times does not overcool the zone and negatively impact comfort.

- As with the MM1 and 2 cases, occupant control of windows is not problematic as long as the HVAC system anticipates adaptive behavior. Optimizing cooling setpoints in the presence of expected mean occupant behavior only reduced savings a few percent. In effect, the system is more robust in the presence of occupant window control and, along with MM4, appears to provide near-optimal performance even using very conventional control heuristics.

- A seasonal changeover point was observed for MM3 in Boulder at daily mean temperatures of around 15 °C. Since the observed changeover points align well with the intersection of the static and adaptive comfort envelopes, depicted in Figure 6.6, there is reason to believe that the exact point may be more a function of comfort standards than of the buildings themselves.

- Climate tests particularly in the cooler marine climates underscored that for successful HVAC interlock, all relevant space conditioning systems, including those not coupled to the ventilation system, should be disabled during NV periods. When MM3's perimeter baseboard heaters were incorporated into the interlock signal, San Francisco and Seattle cases were able to almost completely float during the cooling season period.

6.5.3 MM4

- MM4 solutions exhibited consistent night ventilation, a diurnal slab charging pattern, and ample daytime NV opportunities. Even using very lean, low-exergy cooling delivery, MPC was able to identify energy savings of in the range of 30–40% in adaptive cases.

- MM4 solutions demonstrated that the typical operation of radiant cooling systems during unoccupied periods is not necessarily optimal, and that briefly pulsing the slab throughout the occupied period may be a preferred approach.

- The relatively stable operative temperatures in the facility allowed for more latitude in opening windows without negatively affecting comfort.

- Interlock-enabled changeover operation of the DOAS system provided small an additional 20–40% HVAC energy savings compared to partial changeover cases. Absolute savings are small given the already reduced energy use of this system.

- As with all other cases, solutions seemed less sensitive to the sequencing of occupant-operated windows than to the operation of mechanical systems.

Chapter 7

Near-Optimal Supervisory Control for Select MPC Cases

Offline MPC simulation studies in MM buildings have demonstrated significant energy savings potential for model predictive supervisory control in MM buildings while maintaining acceptable thermal comfort according to prevailing standards. However, for reasons explained in the introductory chapters, there are currently many hurdles to implementing real-time or online MPC in live buildings, even once computational speed and other technical hurdles have been addressed. This chapter explores how closely we can approximate some of the strategies seen in the offline MPC results using more sophisticated heuristics and the rule extraction techniques discussed in previous chapters.

The MM2 and 3 buildings are used as demonstration cases in this chapter. The subsequent chapter investigates an implementation of extracted rules on a test facility very similar in system configuration to MM4.

7.1 Improved Heuristics

A common approach to analyzing and distilling the results of offline MPC simulation studies is to examine solutions for recognizable patterns and logic that resonates with one's understanding of building physics, then codify those patterns into supervisory rules. If successful, practical heuristics can result that approach optimal energy savings and comfort and that are still easily implemented and interpreted by building operators and engineers.

7.1.1 Adaptive Cooling/Heating Setpoint Reset Heuristic

An adaptive cooling and heating reset schedule was developed and applied to both MM2 and 3. The goal of the heuristic is twofold. It first seeks to allow zone temperatures to float within the ASHRAE 55 adaptive comfort window (80% occupant satisfaction bands) during warmer periods (running mean temperatures greater than 18 °C). Secondly, it attempts to take advantage of night-time economizer pre-cooling opportunities by suppressing cooling setpoints before occupancy, while still maintaining thermal comfort at the lower boundary of the comfort window at the beginning of occupancy. During other times, setpoints are simply set to default nighttime set-up (NSU) values. The basic logic is summarized in Figure 7.1.

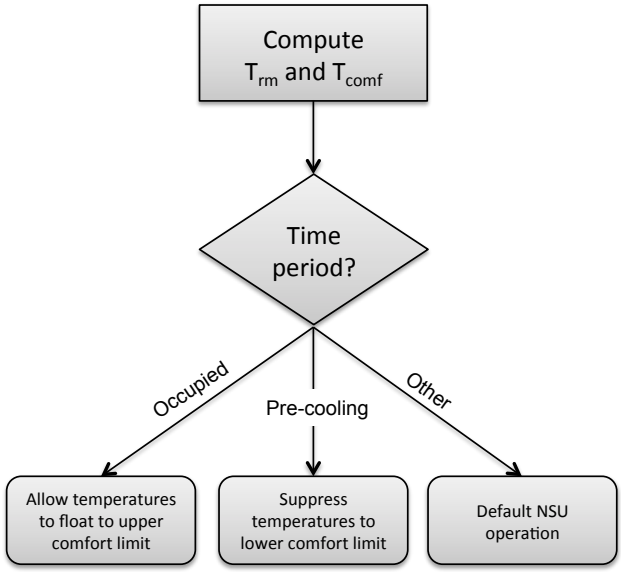


Figure 7.1: Flowchart of logic used for simplified adaptive setpoint reset algorithm.

The adaptive heating and cooling setpoint reset is loosely based on the recommendations of the EU-funded Smart Controls and Thermal Comfort (SCAT) project [71], which developed and prototyped a supervisory controller based on the adaptive thermal comfort criteria of the EN 15251 standard. For the purposes of this research, the ASHRAE 55 thermal comfort criteria were used instead. According to ASHRAE 55 and the underlying adaptive thermal comfort research of Brager and deDear [3, 18, 33, 34], the neutral comfort temperature—the operative temperature at which

occupants will, on average, deliver a neutral comfort vote on the ASHRAE scale—is defined as the following function of the running mean temperature (T_{rm})[1] :

$$T_{\text{comf}} = 0.31T_{rm} + 17.8. \tag{7.1}$$

Upper and lower thermal comfort limits during periods when $T_{rm} > 10$ °C are given by:

$$T_{\text{upper/lower}} = T_{\text{comf}} \pm 3. \tag{7.2}$$

The running mean temperature is an exponentially weighted average temperature, given by

$$T_{rm} = (1 - \alpha)T_{od,-1} + \alpha T_{rm,-1}, \tag{7.3}$$

where $T_{od,-1}$ is the average outdoor dry bulb temperature over the past 24 hours, $T_{rm,-1}$ is the running mean temperature from the previous day, and α is an exponential weighting term usually set to 0.8.

The logic in the adaptive reset schedule is that heating and cooling setpoints during the occupied periods should track the upper and lower bounds of the comfort window. To provide an extra margin of error and avoid overshoot, the temperature setpoints are offset by 0.5K from the boundaries. Thus, cooling and heating setpoints can be established by

$$\begin{aligned} T_{sp,\text{cool}} &= T_{\text{comf}} + 2.5 \\ T_{sp,\text{heat}} &= T_{\text{comf}} - 2.5. \end{aligned} \tag{7.4}$$

Recall that solutions for MM1 and 2 exhibited some mechanical pre-cooling during unoccupied hours. For unoccupied periods between midnight and one hour prior to occupancy, a pre-cooling

[1] Although ASHRAE 55 is typically drawn in terms of mean monthly temperature, discussions with deDear and Brager suggested that an alternate approach using running mean temperatures is also acceptable. The alternate approach is further documented in a 2006 position paper by deDear [32].

strategy has been loosely adapted from the foundational work of Keeney and Braun in which cooling setpoints are suppressed to the lower limit of the comfort window—in this case the ASHRAE 55 adaptive comfort window—until one hour prior to occupancy [59]. The one hour period is used as a warmup period to avoid any cool comfort violations during early occupancy. In this implementation, cooling setpoints are suppressed to the lower adaptive comfort boundary, with a 4K deadband enforced for heating setpoints:

$$\begin{aligned} T_{sp,\text{cool}} &= T_{\text{lower}} \\ T_{sp,\text{heat}} &= T_{sp,\text{cool}} - 4. \end{aligned} \tag{7.5}$$

During all other periods, cooling and heating setpoints follow NSU defaults. The daily parsing of the problem is illustrated in Figure 7.2. Note that since MM3 solutions exhibited less noticeable mechanical pre-cooling and since MM3 possesses the ability to night ventilate, its nighttime setpoints revert to NSU defaults.

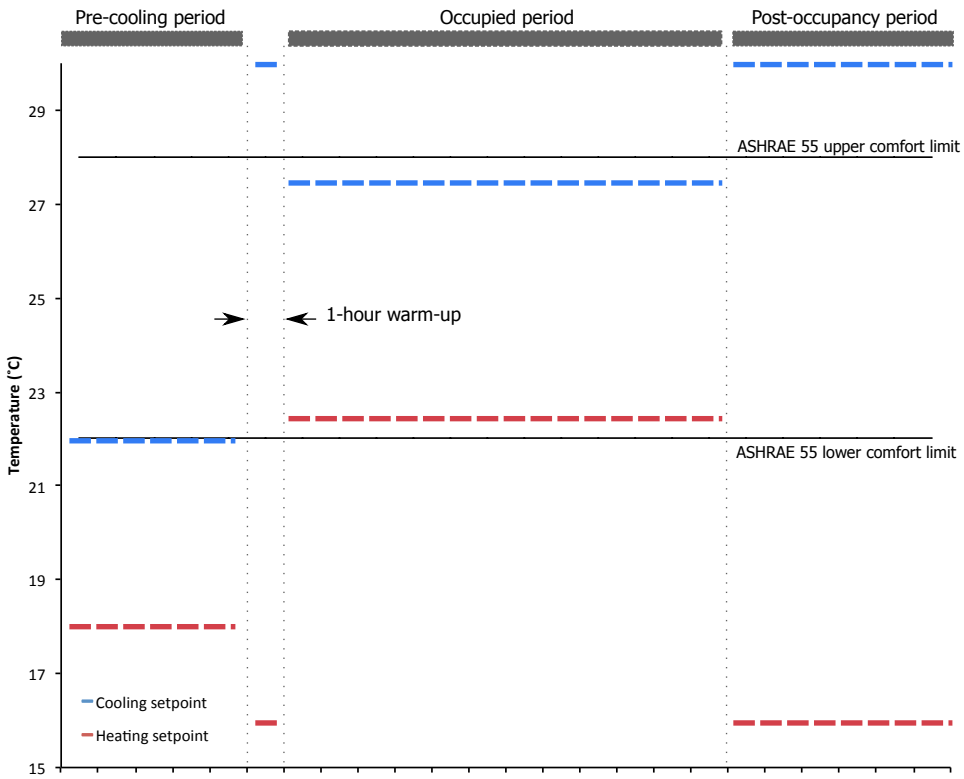


Figure 7.2: Problem parsing for the simplified adaptive comfort reset schedule.

7.1.2 Night Ventilation Heuristic

A night ventilation heuristic was adopted based on the offline MPC results for MM3, which allowed night ventilation during the unoccupied periods. Just like the results for MM3, the night ventilation heuristic takes priority over mechanical/economizer pre-cooling allowed by the adaptive reset schedule presented above. If night ventilation is operational, VAV dampers are shut. The night ventilation heuristic allows windows to open whenever they are capable of meeting the zone cooling setpoint, established by the adaptive setpoint reset heuristic, shown in Figure 7.2. A proportional controller with a 2K proportional band governs the individual opening of windows.

7.1.3 Dealing With Occupants

Recall from the offline MPC study results, there were clear reasons to discourage occupant window opening at certain times of year based on overcooling concerns, and an optimal seasonal changeover point exists at which time the adaptive comfort-penalized solutions begin to outperform the base case (Figure 6.8). Based on observations presented in Figure 6.6, we also saw that this changeover point aligns fairly well with regions of the adaptive comfort standards that allow warmer comfort temperatures than would be allowed by static comfort guidelines. Based on these observations, the simple heuristics and extracted rules only allow occupants to open windows when running mean temperatures are above 18 °C. Even though a seasonal changeover point of 15 °C mean daily temperature was observed in the original results, the 18 °C running mean was used to ensure that weather had completely shifted into the cooling season before allowing window openings. In an actual building, such a policy would likely be enforced by a building engineer, facilities management, etc.

7.2 Rule Extraction

7.2.1 Adaptation of CART Methodology

Rule extraction was conducted for the MM2 and MM3 cases in a manner consistent with the methods described in Chapters 3 and 5. Prior results led to the conclusion that, despite the oftentimes improved performance of adaptive boosting models, CART-based rules were more comprehensible and easier to implement. Therefore, CARTs were used exclusively to mine the MM2 and MM3 optimal training sets for setpoint and night ventilation patterns.

As noted above for the treatment of simplified heuristics, the rule extraction process makes a distinction between swing and cooling season control. This split in training datasets derives directly from the offline MPC results which demonstrated a changeover point in the summer after which natural ventilation and consideration of adaptive comfort become optimal from an energy use standpoint. Swing season rules were trained on offline MPC results from a May–June weather period incorporating static comfort considerations from ASHRAE 55. Cooling season rules were trained on a July–August period using the adaptive provisions of ASHRAE 55 for comfort considerations. The comfort performance of the two extracted rules are subsequently evaluated on different criteria.

In previous examinations, the control parameter or rule response was always a binary variable: the building-wide window opening signal. In the case of MM2 and MM3, setpoints must also be dealt with. In general, setpoints would be considered continuous variables, but because of the discretization chosen for the optimization problem (setpoints were allowed in increments of 2K to reduce the size of the decision space), only about a half dozen discrete setpoint values existed in the training set. This means that, from the standpoint of the CART methodology, setpoint responses could either be treated as discrete classes or continuous numbers, enabling the use of either classification or regression trees. Whereas the value assigned to a classification tree represents the majority class of points present in that node, the value of a node in a regression tree is the mean of points residing in that node.

As discussed in Chapter 3, the CART algorithm seeks to grow trees that minimize the cost-

complexity parameter, R_α, which is a combination of the tree's misclassification error and its complexity (number of nodes). In regression trees, the misclassification error is replaced with a normalized mean squared error estimate which derives directly from classical analysis of variance:

$$RE = \frac{1}{N\text{var}(z)} \sum_{n=1}^{N} (z_n - \hat{z}_n)^2. \tag{7.6}$$

N is the number of points in the dataset, var(z) is the variance of the training set, z_n is the n^{th} response from the training set, and $\hat{z}_n$ is the n^{th} prediction of the tree. The resulting RE is then analogous to the misclassification error and can be used for comparison. Consequently, $1 - RE$ provides the R^2 value commonly used as a goodness of fit measure in classical least squares regression.

7.2.2 Predictor Set Considerations

In results presented in previous chapters, strong rule performance was achieved using relatively simplistic predictor sets. Window opening rules were based on current timestep information, not trend variables or forecasts, and predictor variables were basically constrained to ambient and zone conditions. Obviously the objective function evaluated by the optimizer "sees" more than just the immediate zone conditions, as MPC decisions in a receding horizon problem are impacted by past thermal states and predicted conditions as well. Thus, both trend and forecast variables were incorporated into the predictor set. Only very simple forecast values like daily high/low temperatures were used based on observations from past MPC research, and perfect prediction was assumed.

Given the consideration of global cooling setpoints and the inherent interaction between those setpoints and any potential occupant window opening behavior at the zone level, a variety of airside system states were ultimately included in the predictor sets as well. This was based on observations that optimal setpoint solutions sought to minimize cooling coil loads during potential occupant window opening periods. Loads on the cooling coil are themselves sensitive to a variety of system states, including the enthalpy of both outdoor and return air, the outdoor air fraction, the

supply air flow rate, and the supply air setpoint. When these factors were not taken into account, misclassification errors and relative error values ranged from 0.5 to 0.8. Once additional states were incorporated, *RE* values as low as 0.2 could be achieved.

Also recall that in proof-of-concept formulations, comfort was considered neither in the objective function used in offline MPC nor in the evaluation of rule performance. Comfort penalties were imposed in deriving the MM2 and 3 training sets, as this is a paramount real world constraint, and thus comfort considerations are crucial to rule performance here as well. Adaptive comfort parameters are easy to incorporate as predictors, as they derive from a running mean temperature that can be readily predicted for each day. A summary of predictors is provided in Table 7.1.

7.3 MM2 Results

7.3.1 Boulder, CO

CARTs (both classification and regression variants) were developed for swing and cooling season operation for MM2. Swing season training sets were based on a month of partial changeover results (May 11 to June 11, Boulder-Longmont TMY3 weather) using the ASHRAE 55 static comfort penalty. Cooling season sets were based on a month of partial changeover results (July 13 to August 13, Boulder-Longmont TMY3 weather) with the ASHRAE 55 adaptive comfort penalty. In both seasons, the first week of results was discarded to allow non-optimized thermal states to recede in the thermal history. In other words, we wish to examine how the optimizer behaves in the presence of "already-optimal" thermal states, not the suboptimal thermal states that may linger during the warmup period. The remaining data was split into a training set, comprising the first two thirds of results, and a testing/cross-validation set, comprising the remainder. CARTs were then trained, pruned, and cross-validated for performance in open and closed loop tests.

The inclusion of airside system states—namely flow rates and psychrometric conditions—had significant impact on performance, particularly for regression trees and especially during the cooling season, where *RE* values were lowered from about 0.5 to about 0.1. Error rates for both tree types

Table 7.1: Predictors Examined for MM2 and 3 Rules

Ambient and Zone States			
T_{oa}	Outdoor dry bulb temperature	θ_{wind}	Wind direction
RH_{oa}	Outdoor relative humidity	v_{wind}	Wind speed
h_{oa}	Outdoor air enthalpy	T_{upper}	Adaptive comfort upper limit (ASHRAE 55)
T_{zn}	Zone mean air temperature	T_{lower}	Adaptive comfort lower limit (ASHRAE 55)
RH_{zn}	Zone relative humidity	T_{rm}	Outdoor running mean temperature
z_{sp}	Cooling setpoint value	I_{hor}	Global horizontal insolation
z_{vent}	On/off state of night ventilation	ΔT_{oa-rm}	Difference between outdoor and running mean temperatures
ΔT_{zn-rm}	Difference between zone and running mean temperatures	H	Hour of day
Air System States			
T_{ra}	Return air temperature	T_{ma}	Mixed air temperature
h_{ra}	Return air enthalpy	h_{ma}	Mixed air enthalpy
$\dot{m}_{ra}$	Return air mass flow rate	$\dot{m}_{ma}$	Mixed air mass flow rate
$\dot{m}_{oa}$	Outdoor air mass flow rate	ϕ_{oa}	Outdoor air fraction
$T_{sp,sa}$	Supply air temperature setpoint	$\dot{Q}_{\text{coil}}$	Cooling coil load
ΔT_{oa-sa}	Difference between outdoor and supply air setpoint temperatures	Δh_{oa-ra}	Difference between outdoor and return air enthalpy
24-Hour Sliding Window Trend Information			
$T_{oa,\text{max/min}}$	Max/min outdoor temperatures	$RH_{zn,\text{avg}}$	Average zone relative humidity
$T_{oa,\text{avg}}$	Average outdoor temperature	$T_{rm,-1}$	Prior day running mean temperature
$RH_{oa,\text{avg}}$	Average outdoor relative humidity	$T_{\text{upper},-1}$	Prior day adaptive comfort upper limit[1]
$I_{\text{hor,sum}}$	Prior day total insolation	$T_{\text{lower},-1}$	Prior day adaptive comfort lower limit[1]
$T_{zn,\text{max/min}}$	Max/min zone temperatures	$Q_{\text{coil,sum}}$	Sum of cooling coil loads
ΔT_{oa-rm}	Max/min ceiling surface temperatures		
Forecast Variables[2]			
$T_{oa,\text{max/min},+i}$	Forecast high/low outdoor temperatures	$T_{\text{swing},+i}$	Forecast temperature range
Proxy Variables			
$N_{\text{vent},-1}$	Prior day total hours ventilation operation		

[1]Only used in cooling season cases with adaptive comfort penalty.

[2]Forecasts are provided i days ahead. A 0 corresponds to the forecast for the current day.

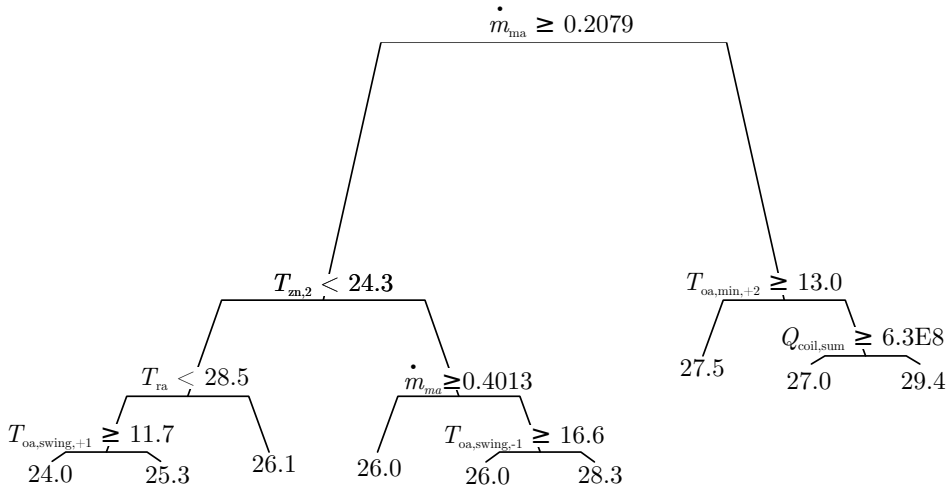
and both seasons are provided in Table 7.2.

Table 7.2: Open Loop Performance: MM2 CARTS

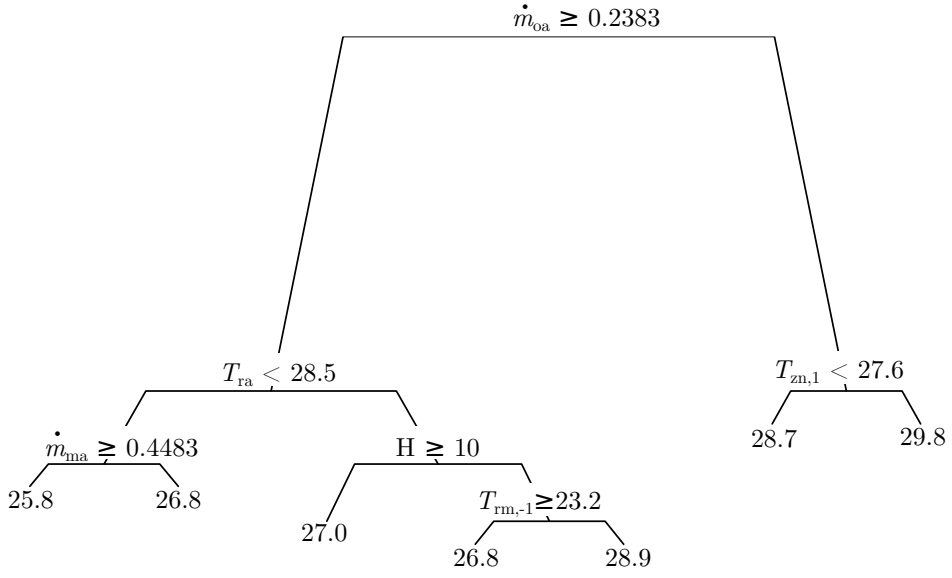
	Swing Season	Cooling Season
Classification R	0.36	0.51
Regression RE	0.16	0.09

Given the relatively large number of classes present and the fact that one does not need to replicate the exact setpoint chosen by the optimizer to decimal place accuracy, regression trees were adopted for closed loop testing and implementation. The swing and cooling season rules are presented in Figures 7.3(a) and 7.3(b). Note that the highest ranking splits are mass flow rates for mixed air and outdoor air, respectively. These choices are likely a proxy for occupancy, since occupancy also ranked highly among competing splits for these nodes. Thus, if we consider the right hand sides of the trees as unoccupied periods, we see that the rule basically pursues a nighttime setup strategy, with some variation depending on forecasts and past loads. The majority of the nodes are on the occupied or left hand side of the tree, where setpoints can be set anywhere within a range of 24 to 28.3 °C, depending on a variety of zone and system states. Trend ($T_{oa,\text{swing},-1}$) and forecast variables ($T_{oa,\text{swing},+1}$) are involved at lower levels of the tree, indicating that a strong predictive element was not discovered through the CART analysis. The rule allows setpoints to creep higher (to a max of 28.3 °C) when zones begin to rise in temperature above 24.3 °C. One interpretation of the logic is that, when zone temperatures rise above this point, this indicates a period of window openings when the optimizer allowed setpoints to rise. In this way, the CART distinguishes between occupied and unoccupied period setpoints.

Closed loop performance tests were conducted for both seasons, and a summary of key performance statistics are provided in Table 7.3. The original base, reference, and optimal cases from the MPC runs are included, as well as results using a simplified heuristic and the extracted rule. Comfort "severity" is presented here as the sum of all deviations outside the allowed comfort boundaries, for all zones. Several observations are obvious. First, the simplified heuristic developed on very



(a) Swing season rule



(b) Cooling season rule

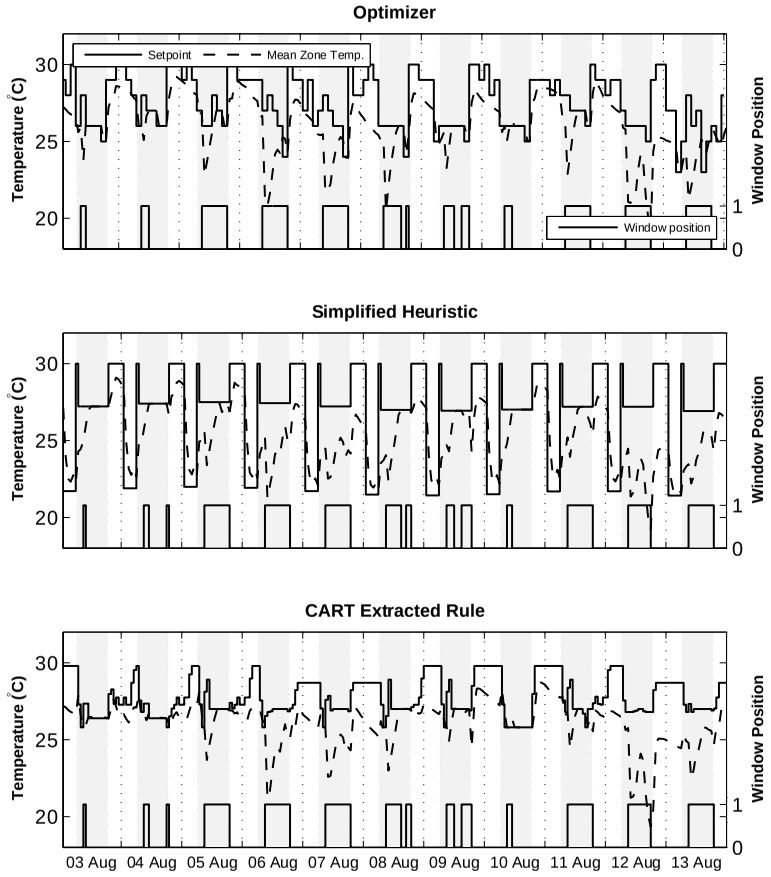
Figure 7.3: Dendrograms of the swing and cooling season CARTs extracted from MM2.

reasonable assumptions does not result in perfect comfort and resides about halfway between the reference and optimal cases in terms of its energy use. The extracted rule achieves similar comfort as the simplified heuristic, but with much deeper energy savings. The rule actually achieves lower energy use than the optimizer in both cases but would have much higher objective function values due to poorer comfort, particularly in the swing season. The range of 80% acceptable comfort conditions is much narrower in the swing season, because static comfort model is applied during this period when NV is generally not utilized. The response to the near-optimal control during the swing season is, not surprisingly, much more sensitive to under- and over-cooling from a comfort standpoint. Results in the cooling season are, however, remarkably close to optimal and with tolerable comfort violations, as this case is evaluated under the more forgiving adaptive comfort provisions of ASHRAE 55.

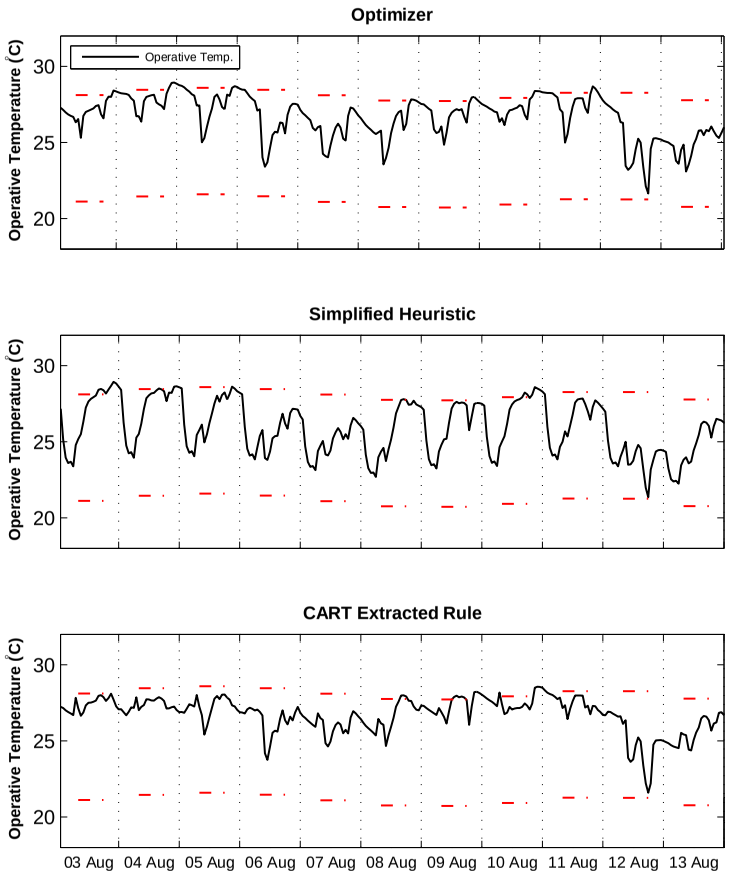
Table 7.3: Closed Loop Performance: MM2 CART

	Swing Season: June 1 - 11				**Cooling Season: August 3 - 13**			
	Energy Use (kWh)		Comfort		Energy Use (kWh)		Comfort	
	Electric	Gas	Violations (hours)	Severity (PMV-hours)	Electric	Gas	Violations (hours)	Severity (K-hours)
Base Case	342	28	0	0	867	0	0	0
Reference Case	389	158	0	0	786	79	9	0
Optimal Case	323	27	7	23	470	71	4	0
Simple Heuristic	404	48	53	42	527	139	14	14
Extracted CART	249	62	67	24	433	69	15	9

Figures 7.4(a) and 7.4(b) provide a comparison of the optimal case, simplified heuristic, and extracted rule in operation, illustrating time series of setpoints, occupant window openings, and zone operative temperatures in a comfort envelope. With regards to comfort, the optimizer solution expends slightly more effort to discourage hot comfort violations on certain days (e.g. the sudden drop in setpoint toward the end of the occupied period on August 8). The CART rule is not nearly as sensitive, but still only 15 relatively minor comfort violations over the course of 11 days.



(a) Cooling season setpoints



(b) Cooling season comfort

Figure 7.4: Comparison of MM2 setpoint reset schedules and comfort for optimal, simple heuristic, and extracted rule cases. Occupant window positions are shown on the secondary axis in the setpoint charts.

7.3.2 Cross-Climate Robustness Tests

Given that extracted rules have been trained on a specific building in one climate, questions arise as to their robustness when applied to other cases. Two climate comparisons were conducted on the CART derived from MM2, one in Las Vegas and the other in San Francisco, both significantly different in the loads they present to the building. The results harken back to the observations made in Chapter 5 in which a cooling season rule was applied to a portion of the heating season: extracted rules are only as robust as the dataset upon which they were trained. A rule geared toward the high diurnal temperature swings and solar gains of Boulder's cooling season should behave differently when applied—even in the same building—in San Francisco's cooler, cloudier, and more stable marine climate.

Table 7.4 summarizes the findings from the original MPC runs, a simplified rule, and the extracted rule. Interestingly, performance of both the simple and extracted rule are very close to optimal, because occupants infrequently open windows in these cases, eliminating the possibility of the cooling system having to "fight" loads introduced by window openings. In the San Francisco case, low running mean temperatures result in our seasonal changeover point of 18 °C never being reached, and occupants are never allowed to operate windows. Implementing a slightly different policy, say allowing occupants to operate windows when $T_{rm} > 15\,^{\circ}\mathrm{C}$, changes performance dramatically, increasing gas usage to 1,300 kWh during the test period (worse than the reference case). Electricity use—which really is the purview of the extracted rule—remained relatively constant, dipping by about 4% with the new window operation strategy.

The success of the MM operational strategy, at least in San Francisco, hinges greatly on occupant behavior and when occupants are allowed to adapt through window openings. Naturally this has implications for future research, which will be discussed later, but of importance here is that the extracted setpoint reset curves for this building maintained similar energy performance compared to optimizer savings, under three different climate conditions (90–110% of optimizer energy use, with somewhat degraded comfort). The setpoint sequences produced by the extracted

Table 7.4: Closed Loop Performance: MM2 Cross-Climate Comparison

	Las Vegas				**San Francisco**			
	Energy Use (kWh)		Comfort		Energy Use (kWh)		Comfort	
	Electric	Gas	Violations (hours)	Severity (K-hours)	Electric	Gas	Violations (hours)	Severity (K-hours)
Base Case	855	0	0	0	208	0	0	0
Reference Case	855	0	0	0	176	1,162	0	0
Optimal Case	663	0	0	0	194	0	0	0
Simple Heuristic	645	12	0	0	248	7	0	0
Extracted CART	708	0	0	0	185	0	18	26

rule—although they by no means reproduce the optimal reset curves—bear a higher correlation to the optimizer reset curves with lower relative error compared to the simplified heuristic, as presented in Table 7.5.

Table 7.5: MM2 Reset Schedule Correlations with Optimal Results

	Correlation		Relative Error	
	Simple Heuristic	Extracted CART	Simple Heuristic	Extracted CART
Boulder - Swing Season	0.17	0.39	2.93	0.97
Boulder - Cooling Season	0.01	0.37	3.82	1.00
Las Vegas - Cooling Season	-0.13	0.26	4.15	1.24
San Francisco - Cooling Season	0.08	0.51	2.43	1.21

7.4 MM3 Results

7.4.1 Boulder, CO

Similar to MM2, CART-based rules were developed based on MPC results for MM3. In the case of MM3, the optimizer was used to coordinate cooling setpoints and night ventilation in the presence of mean occupant window opening behavior. This presented a more challenging test for rule extraction due to the interaction between the two types of decision variables. As with MM2, CARTs were trained for the swing (May 11 to June 11) and cooling seasons (July 13 to August 13) based on a two-week training set and were then tested on a 10-day cross-validation set. Training data for the swing season was based on the ASHRAE 55 static comfort penalty solution (recall that static comfort-penalized solutions generally fared better than adaptively-penalized ones during swing season weather for reasons illustrated in Figure 6.6); data for the cooling season was based on the ASHRAE 55 adaptive comfort penalty.

Both the setpoint reset and night ventilation rules were initially trained on the same predictors presented in Table 7.1. Open loop responses for the ventilation rule were amazingly close to optimal—more so than in any prior case—with misclassification errors of less than 1% under cross-validation; however, closed loop tests yielded effectively no night ventilation response whatsoever. Naturally, after thoroughly debugging the rule implementation code and finding no faults, a search for explanations began.

Setpoint and ventilation strategies exploited by the optimizer are highly interdependent, and one would naturally expect that near-optimal rules should reflect this fact. To this point, an examination of the final CART predictor sets for both rules revealed that airside state variables (psychrometric conditions in the air handler) ranked highly in both rules. As with rules presented for MM2 or those presented in Chapter 5, those same states have feed-forward effects, biasing a rule's operation in its next iteration. However, the interdependence of the two strategies in MM3 leads to an amplification of building responses and a loss of stability in the closed loop case.

The solution was to eliminate some of the tight physical coupling between the rules by limiting the scope of their predictor sets. Rather than basing the ventilation rule on the same potential variables as the setpoint reset rule, only zone and ambient conditions were allowed as predictors. The setpoint rule still included consideration of system states. In open loop testing, this slight alteration in predictor sets had very minor impacts on the ventilation rule (recall in Chapter 5 that ventilation rules could be based solely on zone and ambient conditions and achieve very high accuracy) in open loop tests, but resulted in much more recognizable night ventilation patterns in closed loop tests, suggesting that the looser coupling between the rules in this case was responsible for the improved performance.

Table 7.6: Scope of MM3 Rule Extraction Training Sets

	Setpoint Rule	Ventilation Rule
Ambient Conditions	•	•
Weather Forecasts	•	•
Zone Conditions	•	•
Airside System Conditions	•	

As with MM2, a regression tree was ultimately deemed superior to a classification approach for setpoint schedules (Table 7.7), with acceptable RE values of about 0.2. A binary classification tree was adopted for the ventilation rule, with misclassification errors of about 5%.

The setpoint reset rule relied somewhat more on system states and forecast information than

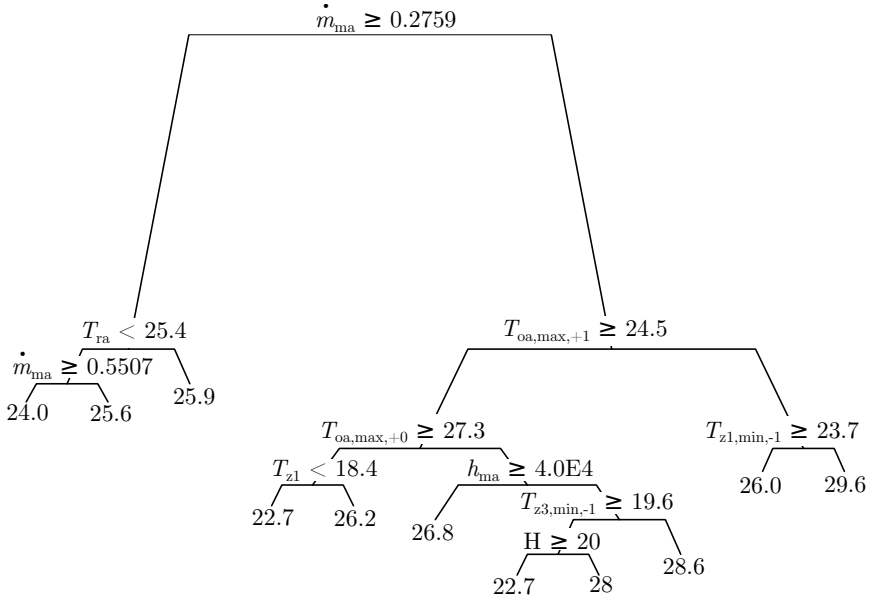
Table 7.7: Open Loop Performance: MM3 CARTs

	Swing Season	Cooling Season
Classification R	0.30	0.50
Regression RE	0.13	0.18

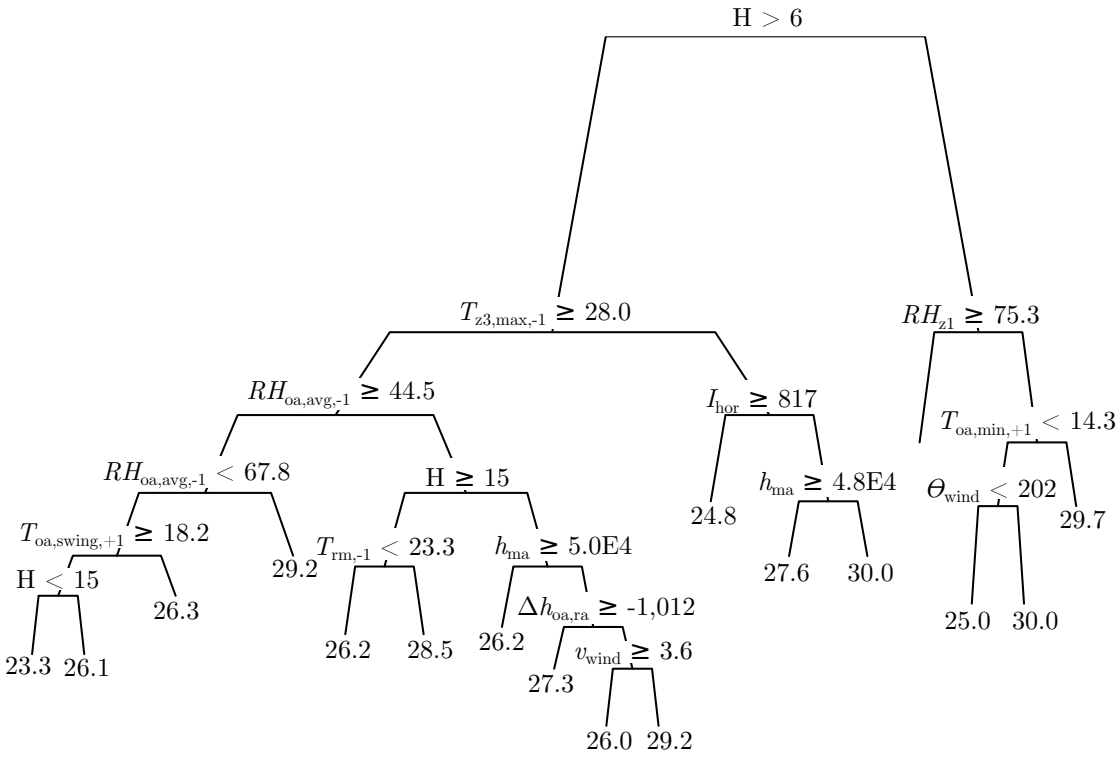
the ventilation rule. Hour of the day was the primary split in the cooling season case, splitting the tree into occupied and pre-occupancy modes. In MM2, the same effect was accomplished through airflow measurements. In actual implementation, it would be much more practical to base these splits on known or expected occupied periods rather than a direct or proxy occupancy measurement, but the "raw" result of the CART algorithm is used here for demonstration purposes. Unlike the MM2 reset curve, the MM3 CART allowed setpoints as high as 30 °C during occupied periods under certain conditions. The rule also incorporated some forecast (e.g. max/min temperatures) and trend information as predictors. Dendrograms for the swing and cooling season are provided in Figures 7.5(a) and 7.5(b).

Ventilation rules were based solely on ambient and zone states, so outdoor air temperatures and zone temperatures dominated. In the cooling season, the CART easily identified the nocturnal nature of the night ventilation strategy and used the occupancy flag as the primary split; in the swing season, this behavior had to be enforced by explicitly training the rule on data from unoccupied periods only. Dendrograms for the swing and cooling season ventilation rules are provided in Figures 7.6(a) and 7.6(b). Ventilation rules, like the cooling setpoint rule for MM2, were formulated entirely on current conditions; trend and forecast values were not identified as significant predictors.

Closed loop performance followed a similar seasonal trend to MM2. In the cooling season, extracted rules were able to maintain over 90% of optimizer savings with some additional comfort violations, whereas the "improved" setpoint reset and night ventilation heuristics developed based on observations of MPC solutions actually consumed more energy with far worse comfort than the original reference cases (Table 7.8). An examination of time series results from the MPC

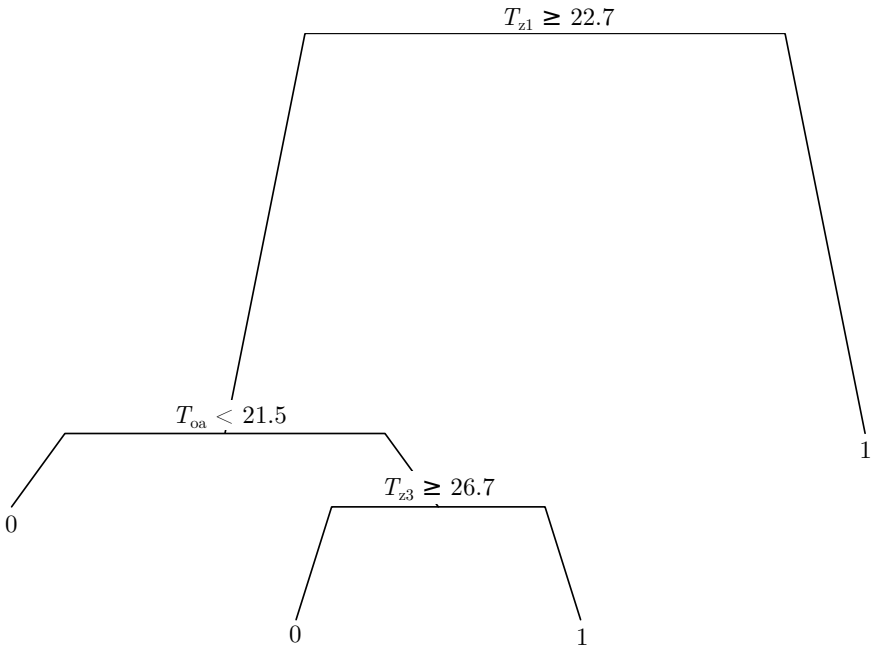


(a) Swing season

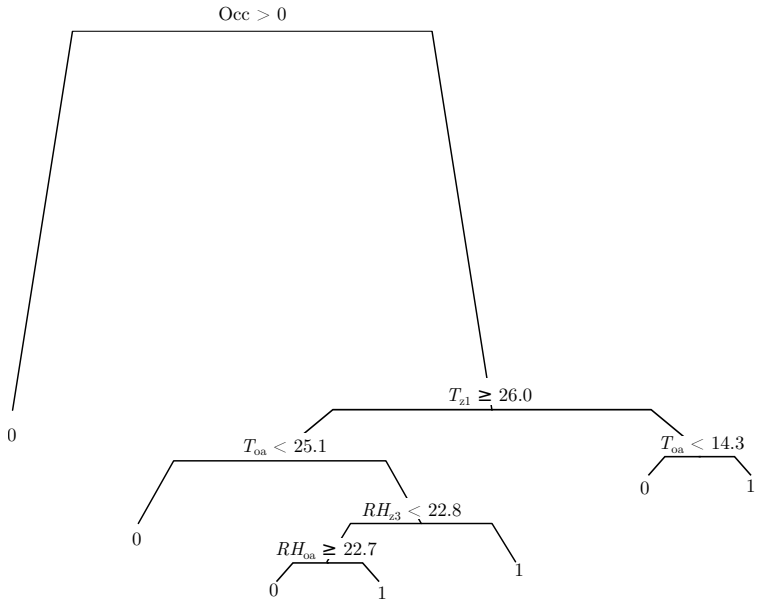


(b) Cooling season

Figure 7.5: Dendrograms of the MM3 setpoint reset rules for the swing and cooling seasons.



(a) Swing season (unoccupied periods)



(b) Cooling season

Figure 7.6: Dendrograms of the MM3 night ventilation rules for the swing and cooling seasons.

solution, heuristic, and extracted rule during the cooling season (Figure 7.7) demonstrates the night ventilation patterns generated by the CART rules adhere much more closely to the original MPC solutions compared to those generated by the heuristic. Cooling setpoints do not exactly track optimizer sequences (opportunities for nighttime setups are missed on several days), but one still observes that the CART manages to explore the full range of setpoints observed in the MPC results on roughly the same diurnal cycle. Swing season results were not as promising, with extracted rules missing the mark somewhat on comfort (40 hours of violations) sometimes due to late-afternoon overheating and more frequently because of cooler temperatures during early occupancy. As with the swing season rules for MM2, these comfort issues during the swing season are attributed to the more restrictive static comfort envelope employed. The extracted heuristic has a very small margin to improve on the reference case because the optimizer itself yielded marginal additional savings, and consequently in the cooling season, the reference, optimal, and extracted rule performance is extremely similar.

Table 7.8: Closed Loop Performance: MM3 CARTs

	Swing Season: June 1 - 11				**Cooling Season: August 3 - 13**			
	Energy Use (kWh)		Comfort		Energy Use (kWh)		Comfort	
	Electric	Gas	Violations (hours)	Severity (PMV-hours)	Electric	Gas	Violations (hours)	Severity (K-hours)
Base Case	253	26	4	4	500	1	0	0
Reference Case	34	591	2	38	129	241	11	14
Optimal Case	174	44	2	2	120	262	6	1
Simple Heuristic	227	84	66	28	209	313	71	171
Extracted CART	185	137	33	15	120	279	15	16

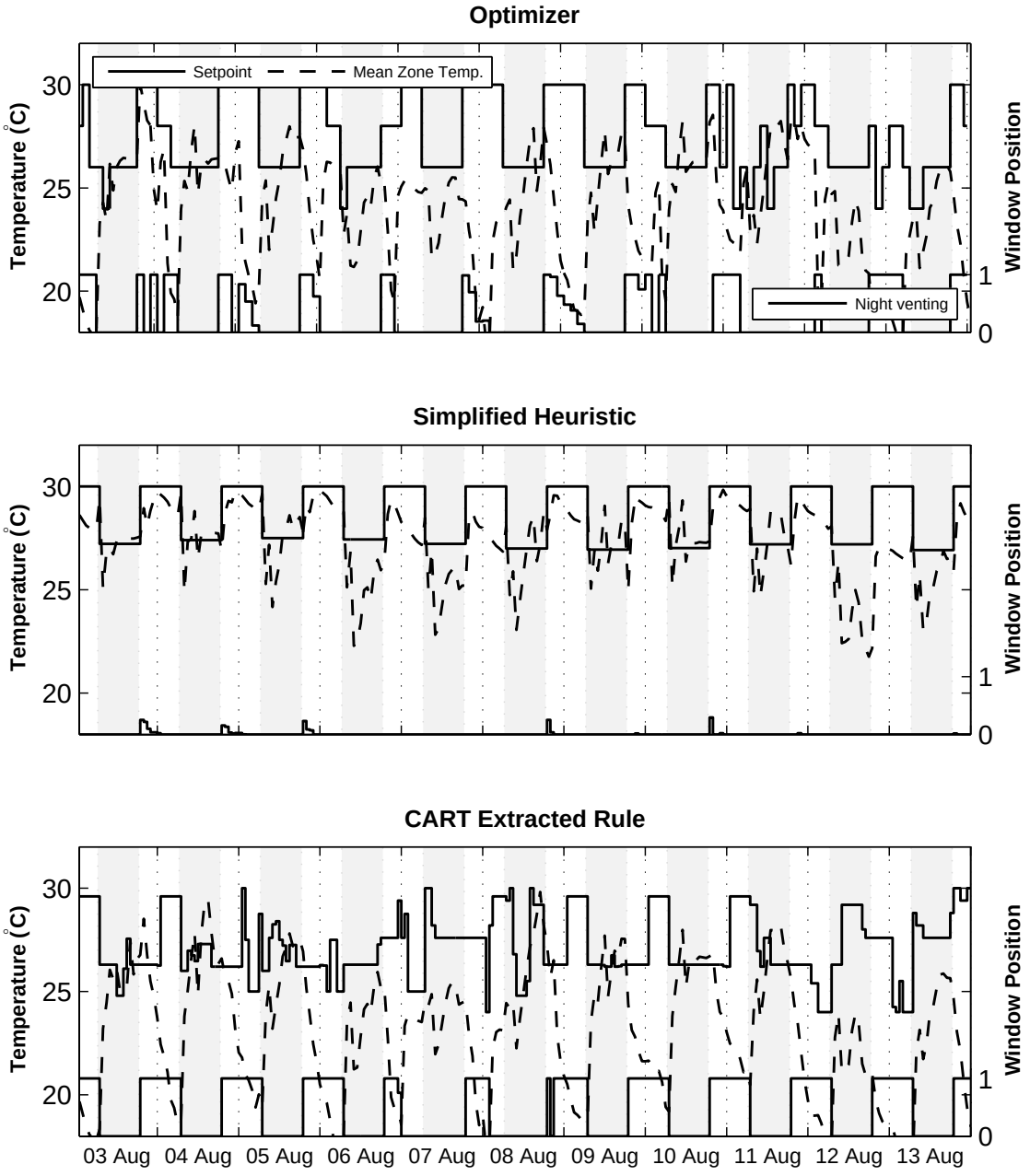


Figure 7.7: MM3 cooling setpoints and night ventilation openings during the cooling season cross-validation period.

7.4.2 Cross-Climate Robustness Test

A cross-climate test was conducted for MM3 in the same manner as MM2 to examine the rule robustness under different ambient conditions and loads. The results, presented in Table 7.9, suggest that the loosely coupled rules for cooling setpoints and ventilation applied in MM3 (to prevent degradation of performance in the closed loop scenario) are somewhat more sensitive to the ambient conditions under which they are trained and applied. Note that the energy performance of the extracted rules, although sometimes an improvement over the heuristics, is often not noticeably better than the base or reference cases. In these cases, the optimizer's savings are far harder to reproduce, regardless of technique. We saw a similar sensitivity to ambient conditions in Chapter 5, in which a cooling season ventilation rule was applied during the heating season, resulting in large spikes in gas use and greater discomfort.

Table 7.9: Closed Loop Performance: MM3 Cross-Climate Comparison

	Las Vegas				**San Francisco**			
	Energy Use (kWh)		Comfort		Energy Use (kWh)		Comfort	
	Electric	Gas	Violations (hours)	Severity (K-hours)	Electric	Gas	Violations (hours)	Severity (K-hours)
Base Case	714	0	0	0	92	3	73	216
Reference Case	620	0	0	0	9	1,560	0	0
Optimal Case	618	0	0	0	129	24	0	0
Simple Heuristic	611	0	77	218	92	3	73	224
Extracted CART	763	0	2	1	10	1,682	0	0

In MM3, however, there is reason to believe that there are additional influences. This case has two separate rules that both depend to some extent on ambient conditions, the night ventilation rule to a greater extent. The night ventilation rule also affects many of the system states upon which the setpoint rule depends, including return air temperatures and coil loads, both of which are second tier splits in the cooling season rule. In this sense, the application of the combined rules under significantly different ambient conditions is likely to result in performance differences that cascade up through the ventilation rule to the setpoint rule. Note in Figure 7.8 the much larger

disparities between both setpoints and night ventilation sequences between the MPC solution and extracted rules. The CART and simplified heuristic both revert to NSU operation and avoid night ventilation altogether, explaining why solutions for both are so similar to the base case building.

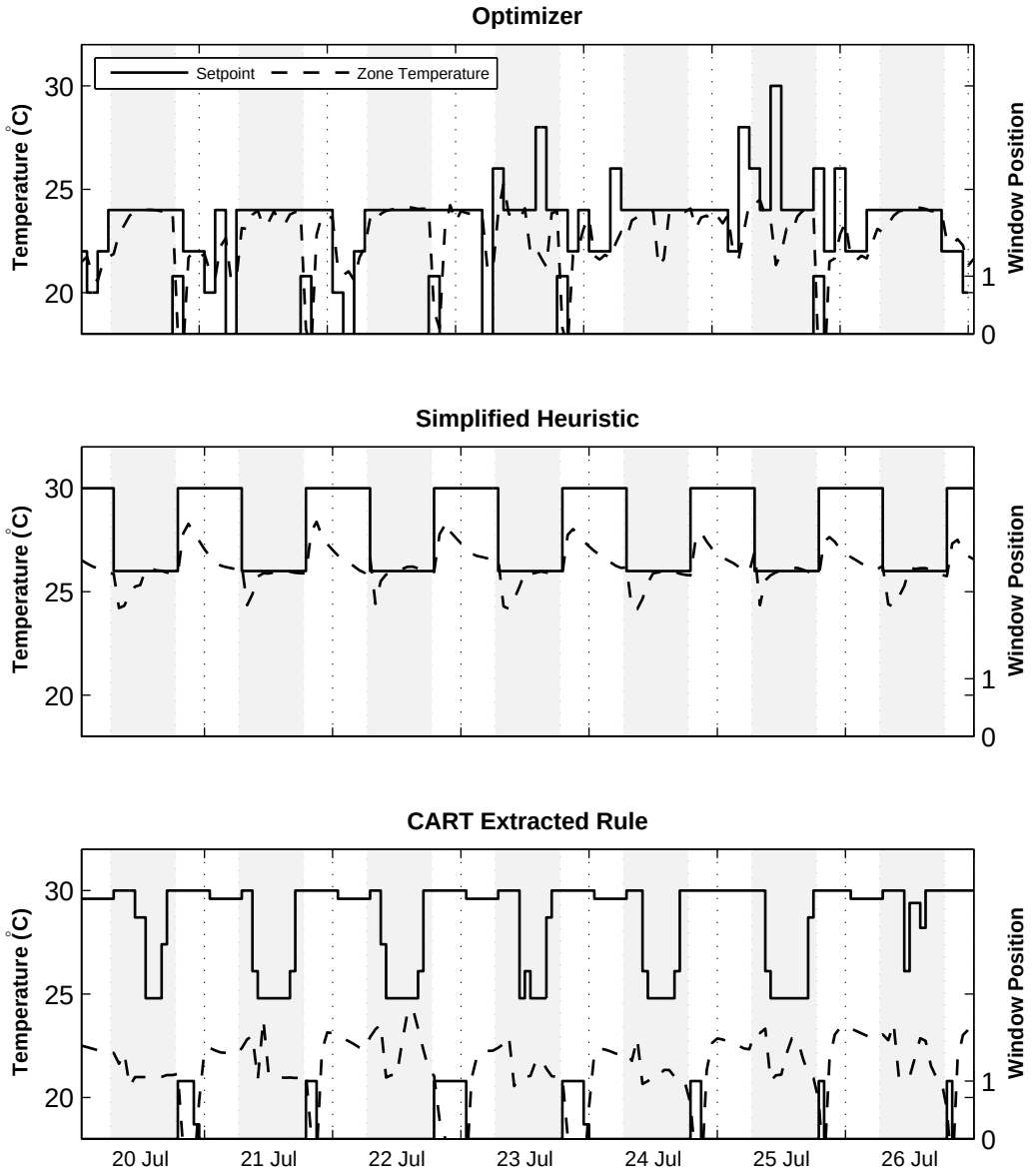
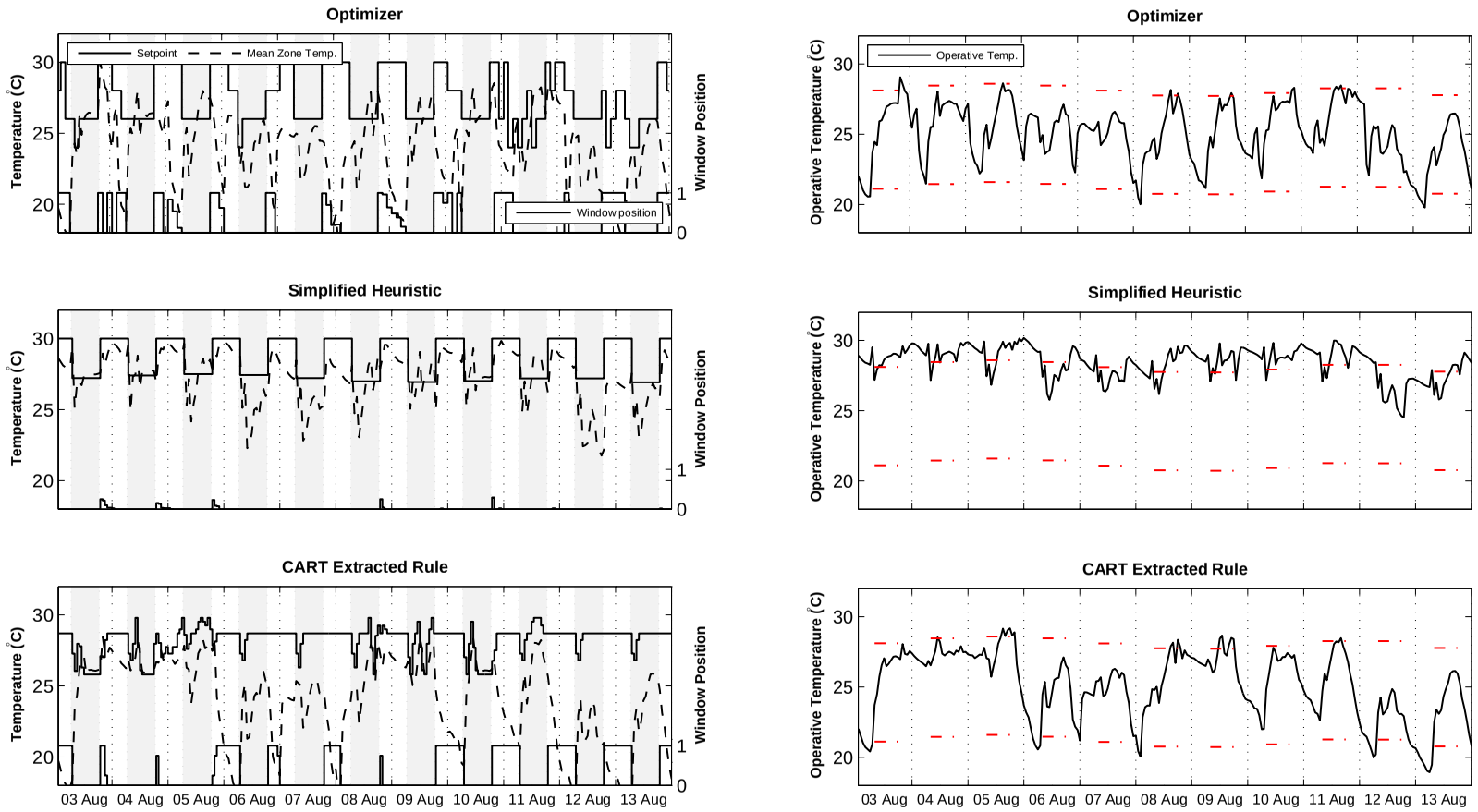


Figure 7.8: Cooling setpoints and night ventilation openings for MM3 under San Francisco weather conditions.

7.4.3 Cross-Building Robustness Test

It is also of interest to examine the performance of extracted rules applied to different building systems. How, for example, would MM3 perform if, instead of its own setpoint reset rule, it used the one trained on MM2? Would the MM2 rule be able to maintain reasonable setpoints and comfort given the different airside systems present in MM3 (a UFAD distribution scheme, with perimeter heating rather than a VAV terminal reheat coil; interlock between the HVAC system and occupant-operated windows; night ventilation capability)? A test was performed by substituting the MM2 setpoint reset CART into MM3, leaving all other elements of the BAS unchanged, including the extracted night ventilation rule. The resulting sequence of setpoints (Figure 7.9(a)) clearly departs from the diurnal patterns seen in the original MPC solution; the heuristic provides a much closer match to the original setpoints, as would be expected since the setpoint patterns from the original MM2 and 3 MPC cases were noticeably different. From a statistical standpoint, the weak correlation between near-optimal and optimal setpoints also decreased slightly. Due to the overwhelming influence of night ventilation on the energy balance and the skill with which the extracted rules reproduce the night ventilation sequence, there are minimal energy and comfort consequences, yet the underlying point remains: the application of the MM2-trained rule to MM3 resulted, not surprisingly, in noticeably different setpoint sequences. Obviously care must be taken in generalizing extracted rules, just as one would take care to generalize on the results of one MPC solution.



(a) Setpoints

(b) Comfort

Figure 7.9: Setpoints and zone temperatures in the ASHRAE 55 adaptive comfort window for MM3 using the setpoint reset rule from MM2.

7.5 Cross-Validation and Seasonality

Supervised learning approaches all involve the training of an inferred function on a curated or supervised training set. The choice of that training set can influence the resulting function or model, so it is important to select training sets appropriately. Throughout this research, a standard process of cross-validation has been used to test the performance of extracted rules under conditions that were not present their training sets. A coarse cross-validation process has been employed in which the first two weeks of offline MPC results are used for training and the last week for cross-validation/testing. The one-third:two-thirds split employed is a common rule of thumb seen in the literature (for example, in [21]). This training process also assumes that the logic harnessed by the optimizer to minimize energy costs is loosely stationary in time—that is, the process remains constant throughout the swing season or cooling season.

This approach easily raises a couple of questions. First, how would the rules differ if we had simply rearranged the training and cross-validation weeks? Would we see similar performance and rule structure, or would this yield distinctly different rules? Second, what if the stationarity assumption does not hold? What if the process being approximated by the rule changes in fundamental ways over time, exhibiting seasonal behavior? The following sections provide illustrative examples for addressing these questions, although deeper investigations are required in future work.

7.5.1 Improving Cross-Validation Procedures

More rigorous cross-validation processes exist that can help illuminate the first question above and determine how sensitive rules are to the specific parsing of their training sets. In the process used, three weeks of data were split into a two-week training set and a one-week cross-validation set. Taking the case of the MM2 setpoint rule as an example, one could have just as easily trained the rule on the last two weeks of data, cross-validating on the first; or testing could have been conducted on the middle week, splitting the training set in two.

Comparing these different permutations of the training set, one sees that the resulting rules

can still have similar structure. CARTs were grown on these different parsings of training data to a uniform depth of five layers. Dendrograms illustrating structural similarities are shown in Figures 7.10 and 7.11. Note that the leading split in each case is still the outdoor air mass flow rate. On the left hand side of the CART (the occupied period), we see the second zone's air temperature recurring as a predictor. On the right hand side, the mixed air temperature and 24-hour cumulative cooling coil load appear in similar positions in each rule. As one progresses further down the tree, predictors begin to diverge more. Keep in mind, however, that many of these nodes would be typically pruned away due to the 1 standard error cost-complexity pruning rule employed.

Table 7.10 shows that these rules would also exhibit comparable statistical performance according to open loop tests. The first case[2] represents the original training set; case 2 uses the first week for cross-validation; case 3 uses the middle week for cross-validation. Each formulation is equally skillful in estimating optimizer decisions based on a survey of observed relative error values.

Table 7.10: Open Loop Statistical Performance of MM2 Rule Permutations

	Training Set Permutation		
	#1	#2	#3
RE	0.05	0.10	0.08
R^2	0.95	0.90	0.92

[2] Results to not exactly match those presented in Table 7.2 because trees have not been pruned in order to allow them to grow to a uniform depth.

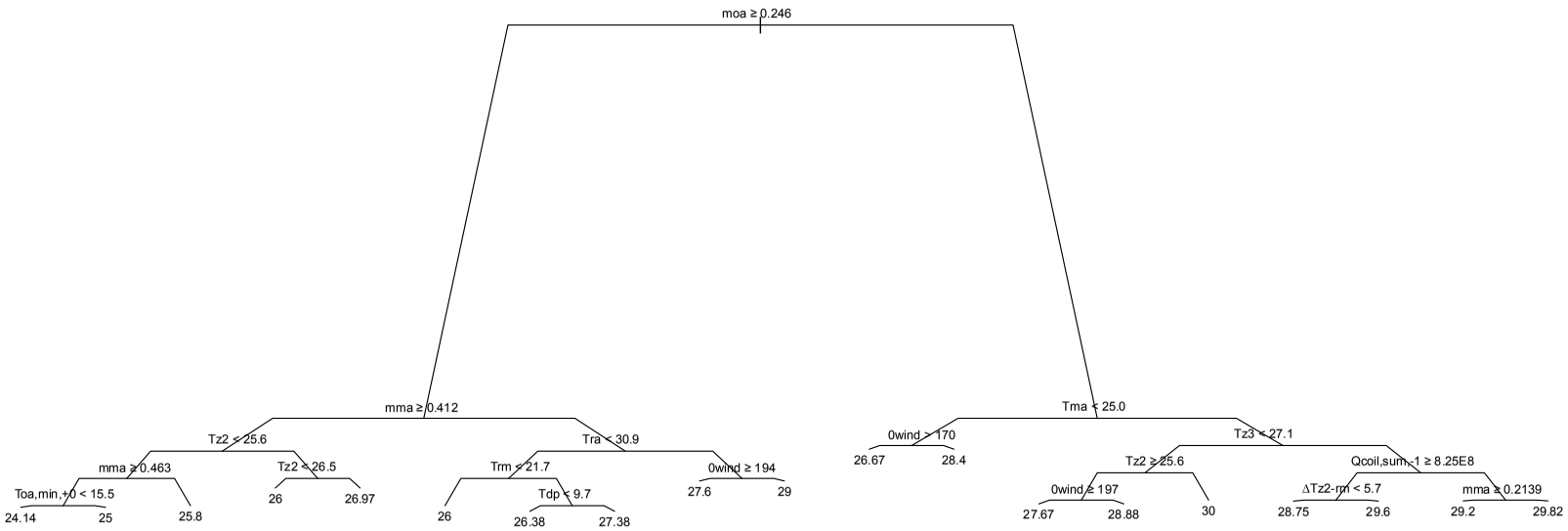


Figure 7.10: Dendrogram of the MM2 setpoint rule when cross-validated on the first week of results and trained on the last two weeks of results.

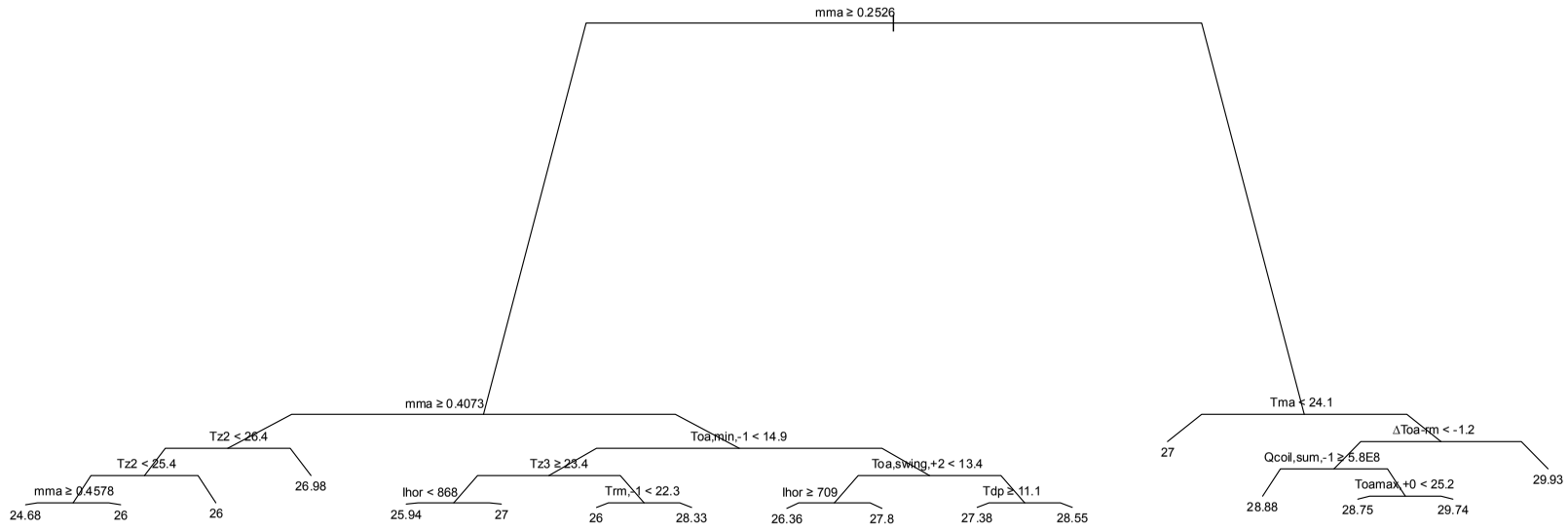


Figure 7.11: Dendrogram of the MM2 setpoint rule when cross-validated on the middle week of results and trained on the first and last week of results.

7.5.2 Investigating Seasonality

The prior section represents a threefold cross-validation, which provides a coarse picture of the sensitivity of rules to different regions of the offline MPC solution. In the case presented, this sensitivity was minimal, but suppose the offline MPC solution spanned an entire season (as was the case in Chapter 5), or the MPC solution displayed seasonality or non-stationarity. In these cases, it might be necessary to develop multiple rules for different seasons or even months of the year. A k-fold cross-validation process could be useful in diagnosing the specific points in time when different rules are required.

In k-fold cross-validation, the offline MPC solution set would be divided into k segments of equal duration. Let us assume that we have three months or about 12 weeks of solution data for a summer period but are unsure whether the solution patterns or relationships change in meaningful ways during that period. One could first employ a very coarse cross-validation—say, threefold—to explore whether rules are sensitive on the month scale. If noticeable differences in statistical figures of merit exist between the three permutations, one could continue to subdivide the solution set into 4, 6, or even 12 segments to examine the specific periods of time during which cross-validation results diverge. Change points in the solution could be identified in this manner, allowing for extraction of seasonally dependent rules. Clearly some expert knowledge would be required to discern the source of these shifts in MPC solutions (e.g. warmer weather trends or changes in seasonal setpoint schedules).

Naturally, some restraint must be exercised in this approach, for if one took the k-fold cross-validation procedure to its logical extreme, one might conclude that it is necessary to extract rules specific to each day of the year. As with model parameter selection, there is a balance between achieving the "best" performance and obtaining a parsimonious model. Although deeper investigation of this issue is required in future rule extraction research, intuition and common sense suggest that—building construction and system configurations aside—only long-term trends like seasonal weather or system operation patterns (e.g. seasonal setpoint reset schedules) should

affect the fundamental structure of MPC solutions and extracted rules. It is therefore reasonable to assume that extracted rules might vary on these longer time scales of months or weeks, but certainly not days.

7.5.3 Alternate Solutions to Interdependence Issues

The set of rules developed for MM3 suffered from instabilities; rules were unable to fully reproduce the interdependent pattern of operation present in the original MPC solutions, and errors cascaded through the system. Based on the limited set of cases evaluated in this dissertation, it is difficult to ascertain at this point whether this sort of result is merely indicative of a very sensitive control problem or an inherent weakness of all multi-rule extraction attempts. Regardless, there may be ways to better formulate MPC problems in the future to avoid such instabilities in the first place.

The concept is to encapsulate multiple control actions in a single decision vector, allowing for extraction of a single rule. For the combined night ventilation and setpoint rules developed for MM3, this could have been accomplished by reformulating the MPC problem in terms of a single decision variable, the cooling setpoint, and allowing this decision to drive the operation of both mechanical cooling and night ventilation. The problem would be reformulated as:

$$\begin{aligned} \text{Minimize } C_{tot} &= f(\vec{y}) = C_e + C_c \\ \text{Subject to} &: \\ & \vec{y} \in \mathbb{R}^{12} \\ & 16\,^{\circ}\text{C} \leq \vec{y}_{t \notin t_{occ}} \leq 30\,^{\circ}\text{C} \\ & 21\,^{\circ}\text{C} \leq \vec{y}_{t \in t_{occ}} \leq 30\,^{\circ}\text{C}. \end{aligned} \tag{7.7}$$

Night ventilation window positions could be governed by either a two-position or proportional local loop controller governed by the cooling setpoint. Availability of the night ventilation system would be dependent on time of day as well as some simple logic comparing zone, outdoor, and setpoint temperatures to ensure that $T_{db} < T_{sp} < T_{zn}$ (i.e. ensuring that outdoor air would be capable of providing cooling). Control would be implemented in a cascading fashion, with the night ventilation local loop evaluating first, followed by mechanical cooling local loops (i.e. VAV dampers). This would allow free cooling to meet as much of the load as possible, with any remaining load met by the mechanical system. Most importantly, a single decision vector of cooling setpoints would result, allowing for extraction of a single rule and eliminating the possibility for interdependence issues.

7.6 Conclusions

7.6.1 MM2 Setpoint Reset Rule

- Extracted rules were able to achieve good performance and agreement with optimal solutions for the cooling season only. This likely has to do with the markedly more stringent comfort restrictions imposed in the swing season (ASHRAE 55 static) versus the cooling season (ASHRAE 55 adaptive).

- It would have been impossible to formulate a reasonably functioning rule without the inclusion of air system states. This stands to reason, as the optimizer selected its solutions not only to maintain zone comfort within acceptable bounds, but also to minimize loads on the cooling coil, fan, etc.

- The extracted rule incorporated little forecast information. Although the MPC solution clearly exhibits decision-making that at least incorporates short-term predictions (e.g. mechanical pre-cooling that does not violate comfort boundaries), the use of one- and two-day-ahead forecasts did not appear to have great sway in solution trends. Current ambient, zone, and system air states were far more significant and should be the foundation for any simplified heuristics in simpler facilities like MM1 and 2.

- In cross-climate testing, the MM2 rule clearly degenerated in terms of its ability to track optimal sequences. The decent comfort and energy performance observed in these tests was a chance event.

7.6.2 MM3 Night Ventilation and Cooling Setpoint Rules

- A natural synergy existed between night ventilation and cooling setpoints from the offline MPC solutions. This same interdependence needed to be preserved in extracted rules, but also posed problem for growing CARTs. It was found that skillful sets of rules could only be developed when predictor sets were explicitly made incongruent. By basing each rule on slightly different states, the rules achieve a slight physical decoupling that lessens the impact of errors made by one rule on decisions of the other. This might prove difficult in cases with more than two rules.

- Extracted rules performed the best in the cooling season, again due to looser comfort restrictions.

- MM3 rules fared much worse when applied to other climates. The presence of two loosely coupled rules magnified the impact of errors.

- Application of the MM2 reset rule to MM3 resulted in very poor match with the original optimizer setpoint solutions. This is not surprising given the fundamentally different systems and loads present in the two buildings. This test underscores the caution that should be used in generalizing extracted rules to diverse building types and climates. A good heuristic is likely still a better tool.

7.6.3 General Observations

- Extracted rules are sensitive to building type, loads, and climate. Based on performance tests conducted thus far, one is convinced that extracted supervisory control rules would have to be custom tailored in order to be successfully applied to a facility. This means

rules would have to be extracted from offline MPC results using a model that accurately reflects the physical relationships of a specific building (for retrofits, a model calibrated to utility data; for design purposes, simply a well-constructed design model).

- More robust rules may be possible by adopting broader training datasets that expose the model to a broader range of potential disturbances. Climate, load scenarios, and weather sequences could all be incorporated while still affording a manageable number of simulation cases.

- While it is important to preserve the interdependence between control decisions in cases where multiple systems interact with each other, it appears that tight physical coupling should be avoided in implementation to eliminate instabilities that result from feed-forward effects. In the cases presented, this has been achieved by training setpoint and NV rules on slightly different datasets. It could also be achieved, as suggested above, by reformulating the MPC problem and resulting rules in terms of a single decision variable.

Chapter 8

Field Test and Experimental Validation

To examine the performance of extracted rules in a physical setting, a suite of tests were conducted in test cells owned by the Fraunhofer Institut for Solar Energy Systems (ISE). Further information is provided in Appendix E, and case studies are available in [105]. The test cells, depicted in Figure 8.1, are equipped with thermally activated building structures (TABS, also know as concrete core conditioning), occupant load simulators, transmission load simulators, and full instrumentation for detailed engineering analyses. The cells were used to investigate the performance of extracted control rules compared to conventional control sequences. Although initially envisioned as a backup test facility (attempts were made to test rules on an occupied building), the cells are perhaps better suited to validation as they completely eliminate the stochastic influence of occupants. Furthermore, the cells have been shown to be nearly adiabatic during the cooling season, effectively eliminating the influence of short-term weather changes on test results.

The field tests proceeded in the following sequence, each of which will be described in detail in the following sections:

(1) Develop calibrated thermal model of test cells

(2) Conduct offline MPC using the calibrated model in the same manner as in the offline simulation study

(3) Extract control rules from offline MPC results and associated equivalent solutions

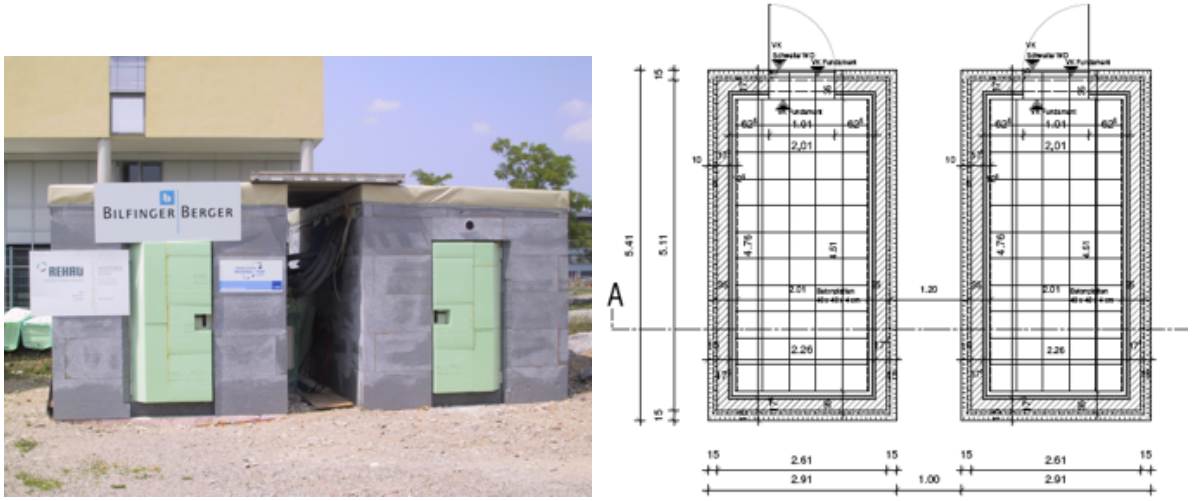


Figure 8.1: TABS test cells at Fraunhofer ISE. Photos courtesy Fraunhofer ISE.

(4) Implement heuristic and near-optimal controls side-by-side during multi-week operational tests

8.1 EnergyPlus Model Development and Calibration

As described above, the test cells are so heavily insulated and well sealed that they are adiabatic and internal load-driven during the warmer months of the year. Even though such a simple system could easily be modeled by a lumped parameter model, an EnergyPlus model was developed so that MPC could be conducted using our existing optimization environment. This model utilized the basic geometric boundaries, material properties, and system descriptions provided by researchers at Fraunhofer ISE. The cell has internal volume of 4.65 × 2.15 × 2.33 m, yielding a floor area of about 10 m^2 and an air volume of about 23.2 m^3. Typical material properties for the heavyweight concrete construction were adopted based on recommended values from ISE researchers (provided in Appendix E). Boundary conditions for all exterior surfaces were assumed to be adiabatic. The TABS system was modeled using the EnergyPlus `LowTemperatureRadiantVariableFlow` object, with tube length, diameter, and spacing per the physical installation.

Natural ventilation was accomplished in the test cell via a combination of supply/exhaust

fans located above the breathing zone in the ends of the test cell. Since we are not concerned with the detailed flow pattern, but rather the bulk heat transfer effects of natural ventilation, we conclude this to be a satisfactory approximation for validation purposes. Unfortunately, prior to experimentation Fraunhofer ISE was only able to provide a calibration dataset for the test cell operating under fully sealed (i.e. unventilated) conditions. Ventilation of the zone was, thus, modeled using a simplified air exchange model (Energy Plus' `ZoneVentilation:DesignFlowRate` object) with design flow rates set to 0.05 m^3/s, the design flow rate of fans acquired by ISE for testing purposes. This results in volume flows of about 180 m^3/h and air change rates of about 8 h^{-1}, which would be reasonable range for single-sided natural ventilation. Due to the very small size of the ventilation openings ($<$ 10cm), wind effects were not included in the model.

The model was calibrated based on two days of detailed measurement data from September 2009 provided by Fraunhofer ISE during which the test cells were in a periodic steady state (cooling capacity and loads were balanced to within about 8% on a daily basis). Measured weather data, including psychrometric conditions, insolation, and wind, from the same time period were applied. Supply water temperature, chilled water flow rate, and internal loads were also fixed according to measured sequences provided by ISE researchers (these parameters are prescribed in the BAS in the experimental setup). Occupant simulators comprised the internal loads (about 37 W/m^2) and were active between 7:00am and 5:00pm. The TABS system was operated between the hours of 6:00pm and 11:00pm, utilizing supply water temperatures of approximately 14 °C. Water was circulated at a constant flow rate of 140 L/h. Return water temperature, TABS cooling capacity, and zone mean air temperature comprised the model response and were used to guide calibration.

Calibration was conducted by adjusting several model parameters, listed in Table 8.1. The tightest model calibration results were achieved when adiabatic exterior boundary conditions were assumed and when a one-dimensional conduction transfer function was used to represent the TABS system. Slight adjustments in the TABS pipe length were made to adjust overall cooling capacity downward. A value of 6m was used, whereas the actual physical length of piping is an order of magnitude higher. Additionally, interior wall and ceiling convection coefficients were explored as

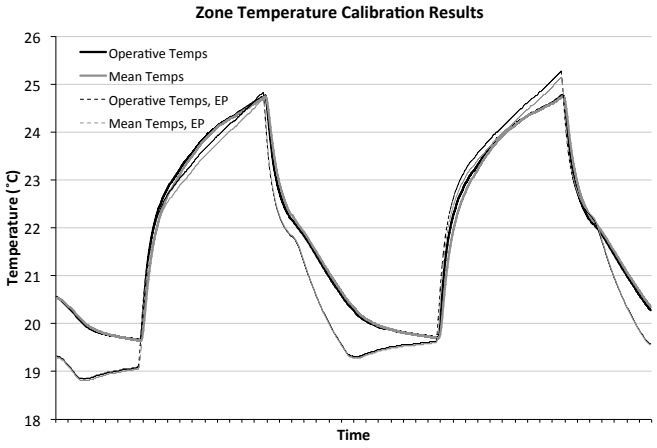
a possible means to increasing calibration fit, especially given the convective coupling between the radiant ceiling and room. Constant values in the range of 0.5–10 W/m^2-K were attempted, but in the end these had minor impacts, and the TARP adaptive convection algorithm was used. Final calibration results for temperature, cooling capacity, and return water temperature response are shown in Figure 8.2. The greatest model mismatch occurs during the charging periods in the late evening when the TABS circulation pump is in operation and heat is being extracted from the ceiling slab. At this time we see mismatch between chilled water return temperatures as well as zone temperatures. Zone mean air temperatures dip up to about 1K below the measured response of the cell; zone temperatures during occupied periods deviated at most by about 0.5K. Overall heat extraction by the TABS system on average remained within 10% of measurements over the calibration period. One possible explanation for this mismatch is that the energy model assumes a well-mixed zone, whereas in reality temperature stratification will occur since this is not an overhead, fully-mixed system. Another potential source of error could be differences in TABS valve controls; it is extremely difficult to replicate the precise local loop implementation of the experimental setup using an hourly simulation tool like EnergyPlus.

Table 8.1: TABS Test Cell Calibration Parameters

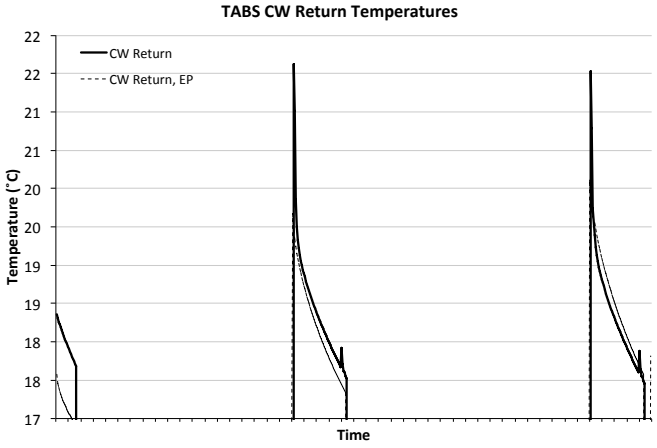
Parameter	Range/Options	Final Value
Exterior convection coefficients	TARP algorithm vs. adiabatic	adiabatic
Interior wall convection coefficients	TARP algorithm vs. constant	TARP algorithm
Radiant ceiling model order	1D vs. 2D CTF	1D CTF
Radiant ceiling pipe length	1–60m	6m

Several of the model assumptions were modified for final offline MPC runs and for initializing the thermal state of the test cells in live tests. Supply water reset curves developed by Olesen were used for the chilled water delivered to the slab [78, 77]. These guidelines vary supply water temperature as a function of outdoor temperature as follows:

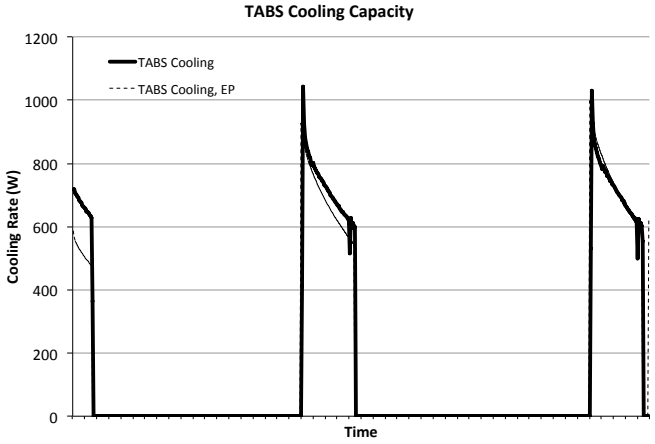
$$T_{chw,s} = 0.35(18 - T_{oa}) - 18, \tag{8.1}$$



(a) Zone temperature calibration results



(b) CHW return temperature calibration results



(c) TABS cooling capacity calibration results

Figure 8.2: Calibration results based on zone temperatures, CHW return temperatures, and TABS system cooling delivered. Greatest mismatch occurs during nighttime hours when the TABS circulation pump is running.

where T_{oa} is the outside dry bulb temperature. If $T_{cw,s}$ ever falls below the outside air dew point temperature, the supply water temperature is set to the current dew point temperature to prevent condensation on radiative surfaces.

8.2 Offline MPC Problem Formulation

Conducting offline MPC on the calibrated EnergyPlus model of the TABS test cell with consideration of natural ventilation is very similar to the problem formulation for MM4 described in Chapter 6. In the case of the TABS test cells, several variables were feasible for optimization, including availability of natural ventilation (binary), availability of pump operation (binary), and CW supply temperature setpoint (continuous). CW supply temperature and pump operation both effectively modulate the capacity of the TABS system, so it was deemed unnecessary to optimize both variables. Rather, the TABS CW setpoint was modulated per Oleson and pump operation was optimized. Because the test cells are outfitted with constant volume pumps, the binary variable for pump operation was transformed into a continuous variable via pulse width modulation (PWM) control for the pump. An hourly on fraction from 0 to 60 minutes was then optimized simultaneously with window openings in 3-hour blocks.

The objective function contained energy and comfort penalty terms. The energy term included circulation pump operation and "purchased" cooling energy[1] . Three different cases were run to examine solutions under different comfort penalties, including ASHRAE 55 (adaptive portion), EN 15251 (adaptive), and an energy-only case for comparison. Each case was run during a monthlong swing season period (May 11 through June 11) and a cooling season period (July 13 through August 13) using ASHRAE International Weather for Energy Calculations (IWEC) data for Strasbourg, France (the closest major IWEC location to Freiburg). The use of Strasbourg weather data clearly ignores any microclimatic effects present in Freiburg itself, but these effects are already masked by using data from generic weather stations rather than one located close to

[1] It was deemed unnecessary to fully model the plant supplying the chilled water, so only the cooling energy directly required to maintain the supply water temperature setpoint is counted in the objective function.

the building in question. Binary and box constraints were used to constrain the optimization. The final problem formulation was:

$$\begin{aligned} \text{Minimize } C(\vec{z}_{\text{vent}}, \vec{z}_{\text{TABS}}) &= C_e + C_c \\ \text{Subject to} &: \\ & \vec{x} \in \{0,1\}^m \\ & \vec{y} \in \mathbb{R}^m \\ & 0 \leq \vec{y} \leq 60 \end{aligned} \tag{8.2}$$

where:

C_e is the total HVAC energy use over the cost horizon,

C_c is the comfort penalty evaluated over the cost horizon,

$\vec{x}$ is a binary vector representing window opening decisions,

$\vec{y}$ is a vector of hourly pump PWM fractions, and

m is the number of time blocks in the planning horizon.

Due to the instrumentation of the test cell, the energy term was limited to include only the sensible cooling delivered by the TABS system. Circulation pump energy was excluded because it was orders of magnitude smaller than the cooling term and was not measured at the existing facility. Comfort penalties accrue when comfort for the optimizer solution fall outside of the given comfort window and are worse than the base case. Execution and planning horizons of 24 hours were used. A cost horizon of 72 hours was used to capture longer term thermal impacts.

8.3 Offline MPC Results

Offline MPC solutions were obtained for each of the three objective functions during swing and cooling season periods. The energy-only solutions, as expected, show the greatest energy

savings at the expense of occupant comfort. They are used here mainly as a benchmark against which to compare comfort-constrained solutions and to demonstrate the foregone energy savings resulting from tighter comfort requirements. Solutions using adaptive comfort penalties were highly comparable due to the similarity in the shapes of the ASHRAE 55 and EN 15251 adaptive comfort envelopes. The EN 15251 solution was ultimately used for rule extraction purposes and is used throughout the remainder of this chapter.

Results for a typical cooling season week (early August) of the EN 15251 solution are shown in Figure 8.3 below. Control behavior for a base and reference case are also shown. The base case represents the default operation of the calibrated model without ventilation; the reference case applies a heuristic to enable night ventilation when outdoor conditions fall within an acceptable range. Pseudocode for the heuristic is provided below:

```
IF Toa <= 20C & Toa >= 8C & Tzone >= 22C
    ENABLE NIGHT VENTILATION
END
```

The 22 °C night venting setpoint was chosen because this value falls near the lower portion of the EN 15251 adaptive comfort envelope (under typical German summer weather conditions) and thus prevented excessive cooling.

The optimizer solution departs from the base and reference cases in several important ways. First, the optimizer maintains ventilation almost continuously throughout the period, during which outdoor temperatures remain lower than indoor temperatures, providing ample free cooling opportunities. Due to the relatively low controlled ventilation rates (0.05 m^3/s), continuous ventilation can be maintained without overcooling the room. Secondly, the optimizer chooses to operate the TABS system at a significantly lower hourly fraction than the base or reference cases. The solution spreads the total cooling out over a longer period (the optimizer can operate TABS throughout the day, whereas the default control scheme pulses the slab only at night). The slab is still activated on a diurnal frequency, but the duration and intensity of the pulsing is altered.

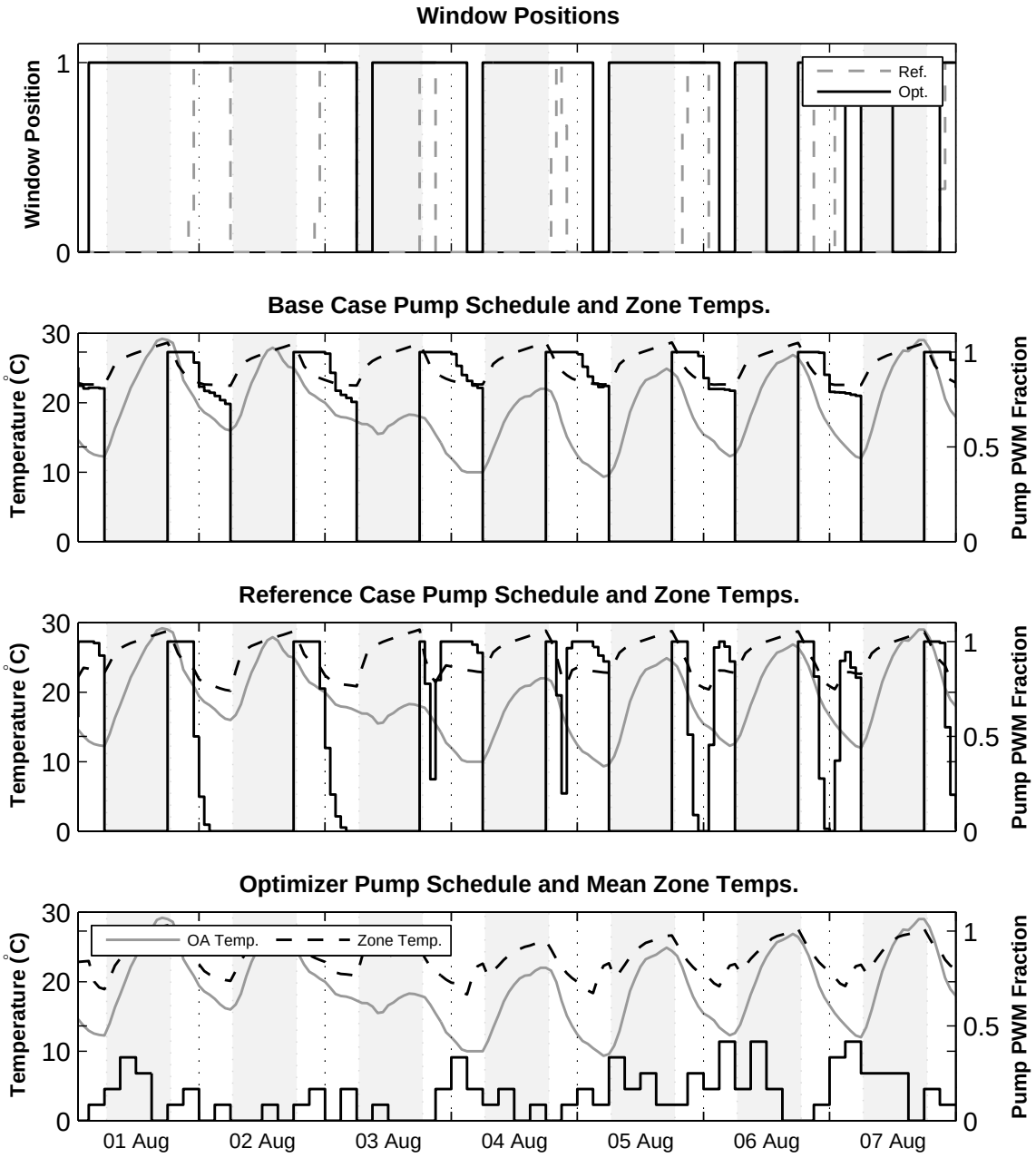


Figure 8.3: Offline MPC solution using the EN 15251 adaptive comfort penalty.

8.3.1 Energy Impacts

HVAC energy savings identified by the optimizer are significant during both the swing and cooling seasons (Table 8.2). The simulated swing season savings are slightly higher due to greater utilization of cooler outdoor air. Savings against the reference case are slightly less than the base case due to the reference case's existing utilization of night ventilation.

Table 8.2: TABS Cell Simulated Energy Use and Savings (kWh)

	Swing Season		**Cooling Season**	
	Use	Savings	Use	Savings
Base Case	150	135 (90%)	148	123 (83%)
Reference Case	129	114 (88%)	126	100 (79%)
Optimizer Case	16	-	26	-

As described above, the optimizer acquires as much free cooling as possible through the natural ventilation openings and displacing the need for TABS cooling. However, the optimizer is also able to deliver less total cooling to the space because the cooling system no longer has to maintain a fixed cooling setpoint. Rather, the optimizer is free to let operative temperatures "float" within the EN 15251 comfort window. As we will see, the optimizer has greater freedom to pre-cool the space, whereas the base and reference cases must adhere to fixed set points. Figure 8.4 illustrates the daily cooling provided to the cell in each of the three cases via TABS and natural ventilation. In the reference case, a greater share of the load is met with natural ventilation, and the mean cooling delivered to the cell remains similar to the base case. As weather warms, the optimizer gradually increases the PWM pulse frequency to the slab, adapting to the decreased availability of free NV cooling. In the optimizer case, natural ventilation is utilized to provide the vast majority of cooling in the cell, but the total cooling provided is also about a third less on average than in the other two cases. We can attribute this approximately 33% decrease in delivered cooling to the switch to a more forgiving, adaptive thermal comfort-guided control objective.

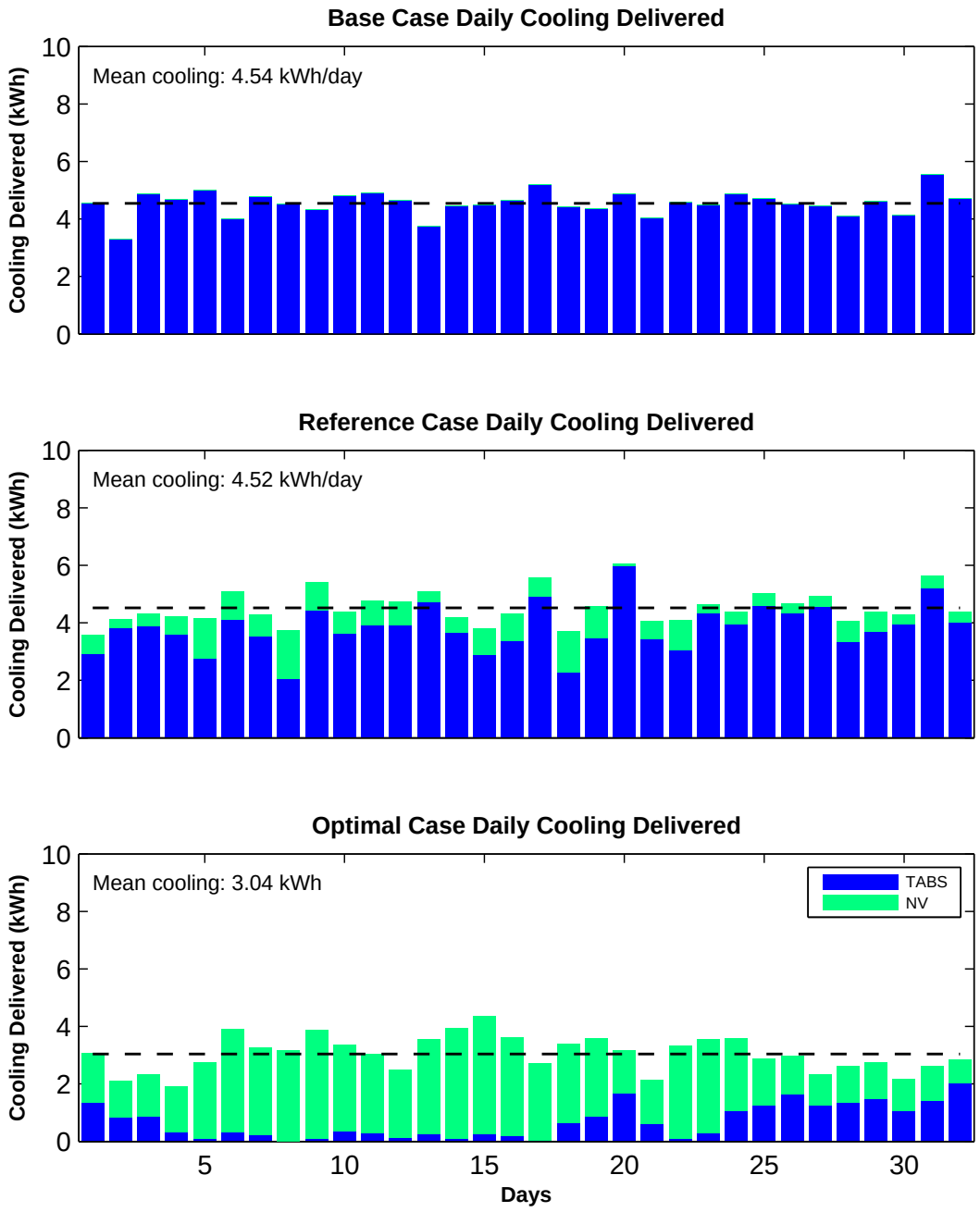
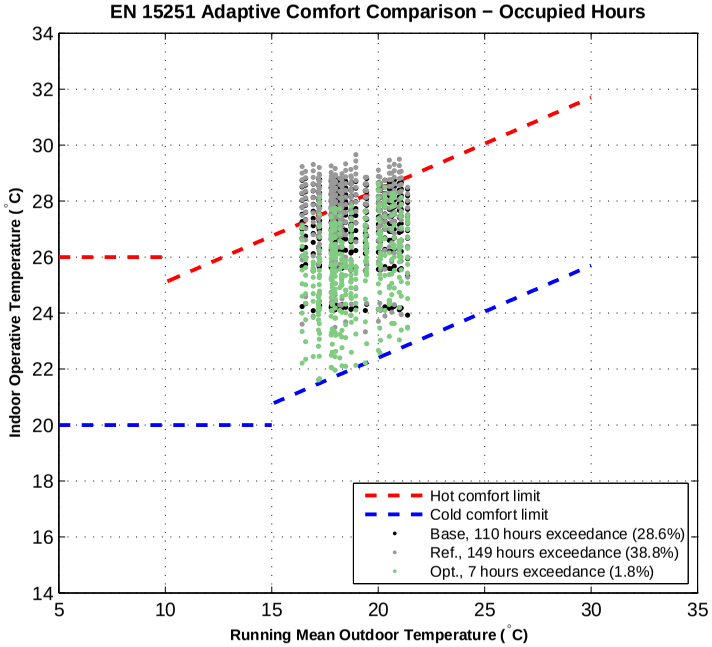


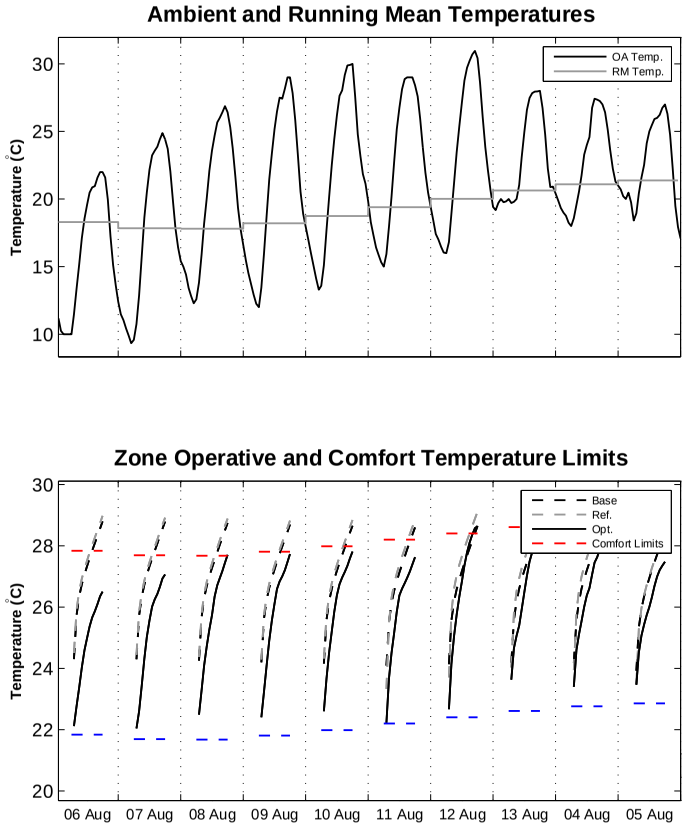
Figure 8.4: Simulated daily cooling energy delivered via TABS and natural ventilation during the cooling season period.

8.3.2 Comfort Impacts

Comfort dramatically improves in the test cell when MPC is applied. As illustrated in Figures 8.5(a) and 8.5(b), operative temperatures commonly swung above the adaptive comfort limit in the base and reference cases, particularly at the end of the day. Better utilization of the lower end of the adaptive comfort window resulted in deeper pre-cooling and ultimately maintained temperatures squarely within the comfort envelope (less than 2% comfort exceedances during occupied hours).



(a) EN 15251 comfort plot for occupied hours



(b) Time series plot of operative temperatures during occupied hours

Figure 8.5: Offline MPC solution comfort results

8.4 Rule Extraction

Given the use of two decision variables in the offline MPC stage, rules had to be developed for both a ventilation signal and a PWM circulation pump signal. GLM, CART, and AdaBoost examinations were conducted for each rule using various permutations of predictor sets (described below). CART-based rules were found to be the most practical, easily implementable, and effective form of rule nearly all cases and thus will be the focus of rule extraction analyses presented in this section. Training and cross-validation datasets were developed directly from the offline MPC results using the first three weeks of each simulation period. The remaining week of simulation was used for cross-validation. In order to provide a larger training set, swing and cooling season results were combined.

Several types of predictors were considered in the rule formation process. First, concurrent sensor values from the test cell, such as zone temperature, supply water temperature and outdoor dry bulb temperature, were used as predictors. Simple trend information was also incorporated into the predictor set. For example, the previous day's minimum outdoor air temperature was captured on a 24-hour sliding window and incorporated into the matrix of potential predictors. Other information can be derived from trend information, such as the running mean temperature and adaptive comfort boundaries on a given day. Simple forecast information could also be incorporated into the predictor set. One- and two-day-ahead forecasts of high and low temperatures—easily obtainable from Internet weather services—were incorporated as well.[2] The current status and trend information of the circulation pump could be used as a predictor in the ventilation rule, and vice versa. Finally, a handful of what we have called "proxy" variables were incorporated to introduce the variability of certain physical processes in the facility that might not normally be measured. For example, most facilities might not have the capability to measure the instantaneous cooling delivered through a single TABS circuit, but they most likely would know the cooling supply water temperature and water flow rate through the system and might be able to measure

[2] For the purposes of testing potential rules, we assumed perfect knowledge of the weather forecast, adopting high and low temperature "forecasts" directly from the Strasbourg weather file.

the surface temperature of a chilled ceiling. A cooling rate proxy variable could then be defined as

$$\dot{Q}_{\text{TABS}} = \dot{V}(T_{\text{surf}} - T_{cw,s}), \tag{8.3}$$

where $\dot{V}$ is the volumetric flow rate of chilled water, T_{surf} is the measured surface temperature, and $T_{cw,s}$ is the chilled water supply temperature. The resulting $\dot{Q}_{\text{TABS}}$ is highly correlated to the actual cooling delivered through the TABS system, but is simply presented in different units (m^3-K/s instead of W). A list of predictor variables considered is provided in Table 8.3.

8.4.1 Ventilation Rule Development

A binary classification rule in the form of a CART was trained on the offline MPC solution and tested against a cross-validation set. The unique physical constraints on ventilation in the experimental setup and the model (relatively low ACH) resulted in offline MPC solutions with ventilation operating over 80% of the time, as discussed above. Thus the training set for the ventilation rule was dominated by "on" operation, even given variations in weather, and there were very few instances of "off" upon which to train the rule. As one would expect, the resulting CART rule for ventilation has very high probabilities of operating NV, and ventilation can only be disabled under a narrow set of conditions. Figure 8.6 provides a dendrogram of the final CART as implemented in September 2011 field tests. The use of $\dot{Q}_{\text{TABS},1}$ as a predictor introduces an important dependency on the operation of the TABS system from the previous day. This is governed by the TABS rule presented in the next section.

8.4.2 TABS PWM Rule Development

A rule was developed to predict the optimal PWM signal (in minutes per hour) for the operation of the TABS chilled ceiling, a task that proved significantly more challenging than the binary ventilation rule. For one, the response variable $\vec{z}_{\text{TABS}}$ is continuous, ranging from 0 to 60 minutes. Furthermore the response of the optimizer tends not to be normally distributed, and so standard least-squares multiple linear regression techniques could not be directly applied. Attempts

Table 8.3: Predictors Examined for TABS and Ventilation Rules

Concurrent Sensor Readings			
T_{oa}	Outdoor dry bulb temperature	θ_{wind}	Wind direction
RH_{oa}	Outdoor relative humidity	v_{wind}	Wind speed
I_{hor}	Global horizontal insolation	T_{upper}	Adaptive comfort upper limit
T_{op}	Zone operative temperature	T_{lower}	Adaptive comfort lower limit
RH_{zn}	Zone relative humidity	$T_{chw,s}$	Chilled water supply temperature
T_{rm}	Outdoor running mean temperature	T_{surf}	Chilled ceiling surface temperature
z_{TABS}	PWM control state of TABS pump	z_{vent}	On/off state of ventilation
24-Hour Sliding Window Trend Information			
$T_{oa,\text{max/min}}$	Max/min outdoor temperatures	$RH_{zn,\text{avg}}$	Average zone relative humidity
$T_{oa,\text{avg}}$	Average outdoor temperature	$T_{rm,1}$	Prior day running mean temperature
$RH_{oa,\text{avg}}$	Average outdoor relative humidity	$T_{\text{upper},1}$	Prior day adaptive comfort upper limit
$I_{\text{hor,sum}}$	Prior day total insolation	$T_{\text{lower},1}$	Prior day adaptive comfort lower limit
$T_{op,\text{max/min}}$	Max/min zone operative temperatures	$T_{op,\text{avg}}$	Average zone operative temperature
$T_{\text{surf,max/min}}$	Max/min ceiling surface temperatures	$T_{\text{surf,avg}}$	Average ceiling surface temperature
Forecast Variables[1]			
$T_{oa,\text{max/min},+i}$	Forecast high/low outdoor temperatures	$T_{\text{swing},+i}$	Forecast temperature range
Proxy Variables			
$N_{\text{TABS},1}$	Prior day total hours TABS operation	$N_{\text{vent},1}$	Prior day total hours ventilation operation
$\dot{Q}_{\text{TABS},1}$	Prior day total TABS cooling delivered	$\dot{Q}_{\text{vent},1}$	Prior day total ventilation cooling delivered

[1]Forecasts are provided i days ahead. A 0 corresponds to the forecast for the current day.

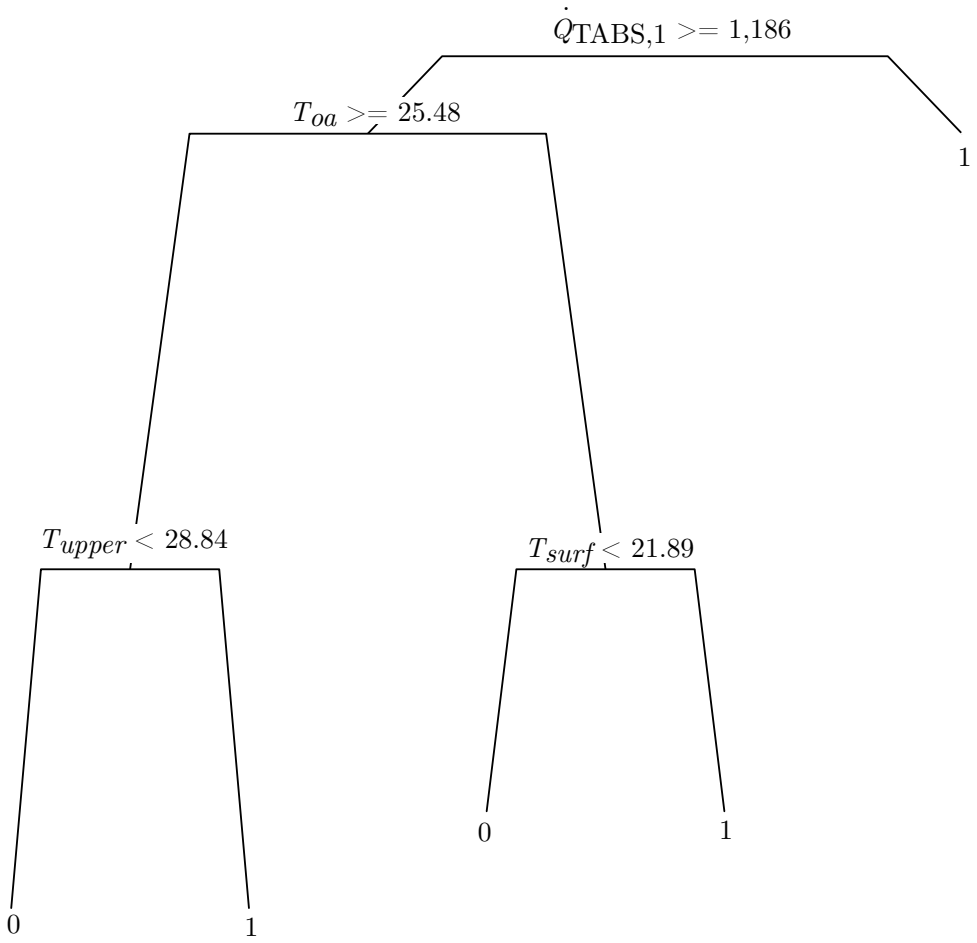


Figure 8.6: The CART dendrogram for the ventilation rule.

were made to form a rule using a GLM with a gamma distribution link function, using stepwise regression to prune the predictor set. Aside from a poor model response, this formulation often forecast PWM values exceeding 60 minutes per hour, which was undesirable.

Regression trees were developed to forecast the PWM signal at various frequencies, ranging from hourly to multi-hour time blocks. None of the models provided any meaningful match with the original optimizer sequence. We postulate that this may have to do with the underlying dynamics of the TABS system itself. Thermally massive buildings exhibit settling times on the order of days due to building materials, such as concrete slabs, with high thermal capacitance. The optimal control of a TABS system is inherently tied to those dynamics, and any relationships describing the optimal control will likely depend greatly on thermal history and forecast information. This is in contrast to the ventilation side of the system, which can immediately impact energy and comfort in the zone when activated. For rule discovery purposes, one might require days worth of temperature states to extract an accurate PWM rule.

Rather than trying to accurately reproduce the PWM signal at each point in time, a simplification was made. Given the high mass of the radiant cooling system and the fact that these systems are mainly designed to provide a large thermal mass at a constant temperature into which a zone can dump its loads, it was decided that a good rule for TABS control need only follow the general PWM profile for a given day, not reproduce any exact sequence.

A two-step process was used to develop a CART-based rule for selecting this profile. First, a k-means clustering analysis was used to examine the daily PWM profiles present in the optimal solution, similar to the examples presented in [81]. Due to the time blocking of the offline MPC problem, each day represented an 8×1 vector of PWM values. Clustering was performed to identify solutions with similar overall shapes. The analysis confirmed that a limited number of profiles existed. Through an iterative process, eight clusters were identified (Figure 8.7) to represent different typical daily profiles implemented by the optimizer. The original optimizer solutions could then be categorized by cluster, yielding a $n_{\text{days}} \times 1$ vector, $\vec{z}_{\text{clust}}$. Classification rules could then be trained to predict $\vec{z}_{\text{clust}}$. These rules classify the type of solution profile to be implemented on a

given day (decisions must be made at the beginning of each day), then implement the center (mean response) of the corresponding cluster as a 24-hour PWM schedule. This can be considered a form of bootstrap resampling.

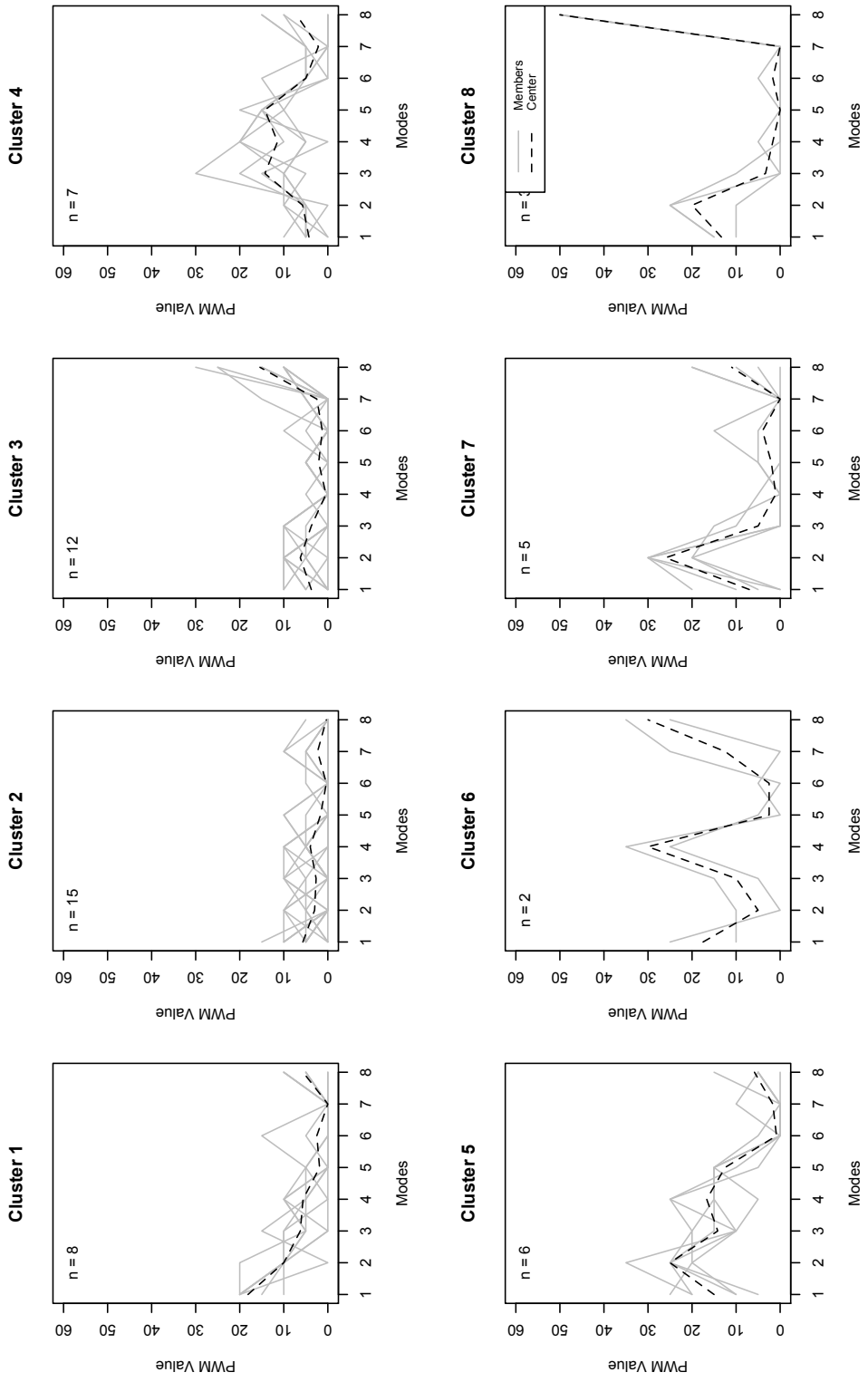


Figure 8.7: k-means clusters for PWM signal. Cluster centers represent the mean of all members in the cluster.

A dendrogram of the resulting rule is shown in Figure 8.8. It has a somewhat more complicated structure than the ventilation rule and has greater dependence on forecast and trend variables owing to the cooling system's slower response. For example, the leading splits in the CART pertain to one- and two-day-ahead maximum temperature forecasts. This proved to be one of the only cases in the course of the entire research program in which longer term forecast variables were significant to rule structure.

Initial simulation tests confirmed that the rule was both skillful at choosing an appropriate category for the given day and approximating the profile of the original optimizer solution (Figures 8.9(a) and 8.9(b)). Although the resulting profiles are not an exact match with the optimizer results, they provide a similar daily quantity of cooling, which may be all the accuracy required for such a heavy mass system. The approach does, however, have one major downside. Like many non-parametric statistical techniques, the TABS rule resamples the original solution. The advantage and disadvantage of this approach is that the rule can only implement solutions explored by the optimizer. PWM values will remain within reasonable ranges, but the PWM profiles will be inherently limited to the eight "prototypical" solutions identified in the clustering analysis.

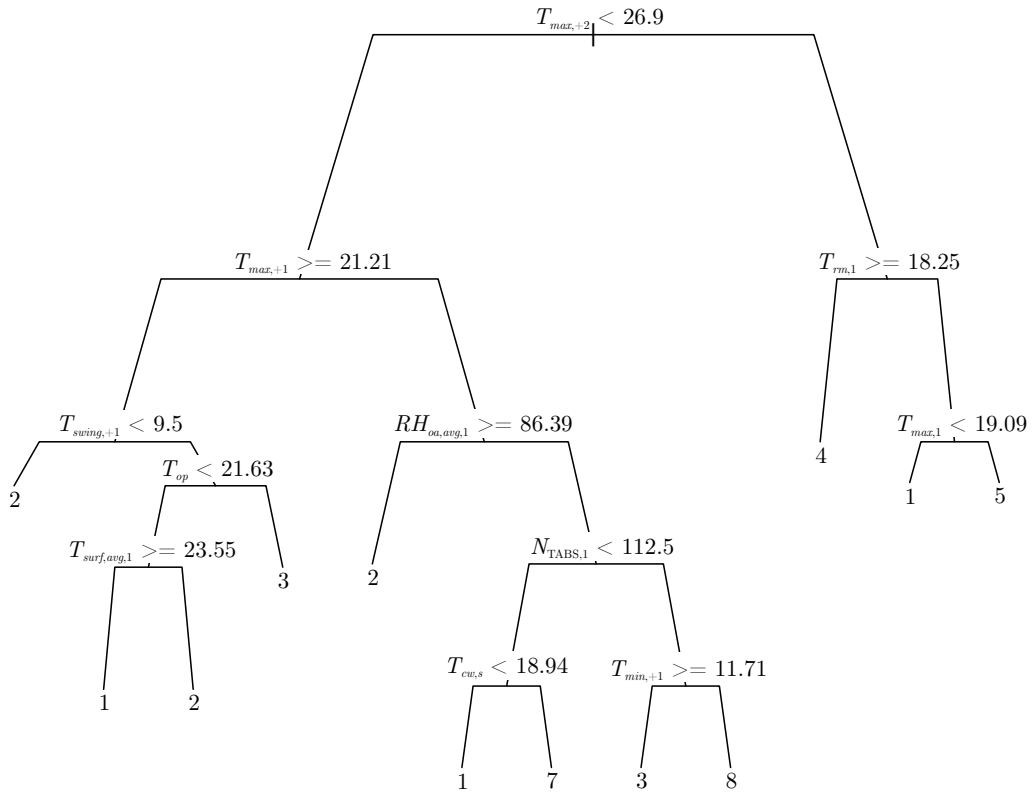
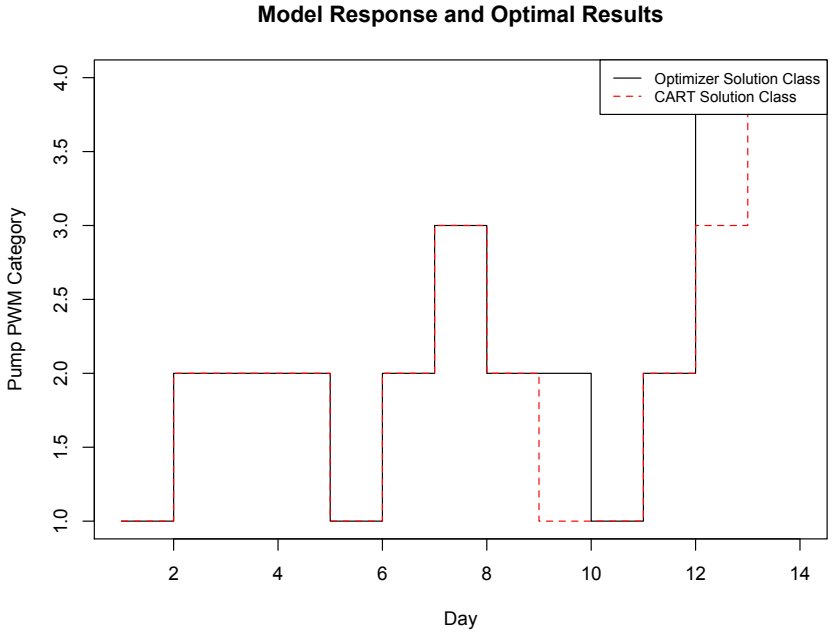
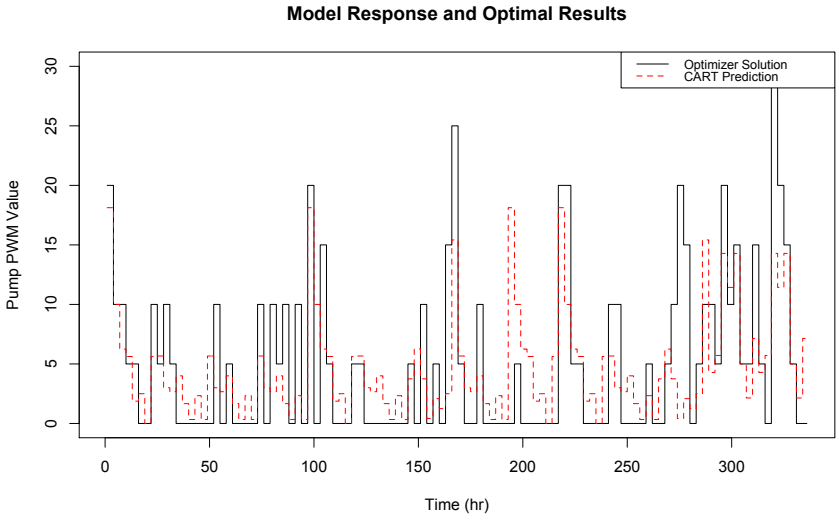


Figure 8.8: A dendrogram of the CART rule for TABS operation. The rule exhibits greater dependence on trend and forecast variables due to the slower response of the TABS system.



(a) Daily classification response.



(b) Hourly PWM sequence.

Figure 8.9: Response of CART-based cluster resampling rule for PWM sequence. The upper graph shows how the rule provides reasonably accurate daily classifications; the lower graph shows how those classifications translate into PWM sequences.

8.5 Field Implementation

8.5.1 Implementation Details

A field experiment was conducted at the Fraunhofer ISE from September 20 through October 1, 2011 using the aforementioned TABS-equipped test cells and their associated monitoring and data acquisition system. The cell was first run using the default TABS control algorithms which match the control scheme used in the base case EnergyPlus simulations. No ventilation was allowed, although a small amount of infiltration did occur due to the openings required for the ventilation fans ($\sim$ 0.1 ACH). The base case was allowed to equilibrate under mild weather conditions for a period of ten days, the last six of which are used for energy and comfort comparisons against the near-optimal rule control case. An implementation of the reference case—base case TABS control plus a simplified night ventilation heuristic—was not attempted due to time constraints.

The two near-optimal rules were implemented using a purpose-built data acquisition and automation system developed by the ISE. The binary ventilation CART was used to directly control the on/off state of ventilation fans on an hourly frequency. The TABS system was controlled via an electronically controlled two-way valve into the chilled ceiling circuit. Local weather forecast information for Freiburg, Germany was obtained from a free Internet weather service[3] and manually entered into the program on a daily basis. All other required predictor variables (along with other variables from Table 8.3) were monitored both in the test cell and associated systems and on a rooftop weather station. Measurements included ambient weather conditions (dry bulb temperature, relative humidity, insolation, wind, etc.), zone air and operative temperatures, supply and return chilled water temperatures and flow rate, internal gains (via load simulator), and temperatures at various points within the chilled ceiling. For all measurements in the experiment, it is safe to assume errors of less than 2% of the measured value. Detailed information regarding all instrumentation and uncertainty can be found in Appendix E.

An unintended mismatch between the calibrated model and the experimental test setup did

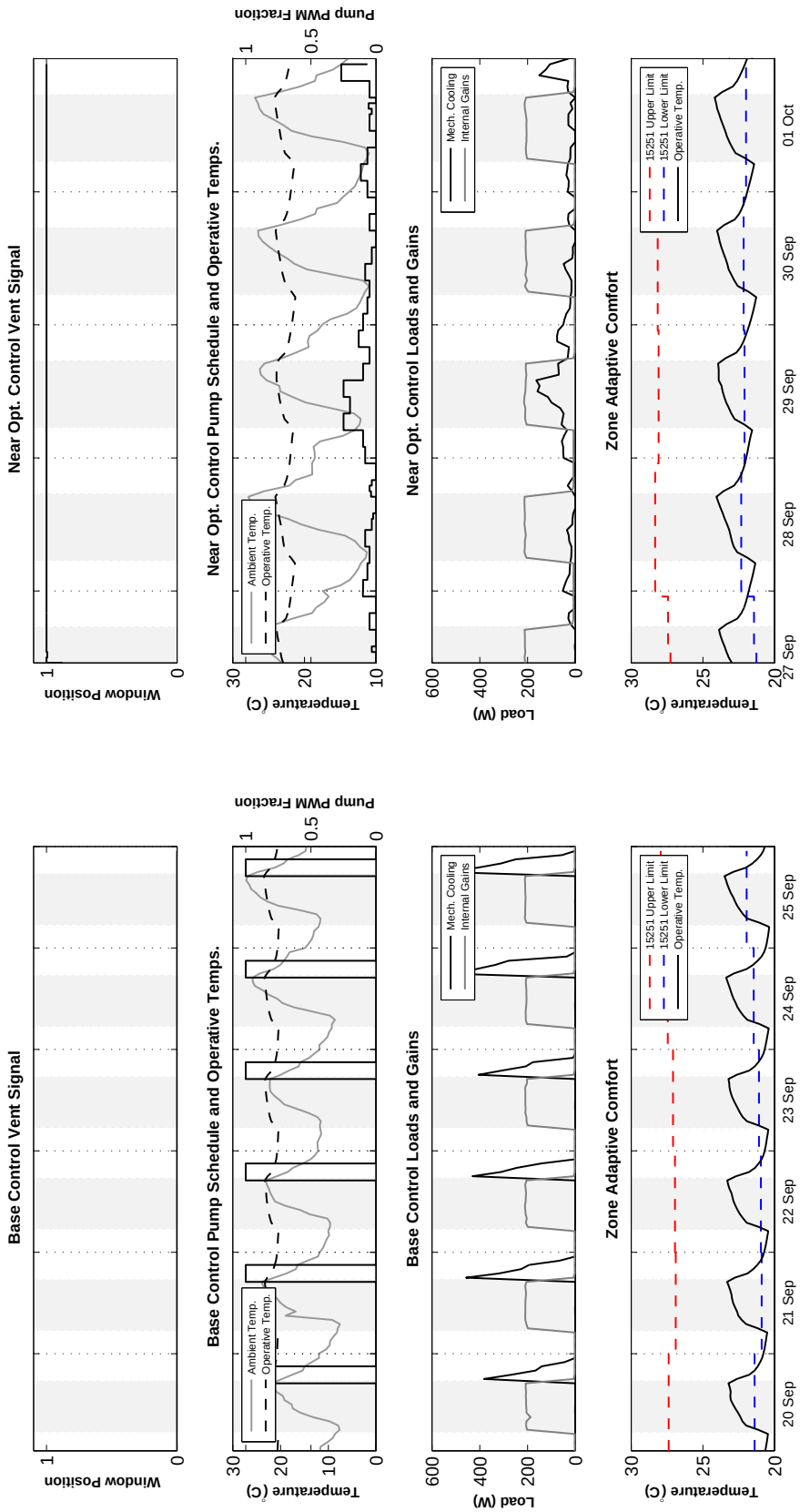
[3] Weather Underground, Inc., available at `http://www.wunderground.com`.

occur. The calibrated model assumed total internal gains of approximately 37 W/m^2, whereas the actual loads applied during tests were about 20 W/m^2. The natural effect of the change is that the internal gains in the cell were roughly halved. Because the cells are internal gain dominated, we see much lower daily temperature profiles, as expected. Although unfortunate, the mismatch does provide an opportunity to examine the robustness of an extracted rule when operating outside the boundary conditions governing its original training set. Due to time constraints and the approach of significantly colder weather, it was not possible to rerun tests with the expected internal gains.

8.5.2 Results

A snapshot of the cell's operation under baseline and near-optimal control is presented in a several time series charts in Figure 8.10, with baseline control on the left and near-optimal control on the right. The uppermost chart shows the binary ventilation signal; the second chart plots ambient and operative temperatures on the left axis and PWM schedules on the right axis; the third chart shows the resulting mechanical cooling compared with internal gains; and the lowermost chart shows the operative temperatures compared with the daily EN 15251 adaptive comfort envelope. One notices immediately that the ventilation rule called for continuous ventilation of the space and that the TABS rule calls for significantly lower frequency, but nevertheless daylong, operation of the TABS system. The mechanical cooling provided by the radiant ceiling is also proportionately lower, as expected. The behavior of the ventilation rule is particularly interesting, because a quick examination of the measurements demonstrate that the rule called for ventilation at times when the ambient temperatures exceeded zone temperatures. This can be explained by the structure of the ventilation rule itself (Figure 8.6), which depends foremost on the proxy cooling variable $\dot{Q}_{\text{TABS},1}$. If the value of $\dot{Q}_{\text{TABS},1}$ never exceeds 1,186, ventilation will be continuously enabled. Not surprisingly, these conditions were never experienced by the system because it would have required greater ΔT over the TABS circuit, experienced during the original offline MPC runs due to higher internal gains. A comparison between Figure 8.10 and the simulated MPC results from Figure 8.3 further confirms the impact of the load mismatch between the model and experimental setup. The

base control operates the TABS system for a smaller fraction of the unoccupied hours due to lower loads on the system, and the zone operative temperatures typically experience only a 3K diurnal range, compared to the 4-5K range seen in the original model.

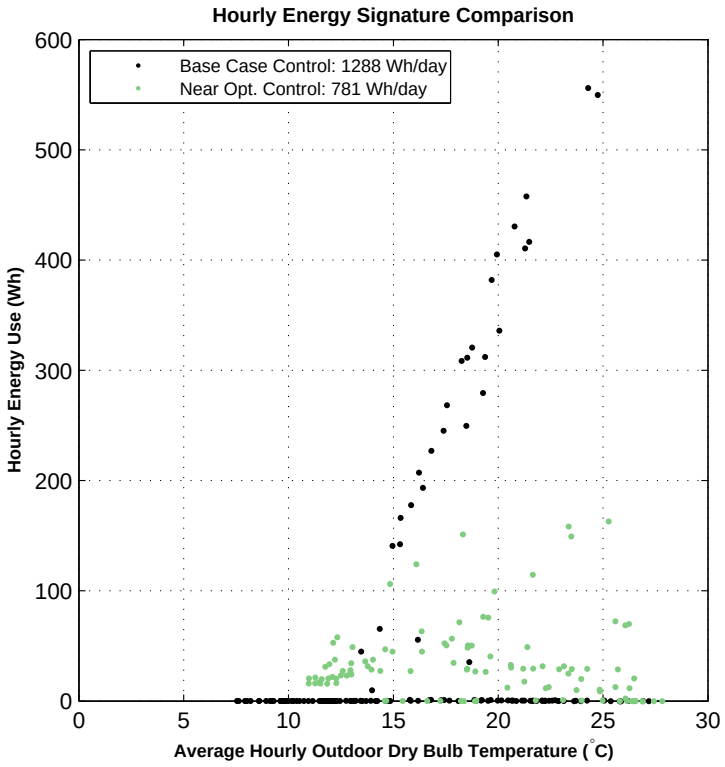


(a) Baseline control

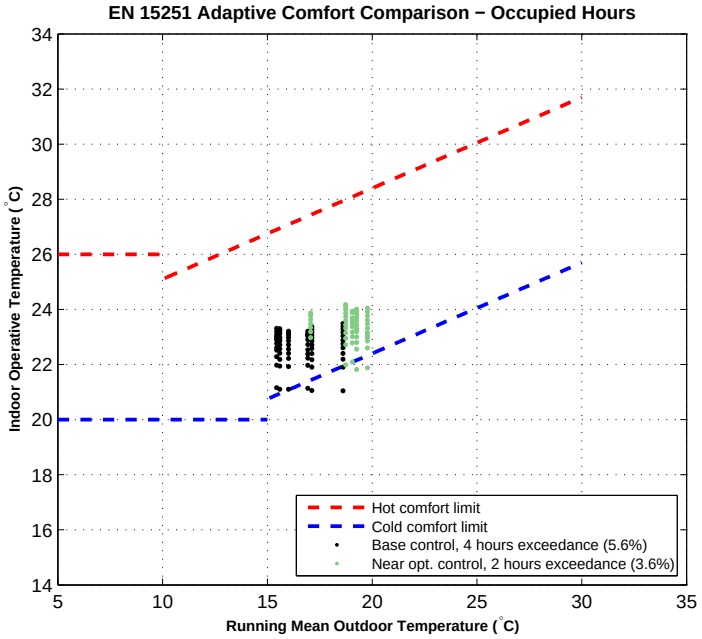
(b) Near-optimal control

Figure 8.10: Time series measurements from field test for baseline and near-optimal control of TABS test cell.

Mismatch aside, one can still make useful energy and comfort comparisons between the two control cases. Figure 8.11(a) shows energy signature plots, and Figure 8.11(b) plots operative temperatures during occupied hours on the EN 15251 comfort envelope. From an energy standpoint, we can see that the combination of rules yielded on average 40% lower energy consumption compared to the baseline control through increased utilization of ventilation and decreased use of TABS. The energy signature of the near-optimal case has a significantly shallower slope than the base case. These benefits were realized with overall improved comfort, with the extracted rules generating fewer and less severe comfort violations during occupied hours. Although much longer testing periods would be required to more completely examine this effect, one even notes that the operative temperatures under near-optimal control appear to more closely track the lower boundary of the EN 15251 envelope, as was the case in the offline MPC training set. The base case control obviously does not contain any adaptive "logic" and simply tries to maintain a fixed setpoint of 22 °C.



(a) Measured daily energy signatures



(b) Measured operative temperatures vs. running mean temperatures

Figure 8.11: Measured energy signatures and zone operative temperatures for the base and near-optimal control cases.

8.6 Model Mismatch, Sub-Optimality, and Robustness

8.6.1 Identifying and Correcting Sources of Mismatch

Two types of mismatch existed between the offline MPC simulations and the experiment: mismatched disturbances and mismatched model parameters. With regards to disturbances, we already know that the loads were approximately one half of the value used in the offline MPC runs, but in addition, the offline MPC runs were also based on TMY weather rather than real weather sequences. These mismatches were easily explored and simulated by using internal loads set to the 20 W/m^2 value and running the original model on weather data measured at ISE. Additional model parameter mismatches were identified and corrected. Some adjustments were made to reflect measured airflow values realized during the tests. First, a background infiltration value of 0.1 ACH was added to the model (previously, the envelope was assumed effectively airtight). Flow rates during active ventilation were halved from 0.022 m^3/s to 0.011 m^3/s. It was necessary to abandon the adiabatic exterior surface boundary conditions and adjust the TABS system size slightly to achieve agreement on TABS delivered cooling. The re-calibrated model, however, deviates less than 1% from the base case measured data with regards to delivered cooling energy in the TABS circuit.

8.6.2 Model Mismatch Impacts

An optimization was rerun on the original model using the measured weather sequence from September 27 through October 1. The resulting solution is very similar to the original MPC solution (Figure 8.3) as well as the near-optimal rule implementation (Figure 8.10), with constant ventilation and diurnal pulsing of the radiant ceiling at low PWM values (hourly fractions less than 0.5). In brief, the differences between actual and IWEC weather sequences did not severely alter the nature of the solution or the potential savings given perfect weather forecasting. The optimizer realizes very similar savings over the baseline control: slightly over 84% versus the 83% realized in the cooling season case. Comparisons between total energy savings cannot be made

because the original optimizations were conducted over a monthlong period; however, daily average savings for the original solution were 3.8 kWh/day compared to about 4.0 kWh/day for the case rerun on measured weather. Thus, differences in ambient conditions at least did not affect the total achievable MPC savings by very much.

When the internal gains mismatch was also taken into account, a significantly different solution emerged. Both the ventilation and TABS decision variables were impacted. Ventilation was enabled less frequently, with several longer periods of closure during the beginning of the optimization period. The TABS system was hardly required at all, as ventilation could meet the lower loads without operative temperatures rising too dramatically. Savings were significantly lower, at 2.1 kWh/day because the baseline energy use was proportionately lower as well. When remaining model mismatches were accounted for, savings dropped an additional 50% to 1.0 kWh/day due to the constrained ventilation cooling capacity. Figure 8.12 provides a snapshot of the fully re-calibrated solution, demonstrating how the optimizer would have almost completely abandoned the use of the TABS system in favor of ventilation, with some breaks in ventilation corresponding with daily extreme temperatures. The solution also hovers somewhat closer to the lower limit of the comfort window, shown in Figure 8.13, although not nearly to the degree seen in the experiment.

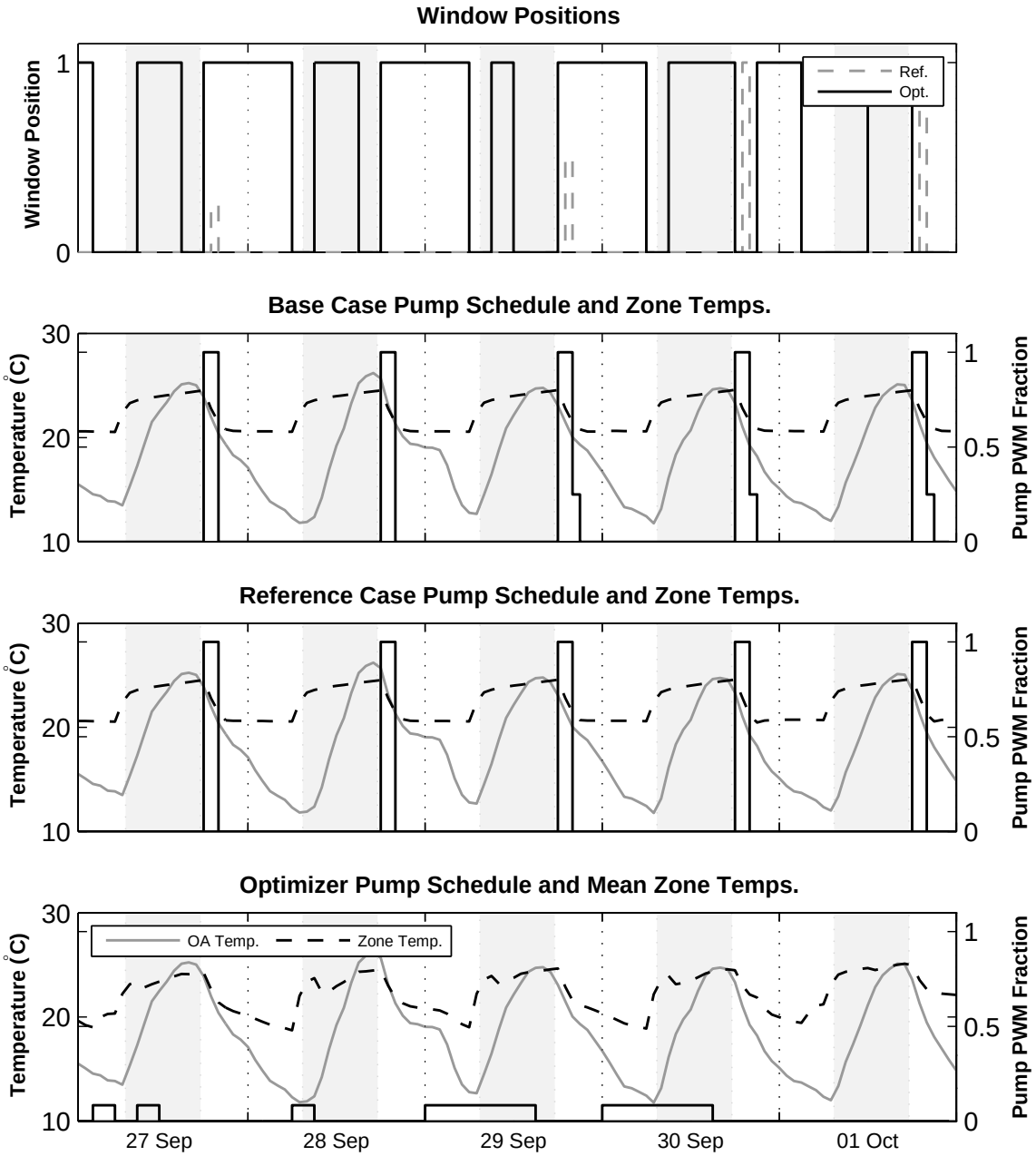


Figure 8.12: Offline MPC solution for re-calibrated test cell model.

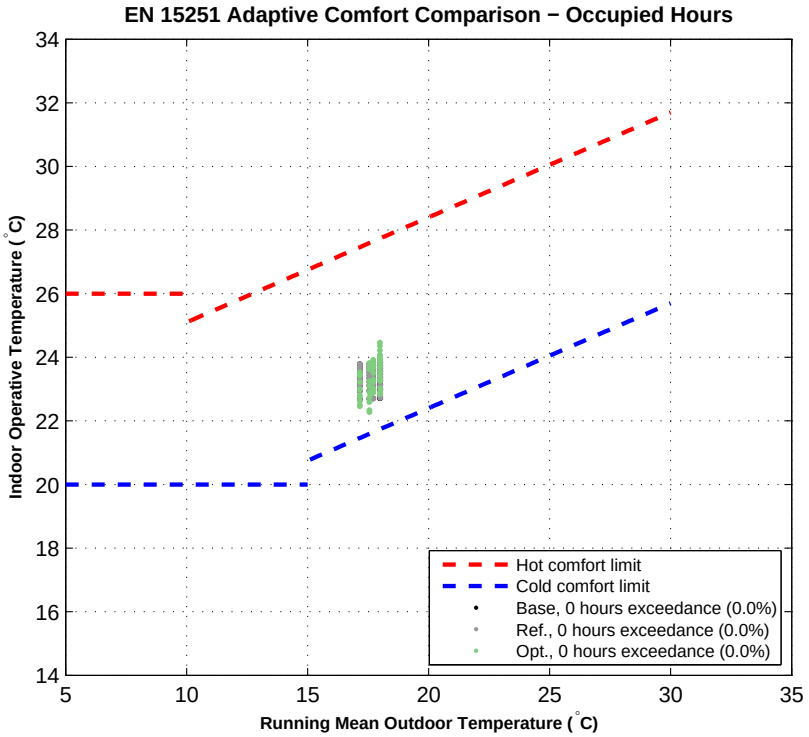


Figure 8.13: Comfort for re-calibrated, re-optimized case.

8.6.3 Weather Forecast Uncertainty

A summary of the field test results and a variety of simulated cases—both mismatched and calibrated—are presented in Table 8.4. Comparisons can be made against the original offline MPC solution, simulated tests of the near-optimal rules, and a variety additional MPC cases. The simulated cases vary by the degree of mismatch with the experimental measurements. Within the offline MPC results, the single largest change in response occurred when internal gains were changed to reflect the lower values used in the experiment. This is not surprising, as lower loads should translate into lower energy use and overall lower savings compared to the base case. When all mismatch was accounted for, the optimizer was able to achieve about 1 kWh/day in savings via free cooling, about a 77% overall energy savings compared to the 40% realized in the field.

The same basic pattern can be seen when the extracted rules were simulated in closed loop

tests. Regardless of the level of model refinement or mismatch corrected in the energy model, closed loop tests showed extracted rules achieving upwards of 90% of optimizer energy savings. The rules did not perform nearly as well on comfort, showing comfort violations in the tens of hours for most simulations. Again, this should not be surprising since the eight solution profiles implemented by the rule were developed by the optimizer under conditions with much higher internal gains, and the rule calls for unnecessary cooling. Still, one can say fairly objectively that the extracted rules should be capable of nearly reproducing optimizer results under ideal conditions.

The results of the experiment, particularly energy savings, still did not match simulated cases even when the aforementioned sources of mismatch were accounted for. The experimental test was only able to achieve half of optimizer savings. The one remaining factor not accounted for in simulation was weather forecast uncertainty. Recall that forecast max/min temperatures featured heavily in the TABS rule, and in fact, two-day-ahead temperature forecasts were the leading split in the tree. Weather forecasts being notoriously uncertain, this proved to be a major factor in the experiment. Once actual weather forecasts utilized during the experiment were incorporated into the closed loop tests, more than half of the remaining disagreement between simulated and measured energy savings were eliminated, bringing measured and simulated daily energy use values within 22% of each other.[4] Forecast uncertainty stands out as the second largest factor explaining the degraded performance of the extracted rules under test.

The remaining 22% error could arise from a number of sources. As alluded to earlier in this chapter, the EnergyPlus model does not take temperature stratification into account which would likely impact surface temperatures along the radiant ceiling. This is likely a minor effect. The model also assumes a constant rate of ventilation supplied to the space when fans are enabled which is a clear simplification. Because the ventilation portals communicate directly with the outdoors, they will be influenced to a degree by wind pressures around the test cell. It was never possible to achieve a tight calibration on the ventilation model (through a blower door test, for

[4] Forecast uncertainty impact was not evaluated using offline MPC because detailed, hourly forecast information for the period in question was not available.

Table 8.4: Summary of Experimental Results and Simulated Responses

	Daily Avg. HVAC Energy		Comfort Violations	
Mismatch Corrected	Use (Wh)	Savings (Wh)	Number (h)	Severity (K-h)
Offline MPC				
None[a]	699	3,845	1	0.2
Weather	883	3,612	6	1.1
Weather, gains	178	2,114	0	0
Weather, gains, model	342	1,013	0	0
Near-Optimal Rules				
None[a]	1,208	3,336	13	10.1
Weather	744	3,751	15	6
Weather, gains	285	2,007	11	8
Weather, gains, model	377	978	14	11
+ Imperfect forecast	607	748	17	14
Experiment	781	507	2	0.772

[a] These cases represent the conditions under which the original rules were trained and include all model mismatch. Original runs were conducted over a monthlong simulation period, so comfort violations have been prorated to a five-day period to enable comparison with the other cases.

example), so this parameter of the model still harbors some uncertainty. Finally, errors may have been introduced through instrumentation. Although the thermocouples used throughout the experiment are calibrated to very tight tolerances (±0.1K for water temperature measurements), their location outside of the conditioned portion of the cell and exposure to ambient conditions could have introduced artifacts, especially when the cell was not operating in a steady state. In short, while it may not be possible to fully account for the remaining 22% error, there are several likely factors both in the model and in physical instrumentation that could have introduced the discrepancy.

8.6.4 Notes on Robustness Under Extreme Conditions

A second field test was attempted several weeks after the very mild cooling season test conditions of late September to examine the robustness of the rule extraction technique. By the time these tests were conducted, the mild late summer weather had dramatically shifted into a more wintry pattern. Ambient conditions were well outside the regime where any normal building would

require natural ventilation, or any other form of cooling: nighttime lows dipped below freezing and daytime highs lingered in the single digits, Celsius. Per the established CART rules, ventilation was enabled, the TABS system ran unnecessarily, and comfort suffered dramatically (Figure 8.14).

The case bears mentioning because it illustrates a pitfall of the rule extraction approach to near-optimal control: extracted rules do not perform well when applied to conditions outside of those experienced in their training set, i.e. those applied to the offline MPC solution. The original optimizer solution was executed under mild swing season and warm cooling season conditions, with temperatures ranging from 7 to 31 °C. The resulting rules are only viable for similar weather conditions. By examining Figures 8.6 and 8.8, one can easily envision situations in which ventilation and TABS cooling would be enabled at even colder temperatures. The cluster resampling CART rule used for the TABS system is particularly vulnerable to this problem, because nowhere in the collection of eight solution profiles is there an "off" profile. Under this rule, the TABS system must be operated at some level regardless of extreme conditions. Naturally, the rule could easily be modified with some commonsense override conditions (e.g. based on ambient conditions or season) that would avoid this problem.

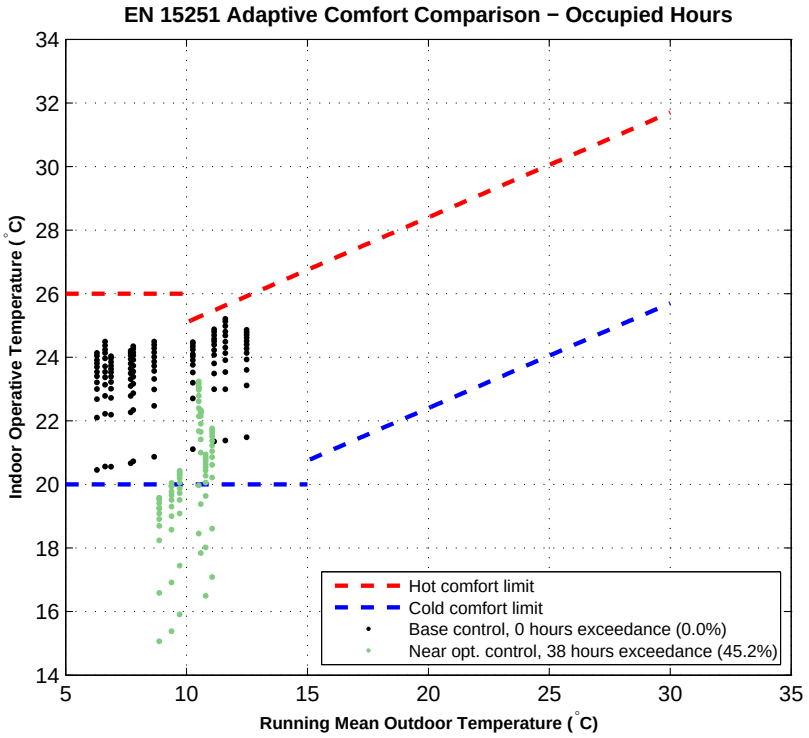
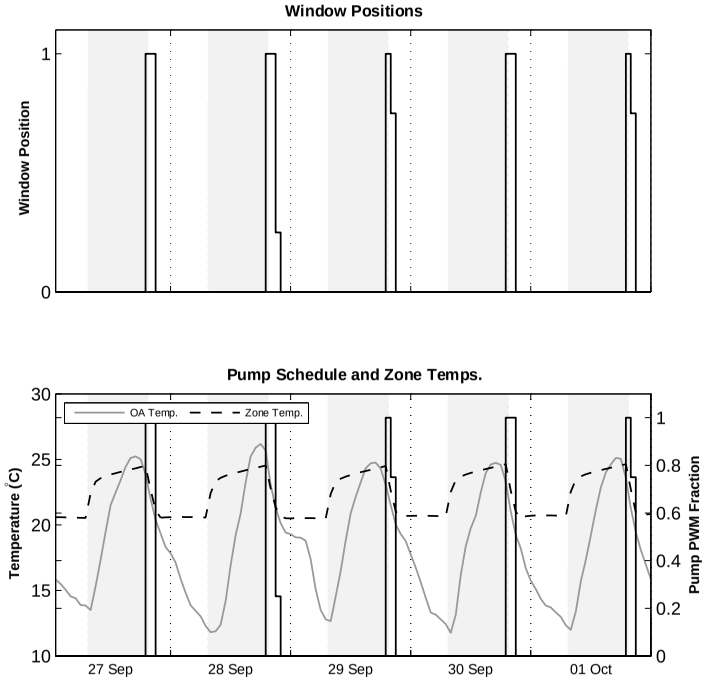


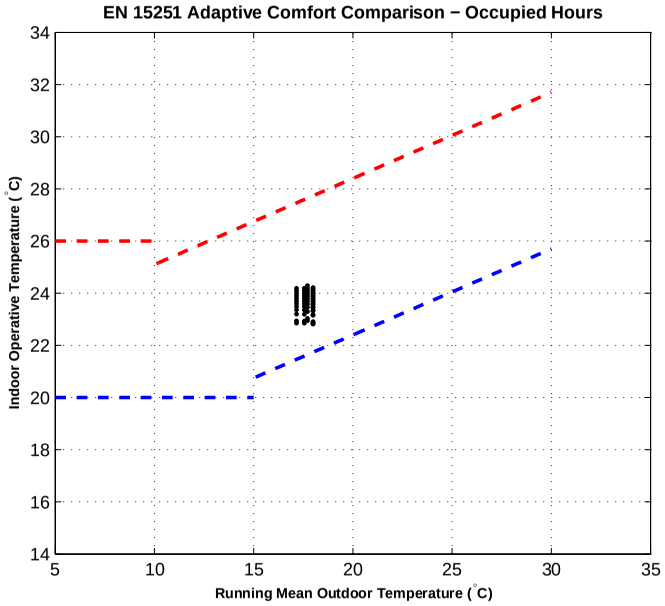
Figure 8.14: Base and near-optimal comfort results for a heating season test.

8.6.5 Simplified Heuristics

Although there was insufficient time to develop and test simplified heuristics in the physical test cells, we can at least wager a guess at their performance using the calibrated simulation model. The simplified night ventilation and TABS control heuristics described earlier in this chapter and used in the reference case model were rerun using measured weather sequences (the September 27 to October 1 sequence measured during the test of the near-optimal control) and a fully calibrated model. While the simple heuristics in this case maintain perfect comfort, they are also extremely timid in pre-cooling the space through night ventilation and result in higher average daily HVAC energy consumption (about 1,350 Wh/day) than with both the simulated and measured near-optimal rules which consumed 607 and 781 Wh/day, respectively. Figure 8.15 illustrates the much briefer periods of night ventilation and very limited exploitation of the EN 15251 adaptive comfort envelope. Despite the aforementioned sensitivities to forecast errors and model mismatch, the near-optimal extracted rules still outperform the heuristics. Additional tuning of the heuristics would be recommended to achieve more aggressive pre-cooling and lengthened periods of natural ventilation. This would likely require decoupling the setpoint temperature of the TABS system from the nigh ventilation system and allowing the night ventilation controller to cool to deeper zone temperatures.



(a) Solution pattern



(b) Operative temperatures and comfort envelope

Figure 8.15: Solution pattern and comfort analysis for calibrated test cell model, operated using simple night ventilation and TABS control heuristics and with measured weather sequences.

8.7 Conclusions

An experimental test was undertaken to test and validate the rule extraction near-optimal control technique explored throughout this work. To the author's knowledge, this was the first test of its kind in the world, and as one would expect, not all aspects of the test ran perfectly. The greatest loss of control in the experiment was the unintentional mismatch between internal gains between the original MPC runs and the experiment, which has been shown to account for a great deal of the discrepancy between experimental and simulated performance. Due to drastic changes in weather and the eventual dismantling of the TABS test facility at the ISE, a follow-on test that controls for all factors is no longer possible.

Nevertheless, we have distilled a handful of useful learnings from the tests and follow-on simulations/optimizations:

(1) **Feasibility** - Tests demonstrated at a bare minimum that simplified rules extracted from offline MPC simulations can be used to approximate the performance of online MPC without the need for "live" optimization of building models. Rules can be extracted in a form that is easily programmable using simple control scripting languages and add minimal additional computational overhead to the BAS. This proof of concept is the first small step toward realizing the ultimate goal of this research, which is the broader implementation of near-optimal control strategies in actual facilities.

(2) **Model Mismatch Sensitivity** - Both tests and simulations demonstrated the sensitivity of solutions to model assumptions. The unintentional halving of internal gains in the tests meant that zone temperatures hovered extremely close to the lower boundaries of the comfort region and that the TABS system was operated more than necessary.

(3) **Forecast Variable Sensitivity** - Disagreement between the simulated near-optimal rules and the experimental data further suggest that rules based on simple forecast variables like max/min temperatures are naturally just as susceptible to forecast uncertainty as any

predictive controller that utilizes weather forecasts as inputs.

(4) **Robustness** - Common sense would dictate that a rule based on heating season conditions should not be applied to the cooling season and vice versa. Additional measurements confirmed this by demonstrating the poor performance of ventilation and TABS rules—originally trained for use under swing or cooling season conditions—under heating season weather.

The tests overall highlighted the need to further "tune" rules for performance robustness. One way to accomplish this might be to simply construct rules from a broader training set, perhaps training rules on multiple seasons or using a parametric study to bracket different expected internal gains values. Of course, attempts to parameterize the training set space echo the findings of Coffey and the lookup table approach [29].

Given the observed sensitivity to parameters with a high degree of uncertainty like weather forecasts, it could be advantageous to develop train sets using **stochastic MPC**. Unlike the deterministic model formulations used in this research, stochastic MPC allows for optimization of control in the presence of stochastic processes like occupant behavior or weather. Introduction of stochastic processes into the optimization loop clearly portends a significant increase in computational time, but it could ultimately improve rule performance in physical settings. Stochastic MPC could be implemented in two ways: one could either identify optimal control vectors for each realization of a process, potentially yielding hundreds or thousands of optimal control vectors, or one could simply identify the single robust control vector that yields the lowest mean objective function value for an ensemble of potential disturbances. The latter approach would be recommended in the rule extraction framework discussed here, because in theory it would allow for extraction of a single rule that provides near-optimal performance in the presence of uncertainty rather than multiple rules for various edge cases.

While the aforementioned process would still yield a single optimal control vector that achieves best overall performance for the ensemble, the states in the ensemble would still vary

depending on which realization of a process is selected. How does one select a single case from the ensemble upon which to train the rule? One might base the training set on a mean or median set of states, or one could use dimensional reduction techniques like principal component analysis (presented in a different context in Chapter 5) to effectively "collapse" states across the ensemble into single values, yielding a training set comprised of principal components.

Yet another alternative to improve overall robustness of extracted rules could be a hybrid between traditional supervisory BAS schemes and the rule extraction approach. Due to the easily digestible structure of the rules, controls engineers or facilities operators could also modify a rule such as the TABS rule presented above to include edge cases not considered in the training set. For example, one might include a "heat wave" solution profile in which TABS operation would be required on a full or half-time schedule. Similarly, a "cold spell" solution category might be added to shut down the TABS for brief periods of colder weather. Fortunately, all of the aforementioned work-arounds can easily be tested with a building energy model to assess approximate performance prior to field implementations.

Chapter 9

Conclusions, Discussion, and Outlook

Despite the diverse array of methods employed in this research, its aim has always been quite simple: explore and demonstrate advanced control techniques applicable to today's high performance buildings that not only provide meaningful energy savings over conventional control schemes but also have potential for uptake in the marketplace. The United States Green Building Council (USGBC) research project that funded much of this research originated from the desire to apply MPC techniques to the study of MM building controls. The presence of passive thermal energy storage in MM designs coupled with their unique comfort considerations provided a unique, challenging, and novel MPC application area. But as the computational complexity of the task grew and the challenges of real-time MPC implementations became apparent, another more fundamental question evolved out of the scope of that original work. USGBC had tasked its project team to develop generalized, near-optimal guidelines for the operation of MM buildings based on the results of offline MPC simulations of typical systems. Clearly statistical methods and data mining had a central role in this analysis, but rather than merely utilizing advanced statistical methods as a diagnostic tool for identifying meaningful correlations in the solution sets—as in prior research by Henze et al. [49, 45]—we asked the question, could such data analysis techniques be used to directly formulate useable control logic? This chapter examines the broader research findings within this context and provides recommendations for future explorations that may help to more fully answer this very broad question.

9.1 Summary of Key Findings

This dissertation first examined several relevant veins of the literature that intersect with MPC and rule extraction. A review of academic and practical experience with MM building designs showed that controls are a crucial and yet understudied design element. MPC and its application to thermal storage problems in buildings was also briefly reviewed, providing a flavor for past approaches and establishing precedent for the offline MPC approach utilizing white-box models. Having established the ultimate motivation for rule extraction, namely to provide near-optimal supervisory control with rules that could easily be scripted in a BAS, the nascent literature on rule extraction from water management research was examined. Coffey's related lookup table approach to approximating MPC was also discussed. Finally, the theory of thermal comfort, both static and adaptive, was covered to explore this important consideration in applying MPC to MM buildings.

Chapter 3 laid out the research "plan of attack" and provided a framework for the particular MPC and rule extraction techniques employed. Chapter 4 reviewed several early validation exercises conducted to tune optimizer performance and examine the reasonableness of MPC solutions. This chapter also provided a brief but important introduction to the concept of solution equivalence, which features prominently in all of the setpoint optimizations throughout the research.

Chapter 5 provided the first real proof of concept of the rule extraction approach in buildings by examining the skill of three different techniques—GLMs, CARTs, and adaptive boosting—in reproducing offline MPC solutions for a simple, binary window opening problem in a MM building. Results of this study showed that two of the techniques in particular, CARTs and boosting, were highly capable in reproducing optimizer energy savings, yielding 85–90% of the optimal results. Experience with rule formulation through this study also demonstrated the importance of closed loop tests in estimating rule performance, as open loop tests often provided more "optimistic" assessments. Results presented in the chapter have been published in two articles [70, 69].

An offline MPC simulation study was conducted on a suite of four prototypical MM buildings modeled in EnergyPlus, presented in Chapter 6. The study effectively compares non-MM

buildings (base case) and "conventional" MM buildings (reference case) against offline MPC results, using MPC as a benchmark against which to measure existing control heuristics as well as mean occupant behavior. Considerations for occupant thermal comfort under a variety of comfort interpretations (ASHRAE 55-2004 static and adaptive as well as EN 15251:2007 adaptive) were introduced through comfort penalties on the objective function. During the cooling season, conventional MM buildings consistently outperformed non-MM buildings, but MPC solutions, particularly those allowing for adaptive comfort, afforded even deeper energy savings with equal or improved comfort. Occupant window opening behavior was shown to be particularly problematic in cooler climates (e.g. San Francisco) or in climates with large diurnal temperature variation (e.g. Boulder, CO) due to a propensity to open windows even when outdoor temperatures fell below conventional heating setpoints. Overall, reference designs that provided space conditioning through radiant cooling, UFAD, or that utilize window-HVAC interlock—effectively systems that do not attempt to provide full zone air mixing with low-temperature supply air while occupant windows are open—approached optimizer performance more closely than other systems investigated. MM3 and MM4 represented the most robust designs with near-optimal performance. In many of the buildings it was found that control of operable windows during occupied periods was not crucial to achieve optimal performance; rather, one simply needed to ensure that zone cooling setpoints were adjusted to "work around" the behavior of occupants. Not surprisingly, this suggests that changeover/interlock strategies are often the optimal design, although they may not be practical or economical in all situations.

Though interesting independent of rule extraction, the simulation study also provided more realistic and complex cases upon which to apply the rule extraction techniques developed in Chapter 5. Chapter 7 examined the control of zone cooling setpoints and night ventilation based on rules extracted from the MM2 and MM3 simulation cases. A wider array of building states, namely air-side system states, were required to successfully train rules for cooling setpoints, and introduction of expert knowledge was required to ensure that highly interdependent sets of rules did not cause unstable operation in closed loop testing. Rules were generally capable of recouping all optimizer

energy savings under closed loop tests, though with increased comfort violations. In cases where there was a narrow margin for improvement between conventional heuristics and the optimizer, such as in MM3, extracted rules provided little benefit and often resulted in worse performance than commonsense heuristics. Follow-on robustness testing indicated that rules were highly sensitive to changes in facility behavior, which was examined by testing Boulder-based rules on alternate climates as well as applying rules developed on the MM2 building to MM3. Although not exhaustive, these investigations demonstrate the care with which extracted rules must be applied and the need for future research into improving their robustness in the presence of a wider range of disturbance variations.

The theme of rule robustness was echoed again in Chapter 8 in which experimental results were presented from a September 2011 test of extracted rules developed for a TABS-cooled test cell. As with previous cases, comfort-penalized offline MPC solutions were used as the basis for the rules' training set, and rules were extracted using the CART methodology. Offline MPC investigations sought to optimize ventilation and pump operation (via PWM fractions). A slight modification to the rule extraction technique was made to preserve typical daily pump PWM profiles. Daily PWM profiles from the training set were binned according to k-means clustering, and then a CART was used to predict the solution profile to be executed on a given day. This technique proved effective in closed loop tests, yielding upwards of 90% of optimizer savings and providing total energy savings of nearly 80% compared to a base case control strategy. Daily savings in the experiment averaged only 40%, and it was determined that over half of the degradation in savings was caused by imperfect weather forecasting (PWM rules were particularly sensitive to one- and two-day-ahead weather forecasts). In other words, uncertainty in weather forecasts, a problematic element of other predictive control schemes, also had a significant influence on the extracted rules.

9.2 Lessons Learned, Limitations, and Notes on Application

This research spanned several different investigations, each fraught with its own difficulties and dotted with successes that do not fit neatly in the the narrative of the results. Some thoughts on

improving process, limitations to the chosen approaches, and suggestions toward future applications are discussed below by topic.

9.2.1 Offline MPC Using White-Box Building Models

The MPC framework utilized throughout this dissertation couples Matlab to EnergyPlus simulations (Figure 3.2), treating EnergyPlus as a black-box cost function evaluator (i.e. Matlab is "blind" to the constitutive equations and relationships comprising the energy model). In Chapter 1, this formulation was shown to be desirable, at least in offline simulation studies, for a number of reasons. For one, this allows MPC to be conducted using validated energy modeling tools. In a design context, this might allow a controls engineer or energy modeler to investigate optimal control strategies using the same hourly simulation models already developed by a mechanical design team (although likely simplifications would be required for speed). In the context of rule extraction, use of white-box models preserves complex, non-linear phenomena such that training sets contain the most realistic physical relationships possible. After all, some reduction in fidelity is already anticipated through the rule extraction process itself. Additional linearization of building responses prior to this step would hamper performance before analysis had even begun.

Still there are obvious inconveniences with the white-box modeling approach, foremost of which is speed. As alluded to throughout this dissertation, investigations can be painfully slow even on modern computing equipment, with optimizations requiring thousands of simulations and the better part of a day to converge for a single 24-hour planning horizon. The two main bottlenecks in the process are the number of individual simulations required to converge on an optimal solution and the runtime per simulation. The obvious computational solution to circumvent these bottlenecks is parallelization. Unfortunately, since energy simulations inherently require a serial solution algorithm, there have been no reasonable proposals to date on parallelizing their operation. The obvious target for parallelization in this research was in the optimization stage, by distributing individual calls to the energy model to a number of separate processing units. This step was absolutely essential in enabling longer simulation periods and exploration of more complex cases

with two and three separate decision vectors.

Despite the current inability to parallelize individual calls to energy simulations, there are still several techniques that, when combined, can dramatically reduce runtimes. When using hourly programs like EnergyPlus, it is extremely important to disable any sizing algorithms that might be used to auto-size systems in the design phase. All such systems should be manually "hard-sized." Secondly, variable outputs written by the simulation program should be kept to the bare minimum required to compute the cost function. Once an optimal decision vector is identified, additional output variables can be obtained by rerunning the model with the optimal policy. Finally and most obviously, models should be kept as simple as possible while still preserving the physical phenomena of interest in the building. In this research, this was primarily accomplished by reducing a three-story model to a single floor.

One feature currently lacking in many energy simulation programs is the ability to explicitly prescribe initial conditions for various building thermal states. A state vector is made available in more systems-focused environments like TRNSYS, but leading government-sponsored building simulation engines like DOE-2 and EnergyPlus currently lack this feature. Recall that the absence of state initialization in EnergyPlus led to the use of the initialization horizon approach described in Chapter 3. In the cases investigated, the initialization horizon comprises 70–90% of the simulation time and a proportionate share of computing time. In other words, offline MPC performed with these tools could experience 70 to 90% speed increases simply by allowing state initialization as opposed to the current "warmup period" approach. In some applications, this single modification could even make real-time MPC applications with tools like EnergyPlus computationally tractable.

9.2.2 Offline MPC Investigations of MM Buildings: Lessons Learned and Areas for Future Research

The offline MPC simulation study of MM buildings yielded a number of interesting insights on the operation of this relatively new building type, the details of which have been summarized in Chapter 6, but the one theme that resonated across all building types and climates was the

need to account for occupant operation of windows. We saw under certain mild conditions how occupant window operation had little effect on building energy use, so long as the mechanical system was able to allow sufficient free running operation during NV periods (enabled either through interlock/changeover or through cooling setpoint setups in the partial changeover cases). Similarly, we also saw how occupant window opening behavior could contribute to significant increases in heating load in certain climates without allowing additional room for the facility to float during these periods. Some of the cases investigated—namely MM3 and MM4 which use UFAD and radiant cooling, respectively—were less sensitive to the potentially greater loads introduced by the occupants, and were thus able to achieve cooling performance and comfort much closer to the optimizer. Across all buildings, the interlock-enabled changeover variants showed near-optimal performance for similar reasons: mechanical systems are not burdened by excessive cooling loads if occupants should choose to open windows at times when ambient temperatures are warm enough to increase zone cooling loads.

Early in the course of the research, a critical decision was made to treat occupant behavior in a deterministic manner. As the first application of MPC to MM buildings, it was decided that the introduction of stochastic behavior models, necessitating stochastic MPC techniques, would add complexity and computational burden to already formidable research objectives. The mean behavioral response approach described in Chapter 3 was then adopted, and a modified version of the Humphreys window behavior model incorporated as a fundamental component of the reference case energy models. Although a necessary simplification for this project, the results of this research by no means suggest that occupant behavior should ideally be treated in this manner. To the contrary, our results demonstrate a clear sensitivity of solutions to occupant behavior, and it is strongly recommended that behavioral uncertainty is considered in future research and even in design of MM buildings, when practical.

Figure 9.1 provides a simple illustration of the variety of model energy use responses one might expect when treating occupant window opening behavior in true stochastic fashion. The MM1 reference case model was evaluated on a week of Boulder, CO weather spanning August 1–8

of the TMY weather file, using the original, stochastic interpretation of the Humphreys model. In this implementation, window openings are determined when the probability of window opening generated by the algorithm, p_w, exceeds a random, uniformly distributed value from 0 to 1. An ensemble of 100 realizations of potential window opening behavior was run, each with a different random number seed value and slightly different window opening patterns. The mean response follows the central tendency of the distribution, as expected, but this naturally does not reflect all expected variation. The results show small variations in gas use (to be expected, as this is a summer week), but a wider distribution of electric consumption, ranging from 646 to 694 kWh. The range of potential weekly energy use values may seem small in this particular case, but consider that we have not included other stochastic influences in the model, particularly occupant presence. When optimizing mechanical system control states in the presence of occupants, even this small range will have an influence on the topology of the objective function.

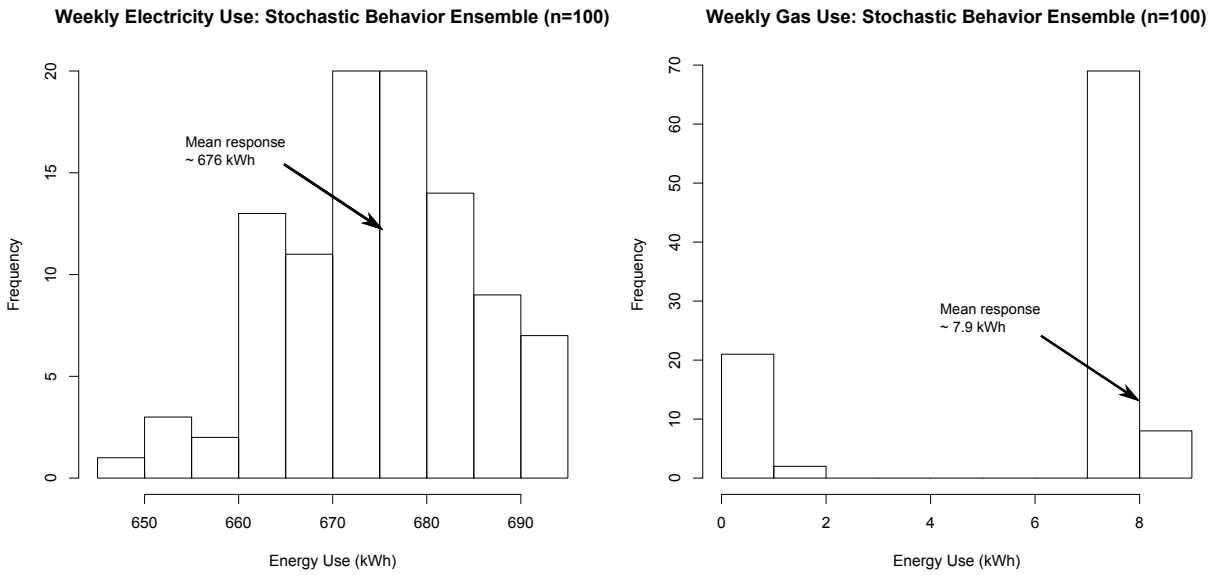


Figure 9.1: Histograms of electricity and gas use for a summer week, using an ensemble of 100 realizations of occupant window opening behavior per the Humphreys algorithm.

Fortunately, a directly related ASHRAE-funded research project (RP-1597, Stochastic Control Optimization of Mixed Mode Buildings) is examining this very issue by applying MPC in the

presence of uncertainty in occupancy as well as a variety of occupant behaviors related to window and shading device operation. Continued research is still essential to characterize adaptive behaviors in MM buildings, as opposed to free-running buildings that form the basis of most research to date.

9.2.3 The First Small Steps With Rule Extraction

The effort required to develop reasonable training sets for rule extraction limited the breadth of actual rule extraction exercises that could be conducted under this dissertation. However, through the cases examined, a basic proof of concept for the technique was provided, some guiding principles were established, and the technique was applied to increasingly complex control situations. The rule extraction theme in the research, of course, culminated with a physical experiment that demonstrated some of the natural limitations of the approach, most prominently that extracted rules only reflect the conditions and MPC solutions present in their training sets. If MPC solution sets are only produced for summer weather sequences or low internal gain scenarios, one can expect rules to reflect these limitations as well.

Extracted rules as envisioned here operate as open loop/feed-forward controllers and generally should be applied in scenarios appropriate to this type of control, such as reset schedules. This research actually was successful in developing rules that directly governed a control action, as was the case for the simple window opening problem presented in Chapter 5 or the night ventilation rule developed for MM3 in Chapter 7. These rules, however, governed very simple binary control actions predominantly based on ambient air conditions, and since mechanical systems were always available as a backup cooling option, thermal comfort was never entirely in jeopardy should the rules have been in error for extended periods. However, it would not have been recommended to develop extracted rules for VAV dampers or three-way heating/cooling coil valves, as such controls are directly responsible for maintaining zone and system setpoints and are best governed by closed loop/feedback control. At this stage, one could only recommend applying rule extraction at the supervisory control level.

9.2.4 Extending Approach to Common Commercial Buildings

Although the cases examined in this research pertain to a small number of high-performance building prototypes, the techniques used could be applied to the broader commercial building stock with some modifications. The issue of occupant comfort in most commercial buildings is, fortunately, much more straightforward than in MM buildings: the lack of operable windows means that comfort currently can only be evaluated using static models. This removes some of the ambiguity associated with comfort evaluation in MM buildings. Unfortunately, it also provides a narrower range of psychrometric conditions under which 80% comfort acceptance can be achieved. Energy-only optimizations would yield limited savings under these comfort considerations, as shown in this research, and a more complete picture of costs, including demand charges, might be the only economically viable problem formulation.

The approach throughout the case studies was to examine energy-optimal operation with comfort penalties, ignoring other energy-related operational costs like monthly demand charges or other utility tariffs. Incorporation of demand charges into the objective function should drive greater pre-cooling and enable significantly greater cost savings, as these charges are often on par with the cost of energy itself. Prior research by Henze et al. demonstrated potential for 10–27% utility cost savings through optimal pre-cooling control in the presence of actual utility tariffs (including time-of-use rates and demand charges). Savings on this order were observed across a range of building types and climates, even under more restrictive static comfort constraints [49, 45]. Solutions incorporating demand charges and time-of-use rates are expected to exhibit sensitivity to local utility rate structures. The resulting enhanced emphasis on pre-cooling will also likely increase the significance of day-ahead predictions (particularly involving extrinsic factors like weather) in solutions and resulting rules.

9.3 Outlook

9.3.1 Directions for Research

From the standpoint of the author, rule extraction still shows promise as a potentially viable means for achieving near-optimal control, in the same class as Coffey's alternative lookup table approaches [29]. This dissertation provides one small step in this direction, and additional research is warranted. Based on the results as they exist today, the following potential research investigations are provided to inspire future directions of rule extraction research in buildings:

- Based on parameter sensitivities identified in Chapters 7 and 8, rule robustness should be explored through a sensitivity analysis that examines the impacts of parameters, such as internal gains, extreme weather, forecast uncertainty, and occupant behavior.

- Determine the extent to which rules can be generalized to large classes of buildings. For example, one could develop a pre-cooling strategy for a standard office building type (rooftop AHUs, VAV air distribution, hot water reheat coils, etc.) and apply the resulting rule to a collection of different permutations of the original building, assessing any performance degradation. Such research could ultimately help inform near-optimal guidelines for conventional buildings.

- Rules may be used directly for supervisory control, but another potential application is to employ near-optimal rules as seeding algorithms to speed convergence of online MPC. Extracted rules could be developed to forecast daily solution profiles (as examined in Chapter 8). The resulting predictions could then be fed into the chosen MPC optimization algorithm as a seed vector. If the seed vectors produced by extracted rules are close enough to optimal, local pattern searches (e.g. Nelder-Meade simplex) might be sufficient to identify the real optima as opposed to more computationally expensive global optimization techniques. Utilizing extracted rules in this manner could enhance the speed and feasibility of certain online MPC applications. An illustration is provided in Figure 9.2.

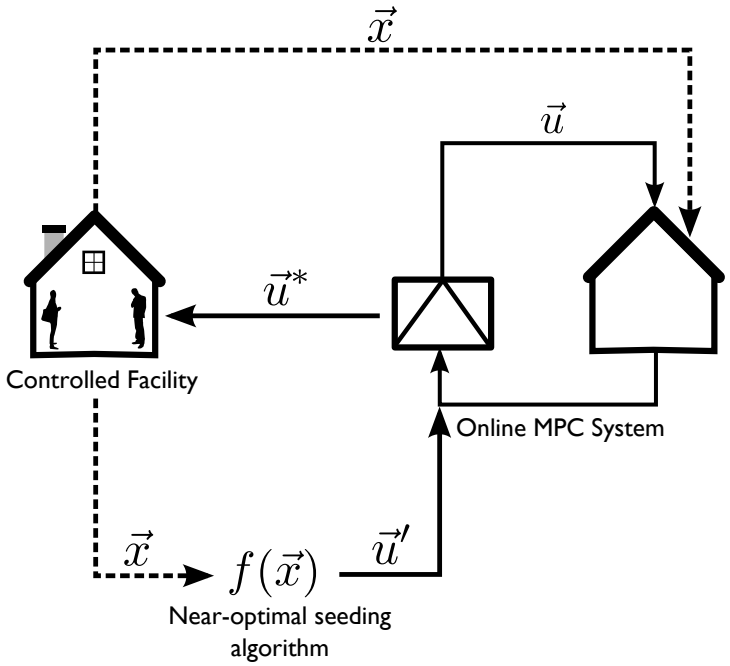


Figure 9.2: An extracted rule is used to provide near-optimal seed values to an online model predictive controller.

- Given the previous notes on occupant behavior influences, rules could be developed based on the results of stochastic optimization, providing a general heuristic that best reflects optimizer behavior under a range of potential disturbances.

- Great success was achieved utilizing a handful of existing data mining techniques for rule extraction; however, data mining and machine learning techniques are evolving at a rapid pace, and many other applicable rule extraction techniques exist. Future research might examine the effectiveness and practicality of other supervised learning approaches such as support vector machines, bagging trees, or random forests, to name a few. As these techniques are not commonly used in the HVAC engineering field, inter-disciplinary collaboration with computer scientists or statisticians is recommended.

- This dissertation only contained limited physical experimentation, and the experiment conducted was applied under highly controlled conditions. This is an important first step, but future research needs to examine performance in "live" facilities, with real occupants (and facilities managers) in the loop. Such a study would ideally include surveys of occupant satisfaction and would likely involve a great deal of collaboration with building engineers and controls contractors to develop rules in a form that not only achieves near-optimal performance but is acceptable to those responsible for programming and maintaining the building and its BAS.

9.3.2 Challenges and Directions for Application

Assuming that some of these key technical questions can be addressed and the technique more thoroughly vetted as a viable approach to near-optimal control, some important practical concerns remain before rule extraction can be implemented in a commercial setting—that is, assuming that online MPC does not evolve into a more turnkey process in the interim.

9.3.2.1 Practitioner Perspectives on Rule Extraction

A cornerstone of this work has been the desire to simplify online MPC, partly to overcome speed issues but also to circumvent potential concerns by building operators that MPC methodology is too complex, opaque, or insecure to implement. Of course, the sentiment of building operators and HVAC controls practitioners is difficult to glean from the available literature, so a brief survey of several industry professionals was conducted to gauge whether the proposed rule extraction approach really might have significant appeal over online MPC.[1]

Those surveyed agreed that, even if computational barriers are alleviated, online MPC will continue to face very high barriers to adoption by facilities managers. There will be very real security concerns due to the required link between the BAS and the model predictive controller. Rule extraction handily circumvents these issues by enabling direct embedding and implementation in an existing BAS; however, it still adds complexity to the BAS, and this complexity could be problematic with certain building operators. According to one of the professionals surveyed, many facility operators cannot effectively manage their existing legacy systems, let alone novel control strategies like MPC or rule extraction. If such issues did not exist, there would not be such a large opportunity for retro-commissioning in commercial buildings. While this does not necessarily eliminate the possibility of implementing approaches like rule extraction, the perceived complexity of the approach may stimulate resistance by overburdened building operators that simply do not need "another fire to put out." Given the academic manner in which rule extraction is presented in this dissertation, the technique would likely stand as much of a chance as online MPC with most facilities managers.

Still, practitioners felt there was reason for optimism. Taylor Engineering, as an example, has investigated a form of manual rule extraction applied to chiller plants [57], so Steven Taylor was confident that extracted rules could be applied, provided that the client understood their basic intent and gained trust with the system. The key for lowering barriers to entry, according to

[1] Professionals surveyed included Peter Curtiss (Principal, Curtiss Engineering), Jeremy Rivera (Principal, Jeremy Rivera Consulting LLC and consulting engineer, Quest Energy Group), and Steven Taylor (Principal, Taylor Engineering, FASHRAE, and current chair of ASHRAE TC-1.4: Control Theory and Application).

Peter Curtiss, is to "keep your methods and procedures simple enough—at least in presentation to a facility manager—so that they would stand a chance of getting implemented." This might mean providing the facility manager with a few high-level "switches" to guide operation of the extracted rules, while deemphasizing the low-level control logic. It could also involve veiling the process in less academic language. For example, "a customized controller for your facility" is much more approachable to practitioners than "an approximation of optimal control discovered through supervised learning techniques." The careful crafting of marketing language and thoughtful consideration of user requirements may prove as important to acceptance among building operators as the technical merits of the approach.

9.3.2.2 Economic Feasibility

One remaining potential barrier preventing widespread adoption of this technique would be cost. Of course, re-commissioning a control system in any manner requires some cost simply for reprogramming of the BAS, documentation, etc. Rule extraction would conceivably have an advantage here over, say, real-time MPC, because a hardware link between an MPC server or cloud computing service would not be required. However, this comes at the expense of significantly greater upfront computational cost which, aside from the time required, was a "free" resource during this research. In a business setting, offline MPC runs would likely be conducted for a specific facility on a dedicated computing facility (this might be realistic for a well-capitalized, larger controls firm) or through a cloud computing service.

Cloud computing services are potentially the more practical option because of their easy expandability, outsourced maintenance and administration, and increasing affordability. Coffey examined this question in his recent dissertation work, mapping out the number of processor-hours and corresponding cost of various optimization run sizes, given different combinations of parameter spaces and individual optimization runtimes (Figure 9.3). Costs were based on prevailing cloud computing service charges from late 2011 of $0.10/processor-hour.

The chart presented in Figure 9.3 was originally meant to apply to lookup table development

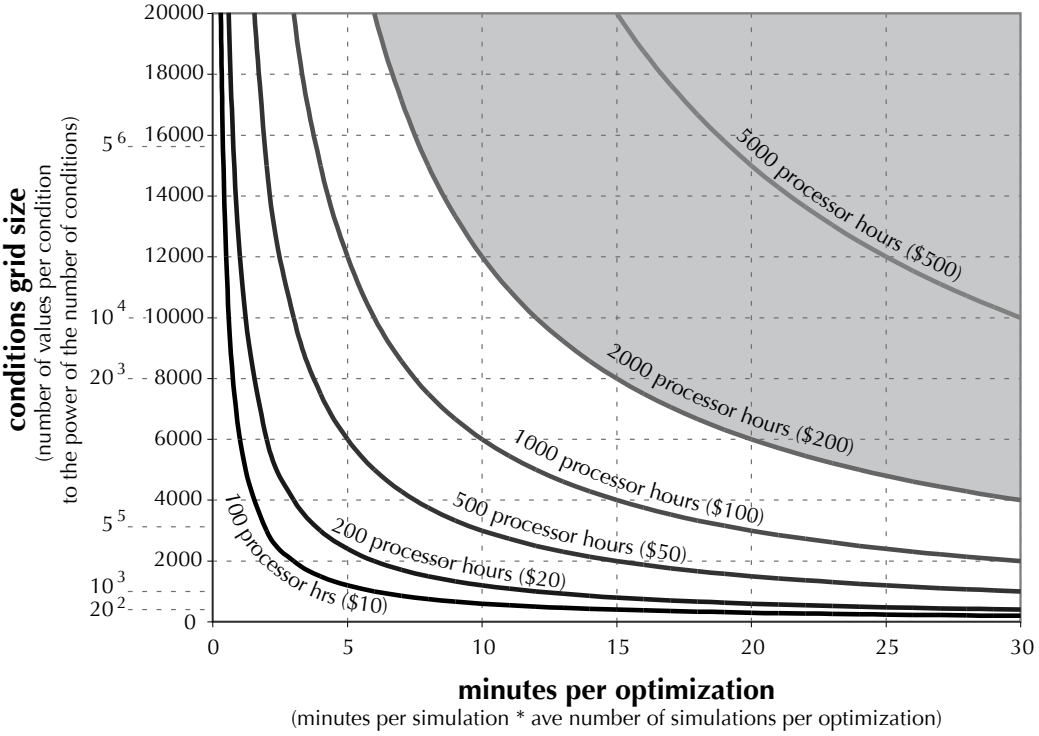


Figure 9.3: Contour lines showing the number of processor-hours of computing resources and corresponding cost of various MPC training runs, originally conceived for developing control lookup tables. The gray shaded reason represents costs that Coffey deemed economically infeasible for small design firms. Source: Coffey, 2011 [29].

and the appropriate sizing of conditioning grids; however, the chart is also instructive in examining how the costs of the rule extraction approach might compare. Due to the initialization horizon approach used in our particular formulation of receding-horizon MPC and the complexity of the models investigated, individual objective function calls ranged from about 3.5s for MM1 to 6s for MM4.[2] Whereas individual optimizations in Coffey's cases might have spanned a day-long planning horizon, optimizations for developing rule extraction training sets were run in receding-horizon fashion over a period of weeks to include consideration of different weather sequences and to ensure that warm-up effects at the beginning of the optimization period could be excluded from the training set. In the end, individual optimizations were orders of magnitude higher than those

[2] Note: this does not include time for file I/O necessary to read output files and generate objective function values, but does represent the vast majority of time required for an individual objective function call.

presented in Figure 9.3, spanning days rather than minutes; however, only a handful of such runs might be required to develop a training set for a given facility, since the weeklong simulation periods span a range of effective grid conditions related to ambient conditions (seasonal scenarios could be run, as in the presented research). The only remaining parameters that might require sensitivity analysis would be internal gains and potentially some variations on occupant behavior. This places rule extraction cases off the righthand side of the chart, but with resource requirements still in the range of 1,000 to 5,000 processor-hours, yielding computing costs of $100–$500. Many such runs could be conducted in parallel fashion, yielding large training sets in a matter of days.

Given the relatively minor cost of computing time even for fairly large training runs, it would appear that the major investment in the rule extraction process would be in staff time to develop appropriate building models, analyze results, and extract/test rules, not to mention installation and commissioning. Despite the availability of clever and effective data mining algorithms, the process is far from automatic and requires a significant degree of expert knowledge at all stages of the process. This expert knowledge and time investment likely will not come cheaply, and one can easily estimate this cost to be two orders of magnitude higher than the cost of computing itself. A conservative estimate would place the total installed cost of even a very simple near-optimal control rule above $10,000, including all of the personnel time mentioned above.

An obvious question follows: given estimated installed costs in excess of $10,000, would rule extraction still make economic sense given the performance advantages demonstrated in this dissertation? This question cannot be answered conclusively from such a limited number of cases, but we can help to place some boundaries around the answer. To achieve a simple payback of 5 years—a rule-of-thumb upper limit on cost-effective energy efficiency measures—an extracted rule-based controller would need to net at least $2,000 in utility savings per year. Since space cooling applications were the emphasis of this dissertation, one could further stipulate that these savings come from cooling electricity reductions during the summer. This is not to imply that opportunities for savings do not exist in other seasons, but merely to build an example directly from cases evaluated in this research. Based on data from the US DOE's CBECS 2003 report [94],

rule extraction would generally be cost-effective for facilities larger than 50,000 sf, assuming that the measure could yield, at a minimum, 20% electric bill savings derived from cooling and ventilation systems. Unless the process could be streamlined, it does not appear that such a measure would generally be cost-effective for smaller facilities. This aligns with expectations, as smaller buildings are generally not the focus of "deep" retro-commissioning efforts.

One particularly easy way to instantly improve the cost effectiveness of the technique would be to alter objective functions to include total utility costs, including demand charges or time-of-use electric rates. Rules would then derive savings from potentially greater utility demand-based incentives and price structures rather than from energy savings alone. Indeed, the "energy-only" approach applied in this research appears to yield large percentage savings across a wide array of buildings and climates. However, one must realize that the solutions here apply to a relatively new and underrepresented style of high performance building, one that incorporates NV and, as a result, is ripe for savings by exploiting broad adaptive comfort envelopes. Most conventional, sealed buildings would not be suitable for this approach and would instead be limited to comfort considerations under the traditional static comfort models. Solutions under static comfort considerations would provide only a fraction of the savings of many of the best results from this research due to their more rigid comfort requirements. As a consequence, use of extracted rules as a demand-limiting strategy may prove to be a more cost-effective application in conventional commercial buildings.

One must also keep in mind that the HVAC engineering community has extensively researched simpler heuristics for energy- and demand-limiting strategies like pre-cooling, and those strategies, although perhaps not as effective in deriving savings for a specific facility, are "shovel-ready" and can yield considerable savings without the need for extensive modeling, simulation, and solution analysis. A "canned" simple heuristic might reduce installed costs by half or more compared to the rule extraction approach, making them applicable to a larger range of buildings from a cost effectiveness standpoint. In certain scenarios, existing heuristics may garner a large fraction of optimal savings and therefore, so rule extraction clearly can only be considered in cases where it maintains a significant energy- or demand-limiting advantage over simplified approaches.

* * *

One wonders if perhaps the greatest energy savings opportunity lies not in more advanced control of conventional buildings, but in re-examination and relaxation of the prevailing static comfort standards around which most buildings' HVAC and controls systems are designed, extending adaptive comfort principles, where appropriate, to the buildings in which the vast majority of office workers spend their weeks and relieving conventional HVAC systems of the burden of "rigid" control that has prevailed for decades. That, however, is a battle to be fought in another dissertation.

Bibliography

[1] S Aggerholm. Hybrid Ventilation and Control Strategies in the Annex 35 Case Studies. IEA, pages 1–31, July 2002.

[2] S Aggerholm. Control of Hybrid Ventilation Systems. International Journal of Ventilation, 1(4):65–75, 2003.

[3] American Society of Heating Refrigeration and Air Conditioning Engineers. ANSI/ASHRAE Standard 55-2004: thermal environmental conditions for human occupancy, 2004.

[4] American Society of Heating Refrigeration and Air Conditioning Engineers. ANSI/ASHRAE Standard 62.1-2004: Ventilation for Acceptable Indoor Air Quality, 2004.

[5] American Society of Heating Refrigeration and Air Conditioning Engineers. The ASHRAE GreenGuide. Butterworth-Heineman, San Francisco, 2006.

[6] ASHRAE. International Weather for Energy Calculations. ASHRAE, Atlanta, GA, 2001.

[7] J Axley and S J Emmerich. A method to assess the suitability of a climate for natural ventilation of commercial buildings. Proceedings: Indoor Air (2002), 2002.

[8] J Axley, S J Emmerich, and G Walton. Modeling the Performance of a Naturally Ventilated Commercial Building with a Multizone Coupled Thermal/Airflow Simulation Tool. Transactions-American Society of Heating Refrigerating and Air Conditioning Engineers, 108(2):1–16, 2002.

[9] N Baker and M Standeven. Thermal comfort for free-running buildings. Energy and Buildings, 23(3):175–182, 1996.

[10] E Bauer and R Kohavi. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning, 36(1):105–139, July 1999.

[11] Fred S. Bauman. Underfloor air distribution (UFAD) design guide. American Society of Heating Refrigerating and Air-Conditioning Engineers, 2003.

[12] F T Bessler, D A Savic, and G A Walters. Water Reservoir Control with Data Mining. Journal of Water Resources Planning and Management, 129(1):26–34, 2003.

[13] J. Bobbin and X. Yao. Automatic discovery of comprehensible control rules by evolutionary algorithms. New Frontier in Computational Intelligence and its Applications, 1999.

[14] J. Bobbin and X. Yao. Automatic discovery of relational information in comprehensible control rules by evolutionary algorithms. In Proceedings of the Third Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems, Canberra, Australia, pages 117–123, 1999.

[15] G S Brager and L Baker. Occupant Satisfaction in Mixed-Mode Buildings. Proceedings of Air Conditioning and the Low Carbon Cooling Challenge, 2008.

[16] G S Brager, S Borgeson, and Y S Lee. Summary Report: Control Strategies for Mixed-Mode Buildings. Technical report, Center for the Built Environment, University of California, Berkeley, CA, October 2007.

[17] G S Brager and R de Dear. Thermal adaptation in the built environment: a literature review. Energy and Buildings, 27(1):83–96, 1998.

[18] G S Brager and R de Dear. A Standard for Natural Ventilation. ASHRAE Journal, 2000.

[19] M R Brambley, P Haves, S C McDonald, P Torcellini, D Hansen, D R Holmberg, and K W Roth. Advanced Sensors and Controls for Building Applications : Market Assessment and Potential R & D Pathways. Technical Report April, Pacific Northwest National Laboratory, 2005.

[20] J.E. Braun. Reducing Energy Costs and Peak Electrical Demand through Optimal Control of Building Thermal Storage. ASHRAE Transactions, 96(2):876–888, 1990.

[21] Leo Breiman. Classification and regression trees. Wadsworth International Group, 1984.

[22] Center for the Built Environment. Mixed-Mode Database. http://cbesurvey.org/mixedmode/database.asp, 2005.

[23] Chartered Institution of Building Services Engineers. CIBSE AM13: Mixed mode ventilation. Page Bros., 2000.

[24] Chartered Institution of Building Services Engineers. CIBSE AM10: Natural ventilation in non-domestic buildings. Page Bros., 2007.

[25] C. Christensen, S. Horowitz, T. Gilver, A. Courtney, and G. Barker. BEOpt: software for identifying optimal building designs on the path to zero net energy. In Proceedings of ISES 2005 Solar World Congress, pages 6–12, 2006.

[26] M Clerc and J Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. Evolutionary Computation, IEEE Transactions on, 6(1):58–73, 2002.

[27] L Coelho. An efficient particle swarm approach for mixed-integer programming in reliability–redundancy optimization applications. Reliability Engineering & System Safety, 94(4):830–837, April 2009.

[28] B Coffey, F Haghighat, E Morofsky, and E Kutrowski. A software framework for model predictive control with GenOpt. Energy and Buildings, 42(7):1084 – 1092, 2010.

[29] Brian Coffey. Using Building Simulation and Optimization to Calculate Lookup Tables for Control. PhD thesis, University of California, Berkeley, 2011.

[30] 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, pages 1–16, January 2012.

[31] G C da Graca, P F Linden, and P Haves. Design and testing of a control strategy for a large, naturally ventilated office building. In Proceedings of Building Simuation 2003, pages 399–406, 2003.

[32] R de Dear. Adaptive thermal comfort in building management and performance. Proceedings of Healthy Buildings 2006. Plenary, 2006.

[33] R de Dear and G S Brager. Developing an adaptive model of thermal comfort and preference. ASHRAE Transactions, 104:1–18, 1998.

[34] R de Dear and G S Brager. Thermal comfort in naturally ventilated buildings: revisions to ASHRAE Standard 55. Energy and Buildings, 34(6):549–561, 2002.

[35] A. Delsante and S. Aggerholm. The use of simulation tools to evaluate hybrid ventilation control strategies. Principles of Hybrid Ventilation, 2002.

[36] M Deru, B Griffith, and P Torcellini. Establishing Benchmarks for DOE Commercial Building R&D and Program Evaluation. In Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings, pages 1–12, Asilomar, CA, 2006. American Council for an Energy-Efficient Economy.

[37] R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2011.

[38] EnergyPlus Development Team. EnergyPlus v6.0, September 2011.

[39] European Committee for Standardization. EN 15251: Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics, 2007.

[40] Fraunhofer Institute for Solar Energy Systems. EnOB: Research for energy-optimised construction, 2010.

[41] Y Freund, R Schapire, and P Vitányi. A desicion-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory - Lecture Notes in Computer Science, 904:23–37, 1995.

[42] J Friedman. Additive logistic regression: A statistical view of boosting. Annals of statistics, 28(2):337–374, 2000.

[43] Jeff Gill. Generalized linear models: a unified approach. SAGE Publications Inc., Thousand Oaks, CA, 2001.

[44] P Heiselberg. Principles of Hybrid Ventilation. Technical report, Aalborg University, Aalborg, Denmark, 2002.

[45] G P Henze, MJ Brandemuehl, C Felsmann, A R Florita, and H Cheng. ASHRAE RP-1313 Final Report: Evaluation of Building Thermal Mass Savings. Technical report, ASHRAE, Omaha, NE, 2007.

[46] G P Henze, C Felsmann, and G Knabe. Evaluation of optimal control for active and passive building thermal storage. International Journal of Thermal Sciences, 43(2):173–183, 2004.

[47] G P Henze, D E Kalz, C Felsmann, and G Knabe. Impact of forecasting accuracy on predictive optimal control of active and passive building thermal storage inventory. HVAC&R Research, 10(2):153–178, 2004.

[48] G P Henze and M Krarti. Predictive Optimal Control of Active and Passive Building Thermal Storage Inventory, 2005.

[49] G P Henze, T H Le, A R Florita, and C Felsmann. Sensitivity Analysis of Optimal Building Thermal Mass Control. Journal of Solar Energy Engineering, 129(4):473–485, 2007.

[50] G P Henze and S Liu. Impact of Modeling Accuracy on Predictive Optimal Control of Active and Passive Building Storage Inventory. ASHRAE Transactions, page 14, July 2004.

[51] G P Henze, J Pfafferott, S Herkel, and C Felsmann. Impact of adaptive comfort criteria and heat waves on optimal building thermal mass control. Energy and Buildings, 39(2):221–235, 2007.

[52] G.P. Henze, R.H. Dodier, and M. Krarti. Development of a predictive optimal controller for thermal energy storage systems. Transactions-American Society of Heating Refrigerating and Air Conditioning Engineers, 104:54–54, 1998.

[53] G.P. Henze and M. Krarti. The impact of forecasting uncertainty on the performance of a predictive optimal controller for thermal energy storage systems. Technical report, Univ. of Nebraska, Omaha, NE (US), 1999.

[54] G.P. Henze, M. Krarti, and M.J. Brandemuehl. A simulation environment for the analysis of ice storage controls. HVAC&R Research, 3:128–148, 1997.

[55] G.P. Henze, M. Krarti, and M.J. Brandemuehl. Guidelines for improved performance of ice storage systems. Energy and Buildings, 35(2):111–127, 2003.

[56] S Huang and R M Nelson. Rule Development and Adjustment Strategies of a Fuzzy Logic Controller for an HVAC System: Part One–Analysis. Transactions-American Society of Heating Refrigerating and Air Conditioning Engineers, 100:841, 1994.

[57] M Hydeman and G Zhou. Optimizing chilled water plant control. ASHRAE Journal, pages 44–54, June 2007.

[58] International Energy Agency. IEA-ECBCS Annex 35 HybVent, 2003.

[59] K. Keeney and J.E. Braun. A simplified method for determining optimal cooling control strategies for thermal storage in building mass. HVAC&R Research, 2(1):59–78, 1996.

[60] J Kennedy and R Eberhart. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, volume 4, pages 1942–1948, Perth, WA, Australia, 1995. IEEE.

[61] J Kennedy and R.C. Eberhart. A discrete binary version of the particle swarm algorithm. In 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, volume 5, pages 4104–4108. IEEE, 1997.

[62] Byung-Il Koh, A D George, R T Haftka, and B J Fregly. Parallel asynchronous particle swarm optimization. International journal for numerical methods in engineering, 67(4):578–595, July 2006.

[63] M Kolokotroni, M Perera, D Azzi, and G S Virk. An investigation of passive ventilation cooling and control strategies for an educational building. Applied Thermal Engineering, 21(2):183–199, 2001.

[64] M Krarti, G P Henze, and D Bell. Planning horizon for a predictive optimal controller for thermal energy storage systems. ASHRAE Annual Meeting, 1999.

[65] M Liddament, J Axley, P Heiselberg, and Y Li. Achieving Natural and Hybrid Ventilation in Practice. International Journal of Ventilation, 5(1):115–130, 2006.

[66] S Liem and A H C van Paassen. Hardware and controls for natural ventilation cooling. Proceedings of Ventilation and Cooling: 18th Annual AIVC Conference, pages 59–68, 1997.

[67] M E Mankibi and P Michel. Development and Assessment of Hybrid Ventilation Control Strategies using a Multicriteria Approach. International Journal of Ventilation, 4(3):227–238, 2005.

[68] M E Mankibi, P Michel, and G Guarracino. Control Strategies for hybrid ventilation development of an experimental device. Proceedings of the 22nd Annual AIVC Conference, 2001.

[69] P. May-Ostendorp, G. P. Henze, C. D. Corbin, B. Rajagopalan, and C. Felsmann. Model-predictive control of mixed-mode buildings with rule extraction. Building and Environment, 46(2):428–437, February 2011.

[70] P. May-Ostendorp, G.P. Henze, B. Rajagopalan, and C. D. Corbin. Extraction of Supervisory Control Rules from Model Predictive Control of Windows in a Mixed Mode Building. Journal of Building Performance Simulation, 2012.

[71] K.J. McCartney and F. J. Nicol. Developing an adaptive control algorithm for Europe. Energy and buildings, 34(6):623–635, 2002.

[72] E McConahey. Mixed mode ventilation: Finding the right mix. ASHRAE Journal, 50(9):36–48, 2008.

[73] Evan Mills. Building Commissioning: A Golden Opportunity for Reducing Energy Costs and Greenhouse Gas Emissions. Technical report, Lawrence Berkeley National Laboratory, Berkeley, CA, 2009.

[74] NEN. NPR-CR 1752: Ventilation for buildings - Design criteria for the indoor environment, 1999.

[75] F Nicol and M Humphreys. Derivation of the adaptive equations for thermal comfort in free-running buildings in European standard EN 15251. Building and Environment, 45(1):11–17, 2010.

[76] F Nicol and L Pagliano. Allowing for thermal comfort in free-running buildings in the new European Standard EN 15251. In Proceedings of the 2nd PALENC Conference, 2007.

[77] BW Olesen. Operation and control of thermally activated slab heating and cooling systems. In IAQVEC, Sendai, Japan, 2007.

[78] BW Olesen and Klaus Sommer. Control of slab heating and cooling systems studied by dynamic computer simulations. ASHRAE Transactions, 108(2):1–10, 2002.

[79] S Olesen, J J Bassing, and P O Fanger. Physiological comfort conditions at sixteen combinations of activity, clothing, air velocity and ambient temperature. ASHRAE Transactions, 78(2):199–206, 1972.

[80] J Pfafferott, S Herkel, D E Kalz, and A Zeuschner. Comparison of low-energy office buildings in summer using different thermal comfort criteria. Energy and Buildings, 39(7):750–757, 2007.

[81] T. Agami Reddy. Applied Data Analysis and Modeling for Energy Engineers and Scientists. Springer, 2011.

[82] S. K. Regonda, B. Rajagopalan, and M. Clark. A new method to produce categorical streamflow forecasts. Water Resources Research, 42(9):W09501, September 2006.

[83] H Rijal. Development of an adaptive window-opening algorithm to predict the thermal comfort, energy use and overheating in buildings. Journal of Building Performance Simulation, pages 17–30, 2008.

[84] H Rijal, M Humphreys, and J Nicol. How do the occupants control the temperature in mixed-mode buildings? Predicting the use of passive and active controls. In Proceedings of Air Conditioning and the Low Carbon Cooling Challenge, pages 1–15, Windsor, UK, 2008. NCEUB.

[85] H Rijal, P Tuohy, M A Humphreys, J F Nicol, A Samuel, and J Clarke. Using results from field surveys to predict the effect of open windows on thermal comfort and energy use in buildings. Energy and Buildings, 39(7):823–836, July 2007.

[86] H B Rijal, M A Humphreys, and J F Nicol. Understanding occupant behaviour: the use of controls in mixed-mode office buildings. Building Research & Information, 37(4):381–396, 2009.

[87] H C Spindler. System identification and optimal control for mixed-mode cooling. PhD thesis, Massachusetts Institute of Technology, 2004.

[88] H C Spindler and L K Norford. Naturally ventilated and mixed-mode buildings—Part II: Optimal control. Building and Environment, 44(4):750–761, 2008.

[89] H C Spindler and L K Norford. Naturally ventilated and mixed-mode buildings—Part I: Thermal modeling. Building and Environment, 44(4):736–749, April 2009.

[90] The Mathworks Inc. MATLAB R2011b, August 2011.

[91] P A Torcellini, M Deru, B Griffith, S Pless, R Judkoff, and D Crawley. Lessons Learned from Field Evaluation of Six High-Performance Buildings. In Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings, pages 1–13, Asilomar, CA, July 2004. American Council for an Energy-Efficient Economy.

[92] United States Green Building Council. LEED-NC for New Construction. USGBC, 2005.

[93] University of Delft. TNO Cp Generator. http://cpgen.bouw.tno.nl/cp/info.asp, 2010.

[94] US DOE EIA. Table E6. Electricity Consumption (kWh) Intensities by End Use for Non-Mall Buildings. http://www.eia.doe.gov/emeu/cbecs/cbecs2003/, 2003.

[95] US Energy Information Administration. Annual Energy Review 2010. Technical report, US Energy Information Administration, Washington D.C., 2010.

[96] A C van der Linden, A C Boerstra, A K Raue, S R Kurvers, and R J De Dear. Adaptive temperature limits: A new guideline in The Netherlands:: A new approach for the assessment of building performance with respect to thermal indoor climate. Energy and Buildings, 38(1):8–17, 2006.

[97] G S Virk and D L Loveday. Model-based control for HVAC applications. In Proceedings of the Third IEEE Conference on Control Applications, volume 3, pages 1861–1866, August 1994.

[98] K Voss, S Herkel, J Pfafferott, and G Löhnert. Energy efficient office buildings with passive cooling–Results and experiences from a research and demonstration programme. Solar Energy, 81(3):424–434, 2007.

[99] Chih-Chiang Wei and Nien-Sheng Hsu. Derived operating rules for a reservoir operation system: Comparison of decision trees, neural decision trees and fuzzy decision trees. Water Resources Research, 44(2), February 2008.

[100] Chih-Chiang Wei and Nien-Sheng Hsu. Optimal tree-based release rules for real-time flood control operations on a multipurpose multireservoir system. Journal of Hydrology, 365(3-4):213–224, February 2009.

[101] Zhang Wen-Jun, Xie Xiao-Feng, and Bi De-Chun. Handling boundary constraints for numerical optimization by particle swarm flying in periodic search space. In Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), pages 2307–2311. IEEE, 2004.

[102] M. Wetter. GenOpt®, Generic Optimization Program. In Seventh International IBPSA Conference, pages 601–608, 2001.

[103] D.S. Wilks. Statistical methods in the atmospheric sciences. Academic Press, San Diego, CA, 1995.

[104] G Zhou, M Krarti, and G P Henze. Parametric analysis of active and passive building thermal storage utilization. Journal of Solar Energy Engineering, 127:37, 2005.

[105] T. Zitzmann, J. Wapler, M. Fischer, and J. Pfafferott. Messkampagne zur Bewertung des Einsatzes von PCM in einer thermisch aktivierten Filigrandecke für den Kühlfall. Masters dissertation., 2010.

Appendix A

Other MM and NV Case Studies

Liem and van Paassen (1997) provided one of the first detailed investigations of control strategies for naturally ventilated buildings, resulting from their collaboration with the pan-European NatVent project. Their work examines ideal control strategies for a naturally ventilated office building through a parametric analysis of a coupled, nodal thermal-ventilation model developed in Simulink. Under normal circumstances, the building's effective opening area (A_{eff}) is controlled by a PI controller with a set point of 22 °C. Several predictive control strategies were examined controlling A_{eff} over nights and weekends. These might involve pre-cooling to offset the anticipated cooling degree-hours for the following day or to meet a specified pre-cooling set point. No one control strategy was found to be most effective, however, the mere presence of any control strategy was found to reduce overheating hours by over 50%. Night cooling strategies as a whole were only found applicable to medium and high "inertia" construction (75 and 100 kg/m^2, respectively) and with limited internal heat gains (22–26 W/m^2 for medium weight and 27–32 W/m^2 for high weight). [66]

Kolokotroni et al. (2001) applied a simple, coupled thermal/airflow simulation to examine the performance (primarily thermal comfort) of a naturally ventilated education building [63]. The building was first monitored over a limited period of time during which pupils were not present. A model of the building was then developed to investigate the effectiveness of various night pre-cooling strategies after calibration against measured data. Modeling of extreme weather events revealed that summer cold spells during the pre-cooling season could result in overcooling if precautionary

measures were not implemented in BMS control algorithms. It was found that even simple control algorithms were capable of achieving good thermal comfort. Similar to the work of Liem and van Passen referenced above, the study found that the specific control algorithm and the precision of its implementation did not matter nearly as much as the mere presence of night pre-cooling control. The authors laid some foundation for future research into the extension of model-predictive control methods to MM and naturally ventilated buildings based on the work of Virk and Loveday (1994) and Huang and Nelson (1994) [97, 56]. Models suggested for consideration include both physical and inverse.

Carrilho da Graça et al. (2004) investigated the MM ventilation control for the upper floors of the San Francisco Federal Building. The building under study consists cross-ventilated office floors, with a combination of automated vents and operable windows manually controlled by users. The authors sought to examine the MM controls and the impact of variations in user behavior. EnergyPlus and its integrated airflow model (`AirflowNetwork`) were used to simulate building performance. Rather than allowing for intractably large numbers of permutations of window openings, window openings were grouped into 10 distinct modes, then organized into an opening mode table. The mode number then became the control variable adjusted by the BMS's control algorithms, which applied rule-based algorithms to coordinate window openings. Considerations for high winds, rain, and overcooling were incorporated into BMS algorithms. To simulate user behavior for the operable windows accessible within the occupied zone, two classes of users were considered: one that behaved in concert with the BMS control algorithms (informed) and a second that opened windows gradually based on perceived discomfort (uninformed). Uninformed users were shown to have a significant impact on the performance of the system due to excessive window opening during warm times of day and a lack of use of slab cooling at night. Indoor temperatures were typically 1K cooler in cases where BMS control of slab cooling was implemented. [31]

Mankibi and Michel (2001), who had earlier developed an experimental test cell for testing of hybrid ventilation controls extended this research by investigating the effectiveness of various controllers via a multi-objective parameter tuning process [68]. Three control architectures (on/off,

PID, and hierarchical fuzzy control) were implemented for a MM ventilated room, then tested for robustness using a sensitivity analysis. Factors such as thermal comfort, energy, demand, and occupant productivity were expressed monetarily and linearly combined to develop the cost functions for each control architecture.

It was found that the manual tuning of the fuzzy controller could reduce total costs by 11% to 15% over the simple on/off and PID controller. Only 3% to 5% of these savings derived from energy, the remainder deriving from thermal comfort and productivity. Sensitivity analysis revealed the fuzzy controller to be inherently more robust in that its performance was less dependent on the specific ventilation implementation chosen. Across all three MM control architectures, the total energy savings realized were on the order of 75% compared to a purely mechanical system. The authors note the need to extend the application of online tuning of the control algorithms through "artificial intelligence techniques such as genetic algorithms or neural networks" to further optimize performance. [67]

The National Renewable Energy Laboratory's (NREL) High Performance Buildings Research Initiative has conducted some of the most thorough quantitative, post-occupancy studies of US buildings that employ natural/MM ventilation. Torcellini et al. (2004) examined the measured energy consumption and operation of six of these buildings for an extended period of time (one year minimum). Of the six studied, three employed mixed-mode or natural ventilation strategies, in addition to other high performance building technologies like ground-source heat pumps, heat recovery, and PV panels. The study used EnergyPlus to simulate the performance of the buildings as designed and compared these results with measured data. Buildings were also compared to base case, minimally code-compliant buildings of the same size, location, etc. Even though buildings were found to generally underperform compared to design, all of the buildings consumed significantly less energy (22% to 77% of source energy) than code-compliant buildings. Although the study makes no particular claims regarding the contribution made by MM ventilation to energy savings, the three buildings with the highest energy savings all employed natural/MM ventilation and achieved source energy savings in excess of 50%. [91]

Liddament et al. (2006) examined the surveyed the operational performance of several dozen buildings with natural/MM ventilation and low-energy cooling through a meta-survey. Buildings included in the study came from the IEA NatVent study, the IEA Low Energy Cooling (Annex 28) study, the IEA HybVent project (Annex 35) previously referenced, the IEA Retrofitting in Educational Buildings project (Annex 36), as well as several other prominent international case studies from Canada, the US, and China. The authors identified several common operational problems spanning a large number of the buildings. Air quality issues such as entrained vehicle pollution, noise, and cold drafts (improper preheating) topped the list. Similar to the findings of previous studies, building components, particularly window actuators and motorized dampers, were found to be unreliable. Components located in hard-to-reach areas, such as an automated vent high in an atrium wall, might be ignored completely if faults occurred due to the difficulty and expense of replacement or repair. Conflicting control strategies were also common in many buildings, such as the operation of perimeter heating systems in the cooling season. Proper commission of these buildings might have identified and eliminated certain of these control issues. [65]

The authors also identified key elements of the most successful natural/MM/passive cooling designs studied. Proper estimation and control of internal heat gains was found crucial to reduce summertime overheating. Night cooling combined with thermal mass was a common and effective strategy, as well as the use of labyrinths or ground tubes to preheat/cool outside air before entering the building. Proper IEQ conditions were best maintained through a use of demand-controlled ventilation, avoidance wherever possible of introducing outdoor pollutants through air intakes, and an airtight envelope that eliminates infiltration. [65]

Axley and Emmerich have published most extensively on the applicability of MM buildings to US climates. In follow-on work to NIST research on the applicability of natural ventilation in US commercial buildings [7], Axley and Emmerich examined the application and control of MM systems for a commercial building constructed in the Netherlands with a multi-zone, coupled thermal/airflow model. The building was comprised of a series of cellular offices and atria, taking advantage of stack and cross-ventilation effects. Occupants in the building were provided with

instructions on the control of vent openings, although automatic control with manual override was maintained for external solar shading and lighting systems. A model of the building was developed in the CONTAM97 coupled thermal/airflow environment. The model was calibrated against measured data from the building site and was then evaluated for applicability in Los Angeles, CA, a much hotter and drier climate. Multi-zone model results (namely temperatures and airflows) for a properly calibrated model were found to correspond very well to local monitoring results for the building, suggesting that natural ventilation and MM design strategies can successfully be modeled in such coupled thermal/airflow simulation environments. However, convergence issues plagued early versions of the multi-zone model, and an alternate model was developed in which the total number of airflow nodes was reduced. Results for the Los Angeles design case indicated that more sophisticated control strategies would be necessary to effectively utilize nighttime ventilation and pre-cooling as a strategy. [8]

Appendix B

Definitions and Taxonomies for MM Buildings

B.1 Defining Mixed-Mode Buildings

Ventilation, in North American practice, simply refers to the process of supplying or removing air from a space to control contaminant levels, temperature, or humidity [4]. It follows, then, that natural ventilation is simply a non-mechanical or passive means of providing ventilation through naturally-occurring effects such as wind pressure on a building faade or stack effects within a building. This definition of natural ventilation is fairly universal, being recognized by several international organizations, including ASHRAE and CIBSE [4, 24].

Mixed-mode ventilation, on the other hand, lacks such a straightforward and universally accepted definition. It has most commonly been referred to as "a hybrid approach to space conditioning that uses a combination of natural ventilation from operable windows (either manually or automatically controlled) or other passive inlet vents, and mechanical systems that provide air distribution and some form of cooling," as put forth by Brager of UC Berkeleys Center for the Built Environment (2007) [16]. Although not yet permanently codified in standards language, ASHRAE adopts a similar definition of mixed-mode ventilation in its GreenGuide (2006) [5]. The Chartered Institute of Building Services Engineers (CIBSE) of the UK has provided the most extensive institutional guidance for professionals on the design and operation of both naturally ventilated and mixed-mode buildings. CIBSE's definition of mixed-mode is also in line with the CBE definition. CIBSE Applications Manual 13 (2000) defines mixed-mode ventilation as "servicing strategies that combine natural ventilation with mechanical ventilation and/or cooling in the most effective manner

maximising the use of the building fabric and envelope to achieve favorable indoor environmental conditions." [23] The US Green Building Council, in its New Construction & Major Renovation Version 2.2 Reference Guide (2007) for the Leadership in Environmental Design (LEED) program describes "mixed-mode conditioning" as a thermal comfort strategy "employing a combination of active and passive systems" [92].

Unfortunately the HVAC research community has further muddied waters by simultaneously introducing the concept of **hybrid ventilation** through European ventilation standards, such as EN 13779, and research projects, such as the IEA Annex 35 HybVent project. The major distinction between hybrid ventilation and mixed-mode is that hybrid ventilation refers only to the building's ventilation design, whereas mixed-mode often refers to mechanical cooling as well. As defined through the Annex 35 project:

> The main difference between a conventional ventilation system and a hybrid system is the fact that the latter has an intelligent control system that can switch automatically between natural and mechanical modes in order to minimize energy consumption. [44]

A hybrid ventilated building, such as those examined in the IEA Annex 35 HybVent project, typically includes no vapor compression cooling systems, although mechanical cooling is not categorically prohibited [44]. An office with fan-assisted natural ventilation is a simple example of this technology.

For the purposes of this research, the author has adopted the broad definition of mixed-mode cooling and ventilation put forth by CBE, which first and foremost describes mixed-mode as a hybrid space conditioning technique. It is this broad definition that most closely aligns with the experience of the USGBC research team members and the surveyed literature.

B.2 What is a typical mixed-mode building?

In practice, when one describes a mixed-mode building, this does not simply imply a conventionally-designed "glass box" structure with operable windows. While the term mixed-mode refers to a specific set of ventilation and cooling strategies, a mixed-mode building's overall performance

and energy savings typically rest on the successful integration of mixed-mode concepts with other high-performance building systems. These buildings often employ natural ventilation alongside other "passive" or low-energy technologies, such as ground-source heat pumps, earth tubes, and superinsulation, to reduce the overall size and energy use of the mechanical system by harnessing low-exergy sources/sinks of energy (ground, nighttime air, bodies of water, etc.) whenever possible.

The International Energy Agency's Annex 35 hybrid ventilation research effort, for example, has acknowledged in its scoping documents that "buildings ventilated by hybrid ventilation often apply other sustainable technologies and an energy optimization requires an integrated approach in the design of the building and its mechanical systems" [58]. Brager and Baker (2008) have also found through a series of recent case studies that, not surprisingly, most mixed-mode buildings utilize mixed-mode as well as other low-energy heating and cooling techniques, coupled with mechanical systems, to meet design goals [15]. They straddle the space in the marketplace between buildings that are fully mechanical in nature and those that are completely passive. They can range from hybrid ventilation retrofits of 1970s office building construction (involving very few passive mechanisms beyond operable windows) to turn-of-the-millennium new construction that incorporates sophisticated lighting controls, solar shading devices, and advanced building facades and glazing.

With such a diverse assortment of existing mixed-mode buildings, how can one hope to develop control guidelines that can be generalized to a wide range of mixed-mode applications? The author and USGBC research team struggled with this question significantly during the early months of research, realizing that the most common mixed-mode categories of "changeover," "concurrent," and "zoned" simply did not capture system complexity when surveying large numbers of buildings. To ensure that simulations studies addressed a representative segment of the MM building population, an informal taxonomy was required to help classify buildings. The sections below provide a review of existing taxonomies for MM buildings as well as a scheme adapted for use in this research.

B.3 Existing Classification Schemes for MM Buildings

A "standard" taxonomy for mixed-mode buildings has developed over the years, starting in the mid-1990s based on the work of Bordass and Leaman and later formalized through CIBSE's AM-13 for mixed-mode buildings [23]. Under the standard taxonomy, mixed-mode buildings are primarily classified by their control types. There are three major classifications: zoned, complimentary, and contingent. Complimentary systems, in turn have a variety of sub-classifications.

In **zoned** mixed-mode systems, the mechanical and natural ventilation systems are separated spatially such that a given zone in the building either be mechanically or naturally ventilated, but not both.

Complimentary mixed-mode systems are designed so that mechanical and natural ventilation systems occupy the same space. Complimentary systems are further broken down into sub-categories which define whether the natural and mechanical systems can operate at the same time. **Concurrent** systems are the only form of mixed-mode ventilation systems in which the mechanical and natural ventilation components can operate both in the same space and at the same time. For example, in a concurrent system, an occupant is allowed to open a window even when mechanical heating or cooling is being provided to the space. In a **changeover** system, natural and mechanical ventilation are not allowed to operate simultaneously. Rather, a supervisory controller determines whether conditions are appropriate for the use of natural ventilation and allows operation to change over into natural ventilation mode. Depending on the environmental conditions, these changes can be made on an hourly, diurnal, or even seasonal basis depending on the system design.

An **alternate** system is effectively the same as a changeover system except that the switch from mechanical to natural ventilation mode is a manual one usually initiated by a facilities manager when seasonal conditions favor natural ventilation.

Finally, **contingent** systems form somewhat of a catch-all category for buildings that could effectively employ mixed-mode ventilation strategies but currently do not. Contingent buildings

can, thus, be considered as buildings that would be suitable candidates for mixed-mode retrofits.

This macro-level classification of mixed-mode buildings has provided a foundational vocabulary through which practitioners and researchers can discuss the broad characteristics of a mixed-mode building. However, such broad categorizations run the risk of oversimplifying. As noted by Brager et al. in a 2007 report, Summary Report: Control Strategies for Mixed-Mode Buildings, the system "is useful for classifying buildings and their operational control strategies as they have been built," but does not fully address the diversity in mixed-mode building design – particularly the natural ventilation implementation – that occurs in practice [16]. One of the most significant criticisms of the current taxonomy offered in the report is that mixed-mode buildings may employ a host of strategies – zoned and changeover, for example – in different areas of a given building, with no one strategy dominating. How then can one assign one unique classification to the whole building?

Brager et al. suggest a significantly more extensive classification framework largely designed to assist in identifying drivers for the design process. Although such an extensive framework may be too detailed for the research purpose at hand, the Center for the Built Environment has formulated many useful attributes that should be accounted for when trying to determine "typical" implementations of mixed-mode buildings. Examples include:

- Consideration of the context in which the building exists, namely its climate and surrounding environment
- Degree of autonomy afforded to occupants in control of ventilation apertures like windows and vents
- Available control inputs
- The function of the control itself, whether purely to regulate ventilation rates, provide space conditioning, regulate structural cooling, etc.

Brager et al. also propose another useful concept in their classification scheme by tracking three

separate "drivers." For example, each summary case study provided in the appendices of the 2007 report provides "mixed-mode strategies at a glance," as shown in B.1. Each strategy is presented in tabular form, and a given building can utilize multiple strategies. A given building may employ both zoned and concurrent operation. It may use a combination of cross-ventilation and stack effect to achieve natural ventilation. This flexibility is highly desirable when classifying such complex systems.

Classification		
Changeover	Concurrent	Zoned
	X	X

HVAC				
GSHP	Panel	Slab	UFAD	Forced Air
		X		X (warehous only)

Controls			
Red/Green notification	Window HVAC interlock	Mechanical window operation	Manual window operation
		X	X

Ventilation			
Windows	Vents	Stack	Cross Vent
X		X	

Figure B.1: Tabular classification of MM buildings. Source: Brager et al. (2007) [16]

B.4 Proposed MM Classification Scheme

Both the traditional Leaman/Bordas/CIBSE taxonomy and the more recently proposed CBE taxonomy have attractive elements. CIBSE's is easy and simply applied; CBE's is significantly more accurate in addressing building complexity. A taxonomy was proposed and vetted amongst a team of mixed-mode and high performance building experts during the early phases of this research (illustrated in Figure B.2. It is an attempt to expand upon the widely accepted CIBSE taxonomy through some of the very useful additions made by CBE.

The proposed taxonomy addresses six important, mutually independent classifications that, overall, tell a fairly detailed story about a given MM building design. Because the scope of research spans both mixed-mode ventilation and mixed-mode HVAC, the categories have been chosen to allow more flexibility in this regard, while still underscoring a primary emphasis on ventilation

modes. The areas of focus are:

(1) **Site Context**: conditions at the building site, including climate and the siting of surrounding buildings or other blockages that may affect natural ventilation

(2) **Building Systems**: a key component to the classification that accounts for the use of any other unique building systems that may interact with and enhance the mixed-mode ventilation scheme

(3) **Mixed-Mode Topology**: the classic mixed-mode classifications of Leaman/Bordas/-CIBSE that define at the highest level the integration of the active and passive systems. Multiple topologies may apply to a given building.

(4) **Mixed-Mode Building Control**: the control states and overall control strategy employed by the building

(5) **Natural Ventilation Control**: the means by which outside air is introduced into the space and the level of automation present in those systems

(6) **Comfort Criteria**: the occupant comfort criteria employed in the design and operation of the building, whether adaptive in nature or more static/deterministic (e.g. PMV-PPD). This helps in understanding the basic intent of the building designers and the level of flexibility allowed in the thermal performance of the building systems.

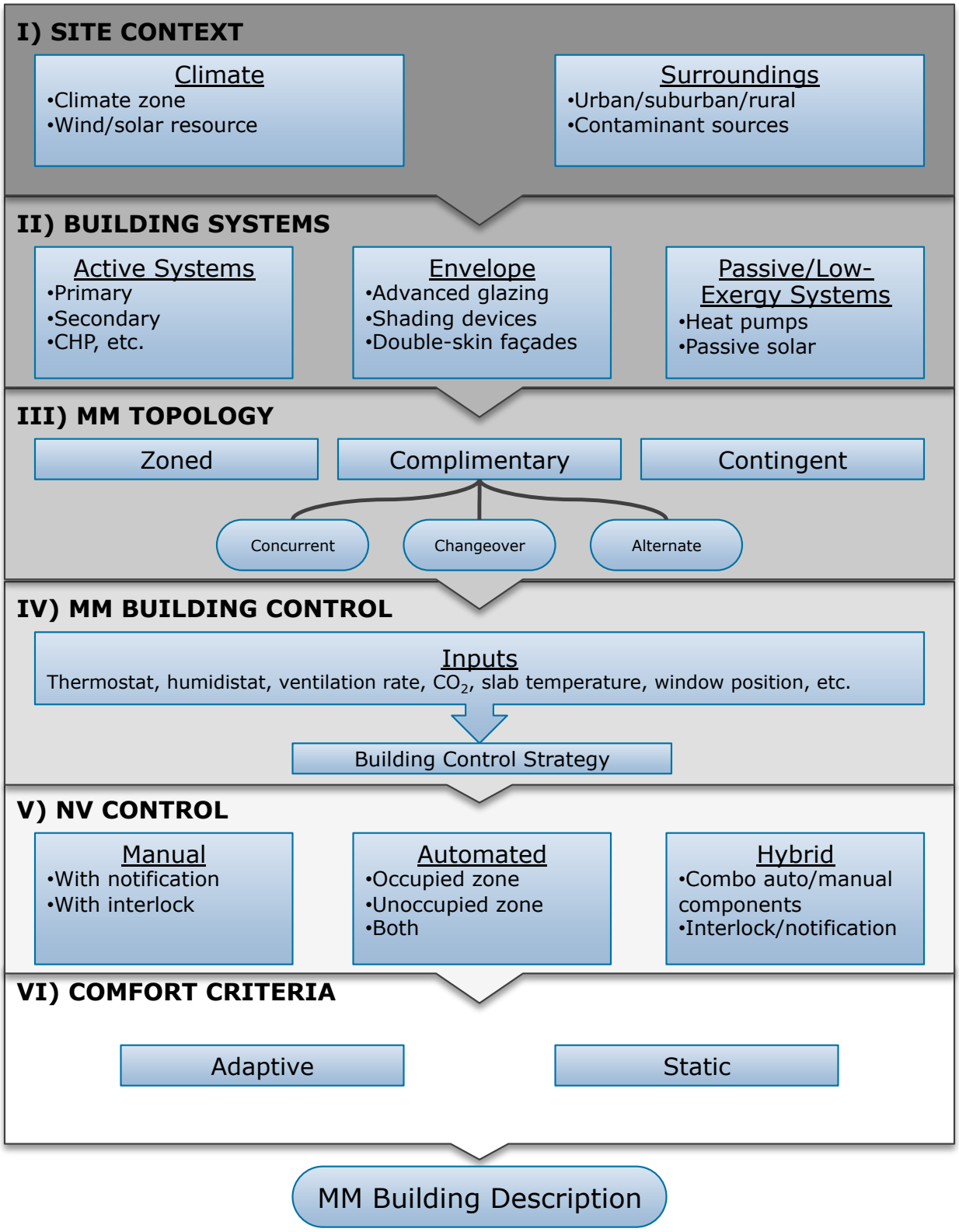


Figure B.2: The MM taxonomy employed categorizes buildings based on six criteria.

Appendix C

Key Matlab and R Codes

C.1 Parallel PSO Implementation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% FUNCTION
% PSO(opts)
%
% DESCRIPTION
% Particle Swarm Optimizer (PSO) based on the Common PSO Algorithm with
% support for evaluation of multiple neighborhoods on a Matlab cluster
% v_i(k+1) = chi*[phi*v_i(k)+alpha_1*(gamma_1i*(p_i-x_i(k)))+alpha_2*(gamma_2i*(G-x_1(
    k)))]
% x_i(k+1) = x_i(k)+v_1(k+1)
% i = particle index
% k = time index
% v_i = velocity of ith particle
% x_i = position of the ith particle
% p_i = best position found by the ith particle
% G = best position found by the swarm
% chi = constriction factor, see Clerc and Kennedy (2002)
% phi = intertial weighting, normally set to 1 for cases employing chi
% alpha_1 = acceleration constant
```

```
20 % alpha_2 = acceleration constant
21 % gamma_1i = random number [0,1] for ith particle
22 % gamma_2i = random number [0,1] for ith particle
23 %
24 % ARGUMENTS
25 % opts:          A structure containing the optimization parameters
26 %   .objfun:     A function handle to the objective function
27 %   .lower:      A column vector representing the lower bound of the decision
28 %                variables
29 %   .upper:      A column vector representing the upper bound of the decision
30 %                variables
31 %   .alpha_1:    Scalar
32 %   .alpha_2:    Scalar
33 %   .phi:        Scalar
34 %   .chi:        Scalar
35 %   .num_particles:   The swarm size
36 %   .max_iterations:   Maximum number of generations to simulate
37 %   .tolerance:   Convergence tolerance for population measured by RMSE
38 %   .time_stuck:   How many generations to go without improvement before
39 %                optimizer exits.
40 %
41 % RETURNS
42 % result:        A structure containing the result
43 %   .f1:         field
44 %   .f1:         field
45 %   .f1:         field
46 %   .f1:         field
47 %   .f1:         field
48 %
49 % AUTHOR
50 % Peter May—Ostendorp
51 % University of Colorado at Boulder
52 % mayosten@colorado.edu
```

```
53 %
54 % CREATED
55 % 19.Apr.2010
56 %
57 % REVISIONS
58 %
59 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
60
61 function result = PSO7distMaster(opts)
62 %%
63     %Time stuck parameter is optional; if not present, defaults to max
64     %iterations.
65     if ¬isfield(opts,'time_stuck')
66         opts.time_stuck = opts.max_iterations;
67     end
68
69     if opts.print
70
71         disp('PSO7dist Optimization Beginning.');
72         disp(' ');
73
74     end
75
76     result = struct;
77     diff = opts.upper—opts.lower;
78     dims = length(opts.lower);
79     particles = struct;
80     increment = opts.increment;
81     tic;    %Start timer
82     neighborhoods = opts.neighborhoods;
83     %termRegion = opts.termRegion;
84     opts.bins = (opts.lower == 1 | opts.lower == 0)...
85         & (opts.upper == 1 | opts.upper == 0) & increment == 1;   %flag position of
```

```
        binary decisions in the decision vector

%-------------------------------------------------------------------
%Create start x and v from Sobol sequence, per Laskari (2002)
%-------------------------------------------------------------------
q = qrandstream('sobol',dims);
xinit = rand(q,[opts.num_particles,dims])';
vinit = rand(q,[opts.num_particles,dims])';

%Initialize particles
for i=1:opts.num_particles

    particles(i).x = xinit(dims,i).*diff+opts.lower;
    particles(i).v = zeros(dims,1)+(2*vinit(dims,i)-1).*increment*opts.
        init_velocity_factor;  % scaled random start
    particles(i).p = particles(i).x;
    particles(i).f = 10^10;
    particles(i).gamma_1 = rand(dims,1);
    particles(i).gamma_2 = rand(dims,1);
    particles(i).alpha_1 = opts.alpha_1;
    particles(i).alpha_2 = opts.alpha_2;
    particles(i).neighborhood = mod(i-1,neighborhoods)+1;

end

%-------------------------------------------------------------------
%Initialize argument-passing structure for neighborhoods
%-------------------------------------------------------------------
for i=1:neighborhoods
   best.(['local',num2str(i)]) = particles(ceil(rand*opts.num_particles));
end

best.global = best.(['local',num2str(1)]);
```

```
117     worst = 0;
118
119     if isfield(opts,'seed')
120
121         ns = size(opts.seed);
122
123         for sc = 1:ns(2)
124
125             seed = best.global;
126             seed.x = opts.seed(:,sc);
127             seed.x(seed.x > opts.upper) = opts.upper(seed.x > opts.upper);
128             seed.x(seed.x < opts.lower) = opts.lower(seed.x < opts.lower);
129             seed.p = seed.x;
130             seed.f = 10^10;
131
132             if opts.print
133
134                 disp(['Running Seed Simulation']);
135                 str = '';
136
137                 %Tag termination horizon variables differently.
138                 for k=1:length(seed.x)
139                     str = [str,'x(',num2str(k),')=',num2str(seed.x(k)),' '];
140                 end
141
142                 disp(str);
143
144             end
145
146             f = opts.objfun(seed.x, opts);
147             seed.f = f;
148
149             if f < best.(['local',num2str(sc)]).f
```

```
150
151                 best.(['local',num2str(sc)]) = seed;
152
153             end
154
155             if f < best.global.f
156
157                 best.global = seed;
158
159             end
160
161             if f > worst
162
163                  worst = f;
164
165             end
166
167             if opts.print
168
169                 disp(['f(x)=',num2str(f)]);
170                 disp(['local min(f(x))=',num2str(f)]);
171                 disp(' ');
172
173             end
174
175         end
176
177     end
178
179     %-------------------------------
180     %Arguments for different labs
181     %-------------------------------
182     args = struct;
```

```
183     for i=1:neighborhoods
184         args(i).best = best;
185         args(i).particles = particles([particles.neighborhood] == i);
186         args(i).worst = worst;
187     end
188 
189     %-------------------------------------------
190     %Initialize some params for parallelization
191     %-------------------------------------------
192 
193     %Special parallel options and file handling
194     subOpts = opts;
195     if isfield(opts.par,'scratch')
196         scratch = opts.par.scratch;
197         scratchFiles = opts.par.files;
198     end
199 
200     for j=1:neighborhoods
201         subOpts(j) = opts;
202         scratchLocal = [scratch,'p',num2str(j),'/'];
203         if ~exist(scratchLocal,'dir')
204             mkdir(scratchLocal);
205         end
206         for k=1:length(scratchFiles)
207             copyfile(scratchFiles{k},scratchLocal);
208         end
209         subOpts(j).user.(opts.par.user{1}).(opts.par.user{2}) = scratchLocal;
210     end
211 
212 %% Find resources, build job, ensure sufficient nodes
213     if isfield(opts.par,'config')
214         sched = findResource('scheduler','configuration',opts.par.config);
215         sched.DataLocation = scratch;
```

```
216         job = createParallelJob(sched,'configuration',opts.par.config);
217     else
218         sched = findResource('scheduler','type','local');
219         sched.DataLocation = scratch;
220         job = createParallelJob(sched,'configuration','local');
221     end
222
223     %Check on neighborhood size
224     if neighborhoods > sched.ClusterSize
225         error('Number of neighborhoods larger than supported parallel configuration.')
                ;
226     end
227
228     set(job,'MinimumNumberOfWorkers',neighborhoods,'MaximumNumberOfWorkers',
            neighborhoods);
229     task = createTask(job,@PSO7distSlave,1,{args,subOpts},'CaptureCommandWindowOutput'
            ,true);
230
231 %% Job submission and gathering of results
232 try
233     submit(job);
234     disp(['Dispatched job to ',num2str(neighborhoods),' neighborhoods.']);
235     disp(' ');
236     wait(job,'finished');
237
238     %Gather outputs, get best result, display screen output.
239     outputs = getAllOutputArguments(job);
240     tasks = job.Tasks;
241     nout = neighborhoods;
242
243     %Check for presensce of errors
244     for i=1:nout
245         if ¬isempty(tasks(i).Error.identifier)
```

```
246             disp(['Errors encountered on neighborhood ',num2str(i)]);
247             throw(tasks(i).Error);
248         end
249     end
250
251     simulations = 0;
252     iterations = 0;
253     f = best.global.f;
254     for i=1:nout
255         simulations = simulations + outputs{i}.simulations;
256         iterations = iterations + outputs{i}.iterations;
257         if outputs{i}.best.global.f < f
258             best = outputs{i}.best;
259             f = best.global.f;
260         end
261         str = job.Tasks(i).CommandWindowOutput;
262         disp(str);
263     end
264
265     result.simulations = simulations;
266     result.iterations = iterations;
267     result.best = best.global;
268
269
270     if opts.print
271
272         disp(['PSO7dist finished in ',num2str(toc,'%0.2f'),' seconds, requiring ',
                num2str(simulations),' function evaluations.']);
273         disp(' ');
274
275     end
276
277 catch ERROR
```

```
278      displayError(ERROR);
279      disp(' ');
280      cancel(job);
281  end
282
283  end % function
```

```
%*************************************************************************
%
% function result = PSO7distSlave(args,subOpts)
%
% DESCRIPTION: slave function for distributed/parallel implementation of
% PSO7 that executes on an individual lab/neighborhood and communicates
% with other labs in the parallel task.
%
% ARGUMENTS:
% args —        structure containing copies of arguments for ALL LABS, from
%               which a copy of arguments will be localized
% subOpts —     structure containing copies of optimizer options for ALL
%               LABS, which will be copied locally
%
% RETURNS:
% result —      structure of the result
%
% AUTHOR:
% Peter May—Ostendorp
% University of Colorado Boulder
% mayosten@colorado.edu
%
% DATE:
% April 21, 2011
%
% SEE ALSO:
% PSO7distMaster.m, PSO7dn.m, PSO7par.m, PSO7sub.m
%
% REVISIONS
% 10.Jul.2011 (PMO) — slight tweaks for use with Thaddeus environment
%
%*************************************************************************

```

```
34 function result = PSO7distSlave(args,subOpts)
35
36 %------------------------------------------------------------------
37 %Localize and initialize values from opts, args
38 %------------------------------------------------------------------
39 neighborhood = labindex;
40 opts = subOpts(neighborhood);
41 args = args(neighborhood);
42 best = args.best;
43 particles = args.particles;
44 worst = args.worst;
45 tic;                          %Start local timer
46 rmse = 10^10;
47 dims = length(opts.lower);
48 increment = opts.increment;
49 diff = opts.upper-opts.lower;
50 const = opts.upper == opts.lower;
51 termRegion = opts.termRegion;
52 num_particles = length(particles);
53 error = ones(1,num_particles);          %100% error to start
54 simulations = 0;
55 taboo = [];
56 jobStatus = 1;                          %Lab active
57
58 %-----------------------------------
59 %Initialize counters for stall time
60 %-----------------------------------
61 bestgen = best.global.f;     %Init value for generational best
62 bestlastgen = bestgen;       %Init value for global min of last generation
63 tstuck  = 0;                 %Init value for time stuck
64
65 i = 1;
66
```

```
67 disp(['Neighborhood ',num2str(neighborhood),' starting.']);
68 disp(' ');
69
70 while i ≤ opts.max_iterations && rmse ≥ opts.tolerance && toc < opts.time_limit &&
       tstuck ≤ opts.time_stuck
71
72     j = 1;
73
74     while j ≤ num_particles && rmse ≥ opts.tolerance && toc < opts.time_limit
75
76         %Check if particle moving and zero out error if particle
77         %stationary.
78         if sqrt(sum((particles(j).v).^2)) ≤ opts.min_velocity
79             error(j) = 0; %Zero out error for stationary particles.
80
81             j = j + 1;
82             continue;
83         end
84
85         particles(j).gamma_1 = rand(dims,1);
86         particles(j).gamma_2 = rand(dims,1);
87         newx = particles(j).x;
88         newv = particles(j).v;
89
90         %--------------------------------------
91         %Include constriction factor, chi
92         %--------------------------------------
93         newv = opts.chi.*(newv+...
94             particles(j).alpha_1*particles(j).gamma_1.*(particles(j).p-newx)+...
95             particles(j).alpha_2*particles(j).gamma_2.*(best.(['local',num2str(
                   neighborhood)]).p-newx));
96         newx = newx+newv;
97
```

```
98          %-------------------------------------------------------------
99          %Enforce binary constraint per Kennedy and Eberhardt, 1997
100         %treating velocities as bit flipping probabilities
101         %-------------------------------------------------------------
102         Sv = 1./(1+exp(-newv));  %Sigmoidal limit transform of v
103         xbin = rand(dims,1) < Sv;     %
104         newx(opts.bins) = xbin(opts.bins);
105
106         %Enforce increments for other variables
107         newx = round(newx./increment(:)).*increment(:);
108
109         %For variables in fully constrained modes fix to bounds. Important to avoid
110         %constraint penalties.
111         newx(const) = opts.lower(const);
112
113         %-------------------------------------------------------------
114         % Vector post-processing for termination region filling.
115         %-------------------------------------------------------------
116         newx = postProcessControlVector(newx,opts);
117
118         %Enforce increments for other variables
119         newx = round(newx./increment(:)).*increment(:);
120
121         %-----------------------------------------
122         %Penalty used to enforce box constraints
123         %-----------------------------------------
124         %Determine feasible region and points outside
125         ubound = newx(~termRegion) > opts.upper(~termRegion);
126         lbound = newx(~termRegion) < opts.lower(~termRegion);
127
128         if sum(ubound) > 0 || sum(lbound) > 0
129
130             udist = sqrt(sum(((newx(ubound)-opts.upper(ubound))./diff(ubound)).^2));
```

```
131             ldist = sqrt(sum(((newx(lbound)—opts.lower(lbound))./diff(lbound)).^2));
132             d = udist + ldist;
133
134             %Update value (with penalty), velocity, position
135             %Penalty proportionate to distance outside the feasible
136             %region
137             f = 10*worst*(1 + d);
138             particles(j).v = newv;
139             particles(j).x = newx;
140
141             if f < particles(j).f
142                 particles(j).p = particles(j).x;
143                 particles(j).f = f;
144             end
145
146             %Error updating for out of bounds particles as well.
147             error(j) = (best.global.f — particles(j).f)/best.global.f;
148             if isnan(error(j))
149                 error(j) = 0;
150             end
151
152             j = j + 1;
153             continue;
154
155         end
156
157         %———————————————————————————————————————
158         %Taboo list, evaluated across all parallel neighborhoods
159         %———————————————————————————————————————
160         trials = size(taboo);
161         found = 0;
162
163         for trial=1:trials(2)
```

```
164
165             if (taboo(:,trial)==newx(:))
166
167                 found = 1;
168                 break;
169
170             end
171
172         end
173
174         if found == 1
175
176             j = j + 1;
177             continue;
178
179         else
180
181            taboo(:,trials(2)+1) = newx(:);
182
183         end
184
185         particles(j).v = newv;
186         particles(j).x = newx;
187
188         simulations = simulations + 1;
189
190         if opts.print
191
192             disp(['Generation=',num2str(i),' Neighborhood=',num2str(neighborhood),'
                    Particle=',num2str(j),' Simulation=',num2str(simulations)]);
193             str = '';
194
195             %Positions
```

```
196             for k=1:length(particles(j).x)
197
198                 str = [str,'x(',num2str(k),')=',num2str(particles(j).x(k)),' '];
199
200             end
201             disp(str);
202
203             str = '';
204
205             %Velocities
206 %                   for k=1:length(particles(j).x)
207 %
208 %                       str = [str,'v(',num2str(k),')=',num2str(particles(j).v(k),3)
    ,' '];
209 %
210 %                   end
211 %                   disp(str);
212
213         end
214
215         %------------------------------------------
216         %Objective function call happens here
217         %------------------------------------------
218         f = opts.objfun(particles(j).x, opts);
219
220         if f < particles(j).f
221
222             particles(j).p = particles(j).x;
223             particles(j).f = f;
224
225         end
226
227         if f < best.(['local',num2str(neighborhood)]).f
```

```
228
229             best.(['local',num2str(neighborhood)]) = particles(j);
230
231         end
232
233         if f < best.global.f
234
235             best.global = particles(j);
236
237         end
238
239         if f > worst
240
241              worst = f;
242
243         end
244
245         if opts.print
246
247             disp(['f(x)=',num2str(f)]);
248             disp(['local min(f(x))=',num2str(best.(['local',num2str(neighborhood)]).f)
                    ]);
249             disp(['global min(f(x))=',num2str(best.global.f)]);
250             disp(' ');
251
252         end
253
254         %Update error for particle
255         error(j) = (best.global.f - particles(j).f)/best.global.f;
256
257         %------------------------------------------------------------
258         %Labs wait for messaging interval to communicate
259         %------------------------------------------------------------
```

```
260         %RMSE implementation, based on objective function
261         %values rather than particle x values.
262         if mod(simulations,opts.par.messageInterval) == 0 && simulations > 0
263             disp('Neighborhoods talking...');
264             disp(' ');
265             err = gop(@horzcat,error);
266             rmse = sqrt(sum(err.^2));
267             globalBest = gop(@horzcat,best.global);
268             globalBest = globalBest([globalBest.f] == min([globalBest.f]));
269             best.global = globalBest(1);
270             globalTaboo = gop(@horzcat,taboo);
271             taboo = unique(globalTaboo','rows')';
272             labsActive = gplus(jobStatus);
273         end
274
275         j = j + 1;
276
277     end
278
279
280     bestgen = min([particles.f]);
281
282     %Check and increment time stuck
283     if bestgen ≥ bestlastgen   %No improvement
284         tstuck = tstuck + 1;
285     else                        %Improved value
286         tstuck = 0;
287     end
288     bestlastgen = bestgen;
289
290     if mod(i,50) == 0           %One more check to ensure check-in every 50
        generations
291             disp('Neighborhoods talking...');
```

```
292             disp(' ');
293             err = gop(@horzcat,error);
294             rmse = sqrt(sum(err.^2));
295             globalBest = gop(@horzcat,best.global);
296             globalBest = globalBest([globalBest.f] == min([globalBest.f]));
297             best.global = globalBest(1);
298             globalTaboo = gop(@horzcat,taboo);
299             taboo = unique(globalTaboo','rows')';
300             labsActive = gplus(jobStatus);
301     end
302
303     if opts.print
304
305         disp(['Neighborhood ',num2str(neighborhood),' at end of generation ',num2str(i
                ),':']);
306         disp(['Simulations=',num2str(simulations)]);
307         disp(['local min(f(x))=',num2str(best.(['local',num2str(neighborhood)]).f)]);
308         disp(['global min(f(x))=',num2str(best.global.f)]);
309         disp(['RMSE=',num2str(rmse)]);
310         disp(['t_s=',num2str(tstuck)]);
311         disp(' ');
312
313     end
314
315     if isfield(opts,'callback')
316
317        opts.callback(particles, opts, best.global, rmse);
318
319     end
320
321     i = i + 1;
322
323 end
```

```
324
325 %% While other labs are active, keep communicating
326 disp(['Neighborhood ',num2str(neighborhood),' complete.']);
327 disp(' ');
328 jobStatus = 0;                  %Job finished
329 labsActive = opts.neighborhoods;
330 while labsActive > 0
331     err = gop(@horzcat,error);
332     rmse = sqrt(sum(err.^2));
333     globalBest = gop(@horzcat,best.global);
334     globalBest = globalBest([globalBest.f] == min([globalBest.f]));
335     best.global = globalBest(1);
336     globalTaboo = gop(@horzcat,taboo);
337     taboo = unique(globalTaboo','rows')';
338     labsActive = gplus(jobStatus);
339 end
340
341 result.best         = best;
342 result.iterations   = i;
343 result.simulations  = simulations;
```

C.2 Mixed-Integer Response Surface

```
% mixIntTestSurface.m
%
% Pseudo-problem formulation of MINLP with global minimum at 280. Binary
% decisions are made on vector Wp and continuous decisions are made on Sp.
% Each vector is of length 12, as if day has been broken into 12 2-hour
% blocks. There is a high equivalancy in globally optimal solutions, but
% optimal solutions when windows are open during hours corresponding to
% 8am through 4pm and when global setpoints are raised during this period.
% This is a typical changeover behavior.

function out = mixIntTestSurface(vector,prob,varargin)

cost = 300;
Sp = vector(1:12);
Wp = vector(13:24);
Wp(Wp ≤ 0) = 0;
Wp(Wp > 0) = 1;

%Assign cost for hours midnight through 8am
if sum(Wp(1:4)) > 0

    cost = cost + 50;

end

%Assign cost for hours 8am through 4pm
if sum(Wp(5:8)) > 0 || sum(Sp(5:8) < 26) > 0

    cost = cost - 20*sum(Wp(5:8))/4 + 40*sum(Sp(5:8) < 26)/4;
```

```
30
31 end
32
33 %Assign cost for hours 4pm through midnight
34 if sum(Wp(9:12)) > 0 || sum(Sp(9:12) < 26) > 0
35
36     cost = cost + 10*sum(Wp(9:12))/4 + 40*sum(Sp(9:12) < 26)/4;
37
38 end
39
40 %Final outputs
41
42 if nargin > 2
43
44     out.f = cost;
45
46 else
47
48     out = cost;
49
50 end
```

C.3 R Functions for Translation of Rules into ERL Code

```
# glmToEP.R
# Source file contains glmToEP.R and helper function makeGLMString

glmToEP = function(glmobj,fileName){

    #Open file for writing.
    fileID = file(description=fileName)
    open(fileID, open="w")

    #Process the glm object
    glmString = makeGLMString(glmobj)

    #Write it all out and close file connection.
    writeLines("!-------- Program stub created by glmToEP --------!",con=fileID)
    writeLines("!-------- EnergyManagementSystem:Sensor --------!",con=fileID)
    writeLines(glmString$sensors,con=fileID)    #sensors
    writeLines("!-------- EnergyManagementSystem:Actuator --------!",con=fileID)
    writeLines(glmString$actuator,con=fileID)   #actuator
    writeLines("!-------- EnergyManagementSystem:Program --------!",con=fileID)
    writeLines(glmString$program,con=fileID)    #program
    writeLines(glmString$initconstants,con=fileID)
    writeLines("!-------- EnergyManagementSystem:GlobalVariable --------!",con=fileID)
    writeLines(glmString$global,con=fileID)
    close(fileID)
}

makeGLMString = function(glmobj){

    #Localize object
```

```
30      coef = glmobj$coefficients
31      n = length(coef)
32      coef0 = coef[1]
33      coef = coef[2:n]
34      sensors = names(coef)
35
36      #Make the program
37      str = "EnergyManagementSystem:Program,\n"
38      str = paste(str,"\t<PROGRAM NAME>,\n",sep="")
39      str = paste(str,"\tSET theta = beta0,\n",sep="")
40      for(i in 1:(n-1)){
41          str = paste(str,"\tSET theta = theta + beta",i,"*",sensors[i],",\n",sep="")
42      }
43      str = paste(str,"\tSET theta = 1/theta,\n",sep="")
44      str = paste(str,"\tSET signal = theta;\n",sep="")
45
46      makeGLMString = list()
47      makeGLMString$program = str
48
49      #Create a stub to initiatlize constants
50      str = "EnergyManagementSystem:Program,\n"
51      str = paste(str,"\t<PROGRAM NAME>,\n",sep="")
52      coefstr = coef0
53      if(coef0<0){coefstr = paste("0 - ",abs(coef0),sep="")}
54      str = paste(str,"\tSET beta0 = ",coefstr,",\n",sep="")
55      for(i in 1:(n-1)){
56          coefstr = coef[i]
57          if(coef[i]<0){coefstr = paste("0 - ",abs(coef[i]),sep="")}
58          if(i==(n-1)){
59              str = paste(str,"\tSET beta",i," = ",coefstr,";\n",sep="")
60          }else{
61              str = paste(str,"\tSET beta",i," = ",coefstr,",\n",sep="")
62          }
```

```
63      }
64
65      makeGLMString$initconstants = str
66
67      #Declare global variables, i.e. constants
68      global = "EnergyManagementSystem:GlobalVariable,\n"
69      global = paste(global,"\tbeta0,\n",sep="")
70      for(i in 1:(n-1)){
71          if(i==(n-1)){
72              global = paste(global,"\tbeta",i,";\n",sep="")
73          }else{
74              global = paste(global,"\tbeta",i,",\n",sep="")
75          }
76      }
77
78      makeGLMString$global = global
79
80      #Create a list of unique predictor variables used and generate EP sensor stubs
81      ns = length(sensors)
82      sensorString = vector(length=ns)
83      for(i in 1:ns){
84          str = "EnergyManagementSystem:Sensor,\n"
85          str = paste(str,"\t",sensors[i],",\n",sep="")
86          str = paste(str,"\t<INSERT KEY NAME>,\n",sep="")
87          str = paste(str,"\t<INSERT VAR NAME>;\n",sep="")
88          sensorString[i] = str
89      }
90
91      makeGLMString$sensors = sensorString
92
93
94      #Create a stub for the actuator
95      actuator = "EnergyManagementSystem:Actuator,\n"
```

```
96      actuator = paste(actuator,"\tsignal,\n",sep="")
97      actuator = paste(actuator,"\t<ACTUATED COMPONENT UNIQUE NAME>,\n",sep="")
98      actuator = paste(actuator,"\t<ACTUATED COMPONENT TYPE>,\n",sep="")
99      actuator = paste(actuator,"\t<CONTROL TYPE>;\n",sep="")
100
101     makeGLMString$actuator = actuator
102     return(makeGLMString)
103
104 }
```

```
#
#   treeToEP source file contains
#   adaToEP:    function for tuning ada package boost objects
#               into EnergyPlus EMS code
#   rpartToEP:  similar function for handling rpart classification
#               and regression trees
#
#   USE REQUIRES THAT YOU SOURCE HELPER FUNCTIONS MAKEBOOSTSTRING
#   AND MAKETREESTRING
#
#*****************************************************************

# source("makeBoostString.r")
# source("makeTreeString.r")

adaToEP = function(boostobj,fileName){

    #Open file for writing.
    fileID = file(description=fileName)
    open(fileID, open="w")

    #Process the boost object.
    boostString = makeBoostString(boostobj)

    #Write it all out and close file connection.
    writeLines("!-------- Program stub created by adaToEP --------!",con=fileID)
    writeLines("!-------- EnergyManagementSystem:Sensor --------!",con=fileID)
    writeLines(boostString$sensors,con=fileID)  #sensors
    writeLines("!-------- EnergyManagementSystem:Actuator --------!",con=fileID)
    writeLines(boostString$actuator,con=fileID) #actuator
    writeLines("!-------- EnergyManagementSystem:Program --------!",con=fileID)
    writeLines(boostString$program,con=fileID)  #program
    close(fileID)
```

```
34 }
35
36 #Secondary function for handling pure CARTs
37 rpartToEP = function(treeobj,fileName){
38
39     #Open file for writing.
40     fileID = file(description=fileName)
41     open(fileID, open="w")
42
43     #Process the boost object.
44     treeString = makeTreeString(treeobj)
45
46     #Write it all out and close file connection.
47     writeLines("!———— Program stub created by adaToEP ————!",con=fileID)
48     writeLines("!———— EnergyManagementSystem:Sensor ————!",con=fileID)
49     writeLines(treeString$sensors,con=fileID)   #sensors
50     writeLines("!———— EnergyManagementSystem:Actuator ————!",con=fileID)
51     writeLines(treeString$actuator,con=fileID)  #actuator
52     writeLines("!———— EnergyManagementSystem:Program ————!",con=fileID)
53     writeLines(treeString$program,con=fileID)   #program
54     close(fileID)
55 }
```

```
# makeTreeString = function(boostmodel)
#
#
#*****************************************************************

#Base version of the function for use with rpart objects
makeTreeString = function(treeobj){

    #Localize some stuff from the tree object
    frame = treeobj$frame
    splits = treeobj$splits
    splits = splits[splits[,1]!=0,]
    nodeNames = as.vector(frame[,1])
    terminal = nodeNames == "<leaf>"
    nodeVals = frame[,5]
    if(treeobj$method=="class"){
        nodeVals[nodeVals==1] = -1
        nodeVals[nodeVals==2] = 1
    }
    nodeNums = attr(frame,"row.names")

    nnodes = dim(frame)[1]
    nsur = treeobj$control$maxsurrogate
    ii = dim(splits)[1]/nsur
    tmp = vector()  #For equiv signs
    tmp2 = vector() #For breaks

    #Find where the final splits are
    for(i in 1:ii){
        tmp[i]=splits[(nsur*(i-1)+1),2]
        tmp2[i]=splits[(nsur*(i-1)+1),4]
    }

```

```
34      #Coerce splitEquality into right symbols and pad it
35      splitEquality = vector(length=nnodes)
36      splitEquality[terminal] = ""
37      splitEquality[!terminal] = tmp
38      splitEquality[splitEquality==1] = "≥"
39      splitEquality[splitEquality==-1] = "<"
40
41      #Pad split breaks to be the same length as the frame
42      splitBreaks = vector(length=nnodes)
43      splitBreaks[terminal] = -99999
44      splitBreaks[!terminal] = tmp2
45
46      #Generate EP ERL output for the tree, gotta do this recursively
47      str = "EnergyManagementSystem:Program,\n"
48      str = paste(str,"<PROGRAM NAME>,\n",sep="")
49
50      #Recursively process the nodes with makeNodeString
51      str = makeNodeString(1,nodeNames,splitEquality,splitBreaks,nodeNums,str,terminal,
            nodeVals)
52      str = paste(str,"SET action = vote;\n",sep="")
53
54      makeTreeString = list()
55      makeTreeString$program = str
56
57      #Create a list of unique predictor variables used and generate EP sensor stubs
58      sensors = unique(nodeNames[nodeNames!="<leaf>"])
59      ns = length(sensors)
60      sensorString = vector(length=ns)
61      for(i in 1:ns){
62          str = "EnergyManagementSystem:Sensor,\n"
63          str = paste(str,"\t",sensors[i],",\n",sep="")
64          str = paste(str,"\t<INSERT KEY NAME>,\n",sep="")
65          str = paste(str,"\t<INSERT VAR NAME>;\n",sep="")
```

```
66          sensorString[i] = str
67      }
68
69      makeTreeString$sensors = sensorString
70
71      #Create a stub for the actuator
72      actuator = "EnergyManagementSystem:Actuator,\n"
73      actuator = paste(actuator,"\taction,\n",sep="")
74      actuator = paste(actuator,"\t<ACTUATED COMPONENT UNIQUE NAME>,\n",sep="")
75      actuator = paste(actuator,"\t<ACTUATED COMPONENT TYPE>,\n",sep="")
76      actuator = paste(actuator,"\t<CONTROL TYPE>;\n",sep="")
77
78      makeTreeString$actuator = actuator
79      return(makeTreeString)
80
81  }
82  #end function
83
84  #A version of the function specifically designed for use with processBoost.r
85  #which works on ada objects. Doesn't return ERL program headers.
86  makeBoostTreeString = function(treeobj){
87
88      #Localize some stuff from the tree object
89      frame = treeobj$frame
90      splits = treeobj$splits
91      splits = splits[splits[,1]!=0,]
92      nodeNames = as.vector(frame[,1])
93      terminal = nodeNames == "<leaf>"
94      nodeVals = frame[,5]
95      if(treeobj$method=="class"){
96          nodeVals[nodeVals==1] = -1
97          nodeVals[nodeVals==2] = 1
98      }
```

```
99      nodeNums = attr(frame,"row.names")
100
101     nnodes = dim(frame)[1]
102     nsur = treeobj$control$maxsurrogate
103     ii = dim(splits)[1]/nsur
104     tmp = vector()  #For equiv signs
105     tmp2 = vector() #For breaks
106
107     #Find where the final splits are
108     for(i in 1:ii){
109         tmp[i]=splits[(nsur*(i-1)+1),2]
110         tmp2[i]=splits[(nsur*(i-1)+1),4]
111     }
112
113     #Coerce splitEquality into right symbols and pad it
114     splitEquality = vector(length=nnodes)
115     splitEquality[terminal] = ""
116     splitEquality[!terminal] = tmp
117     splitEquality[splitEquality==1] = "≥"
118     splitEquality[splitEquality==-1] = "<"
119
120     #Pad split breaks to be the same length as the frame
121     splitBreaks = vector(length=nnodes)
122     splitBreaks[terminal] = -99999
123     splitBreaks[!terminal] = tmp2
124
125     #Generate EP ERL output for the tree, gotta do this recursively
126     str = vector()
127
128     #Recursively process the nodes with makeNodeString
129     str = makeNodeString(1,nodeNames,splitEquality,splitBreaks,nodeNums,str,terminal,
            nodeVals)
130
```

```
131     makeBoostTreeString = list()
132     makeBoostTreeString$program = str
133
134     #Create a list of unique predictor variables used and generate EP sensor stubs
135     sensors = unique(nodeNames[nodeNames!="<leaf>"])
136     ns = length(sensors)
137     sensorString = vector(length=ns)
138     for(i in 1:ns){
139         str = "EnergyManagementSystem:Sensor,\n"
140         str = paste(str,"\t",sensors[i],",\n",sep="")
141         str = paste(str,"\t<INSERT KEY NAME>,\n",sep="")
142         str = paste(str,"\t<INSERT VAR NAME>;\n",sep="")
143         sensorString[i] = str
144     }
145
146     makeBoostTreeString$sensors = sensorString
147
148     #Create a stub for the actuator
149     actuator = "EnergyManagementSystem:Actuator,\n"
150     actuator = paste(actuator,"\tvote,\n",sep="")
151     actuator = paste(actuator,"\t<ACTUATED COMPONENT UNIQUE NAME>,\n",sep="")
152     actuator = paste(actuator,"\t<ACTUATED COMPONENT TYPE>,\n",sep="")
153     actuator = paste(actuator,"\t<CONTROL TYPE>;\n",sep="")
154
155     makeBoostTreeString$actuator = actuator
156     return(makeBoostTreeString)
157
158 }
159
160 #Recursive function for node processing
161 makeNodeString = function(n,nodeNames,splitEquality,splitBreaks,nodeNums,str,terminal,
        nodeVals){
162     #Define the split for this node
```

```
163     str = paste(str,"IF ",nodeNames[n]," ",splitEquality[n]," ",round(splitBreaks[n],
            digits=4),",\n",sep="")
164
165     n = nodeNums[n]
166
167     #Check for a left connection
168     left = 2*n
169     if(any(nodeNums == left)){
170
171         idx = which(nodeNums==left)
172         #Is it a terminal node?
173         if(terminal[idx]){
174             wd = nodeVals[idx]
175             if(nodeVals[idx]<0){wd=paste("0 - ",abs(nodeVals[idx]),sep="")}
176             str = paste(str,"SET vote = ",wd,",\n",sep="")
177         }else{
178             str = makeNodeString(idx,nodeNames,splitEquality,splitBreaks,nodeNums,str,
                    terminal,nodeVals)
179         }
180     }
181
182     #Create a path to the right side
183     str = paste(str,"ELSE,\n",sep="")
184
185     #Check for a right connection
186     right = 2*n + 1
187     if(any(nodeNums == right)){
188
189         idx = which(nodeNums==right)
190         #Is it a terminal node?
191         if(terminal[idx]){
192             str = paste(str,"SET vote = ",nodeVals[idx],",\n",sep="")
193         }else{
```

```
194             str = makeNodeString(idx,nodeNames,splitEquality,splitBreaks,nodeNums,str,
                    terminal,nodeVals)
195         }
196     }
197
198     #Close up the IF/THEN statement for this node.
199     str = paste(str,"ENDIF,\n",sep="")
200
201     makeNodeString = str
202     return(makeNodeString)
203 }
```

```
# makeBoostString = function(boostobj)
#
#
#*********************************************************

makeBoostString = function(boostobj){

    #Localize
    trees = boostobj$model$trees
    alphas = boostobj$model$alpha

    n = length(alphas)

    #Chain all learner code together
    sensors = vector();
    str = "EnergyManagementSystem:Program,\n"
    str = paste(str,"<PROGRAM NAME>,\n",sep="")
    str = paste(str,"SET sum = 0,\n",sep="")
    for(i in 1:n){

        tmp = makeBoostTreeString(trees[[i]])
        str = paste(str,tmp$program,sep="")

        #Add weighting to the vote term and update sum
        #Assumption that alpha will ALWAYS be a positive value,
        #otherwise have to handle for rtarded way EP deals with
        #setting negative values, i.e. 0 - number.
        wd = round(alphas[i],digits=4)
        str = paste(str,"SET alpha",i," = ",wd,",\n",sep="")
        str = paste(str,"SET vote = vote*alpha",i,",\n",sep="")
        str = paste(str,"SET sum = sum + vote,\n",sep="")

        sensors = c(sensors,tmp$sensors)
```

```
34      }
35
36      #Determine the sign of sum for control action
37      str = paste(str,"IF sum > 0,\n",sep="")
38      str = paste(str,"SET h = 1,\n",sep="")
39      str = paste(str,"ELSE,\n",sep="")
40      str = paste(str,"SET h = 0,\n",sep="")
41      str = paste(str,"ENDIF;\n",sep="")
42
43      makeBoostString = list()
44      makeBoostString$program = str
45
46      #Establish unique sensors
47      sensors = unique(sensors)
48      makeBoostString$sensors = sensors
49
50      #Create an actuator stub for the control action
51      actuator = "EnergyManagementSystem:Actuator,\n"
52      actuator = paste(actuator,"\th,\n",sep="")
53      actuator = paste(actuator,"\t<ACTUATED COMPONENT UNIQUE NAME>,\n",sep="")
54      actuator = paste(actuator,"\t<ACTUATED COMPONENT TYPE>,\n",sep="")
55      actuator = paste(actuator,"\t<CONTROL TYPE>;\n",sep="")
56
57      makeBoostString$actuator = actuator
58      return(makeBoostString)
59
60  }
```

Appendix D

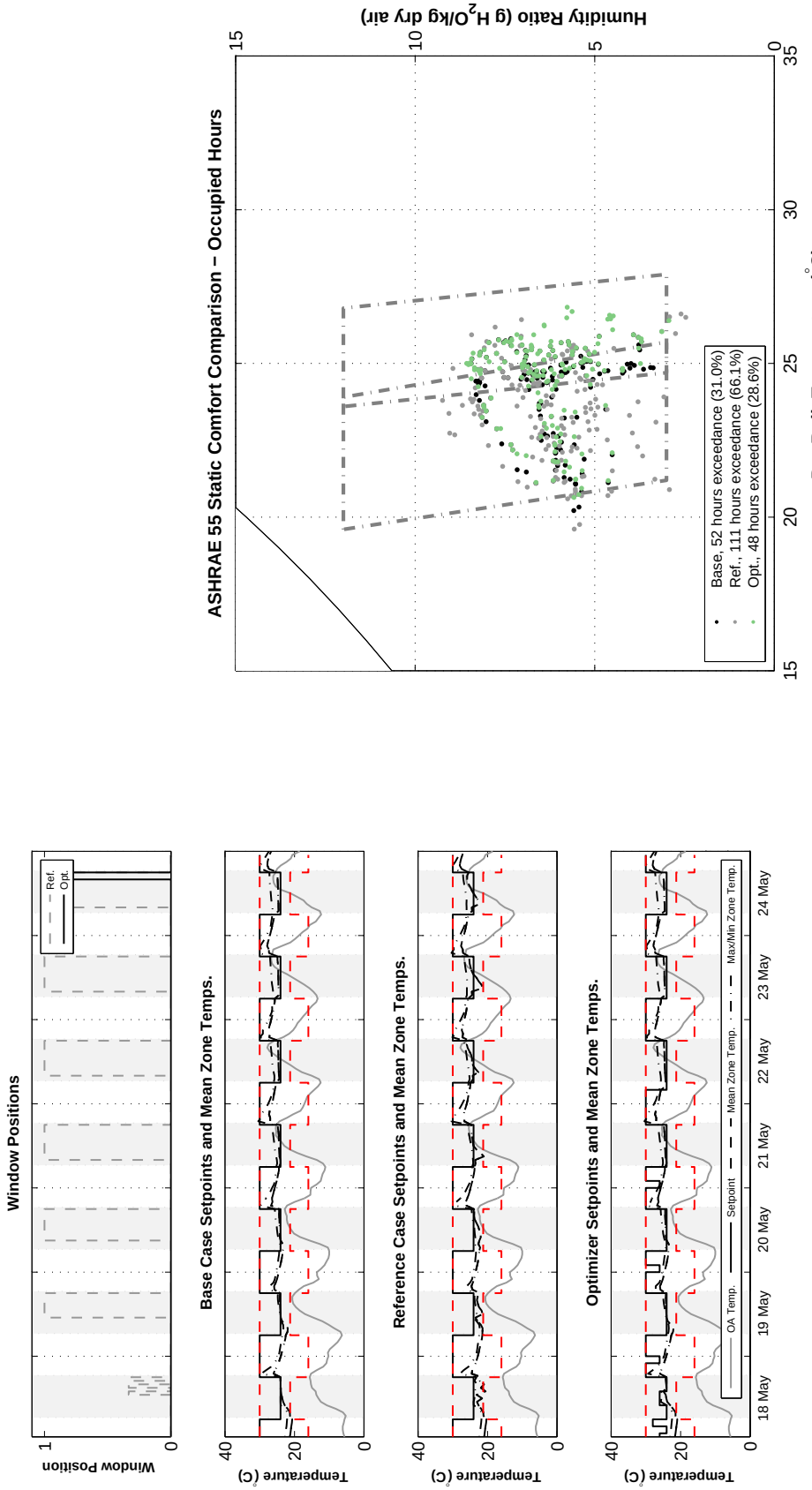
Supplementary Offline MPC Study Results

Summary result tables and plots from the ASHRAE 55 static (swing season) and 55 adaptive (cooling season) penalized solutions are also provided. These combinations consistently represented the best energy-comfort combinations for a given season (MM4 is the only exception; 55 adaptive solutions are shown for both seasons). Care should be taken in analyzing energy savings, as statically penalized solutions are tied to the base case in terms of comfort, whereas adaptively penalized solutions are tied to the reference case. A solution may achieve minor savings over one case and not the other because of comfort constraints.

D.1 MM1 Partial Changeover Results

Table D.1: MM1 Energy and Comfort Solution Summary

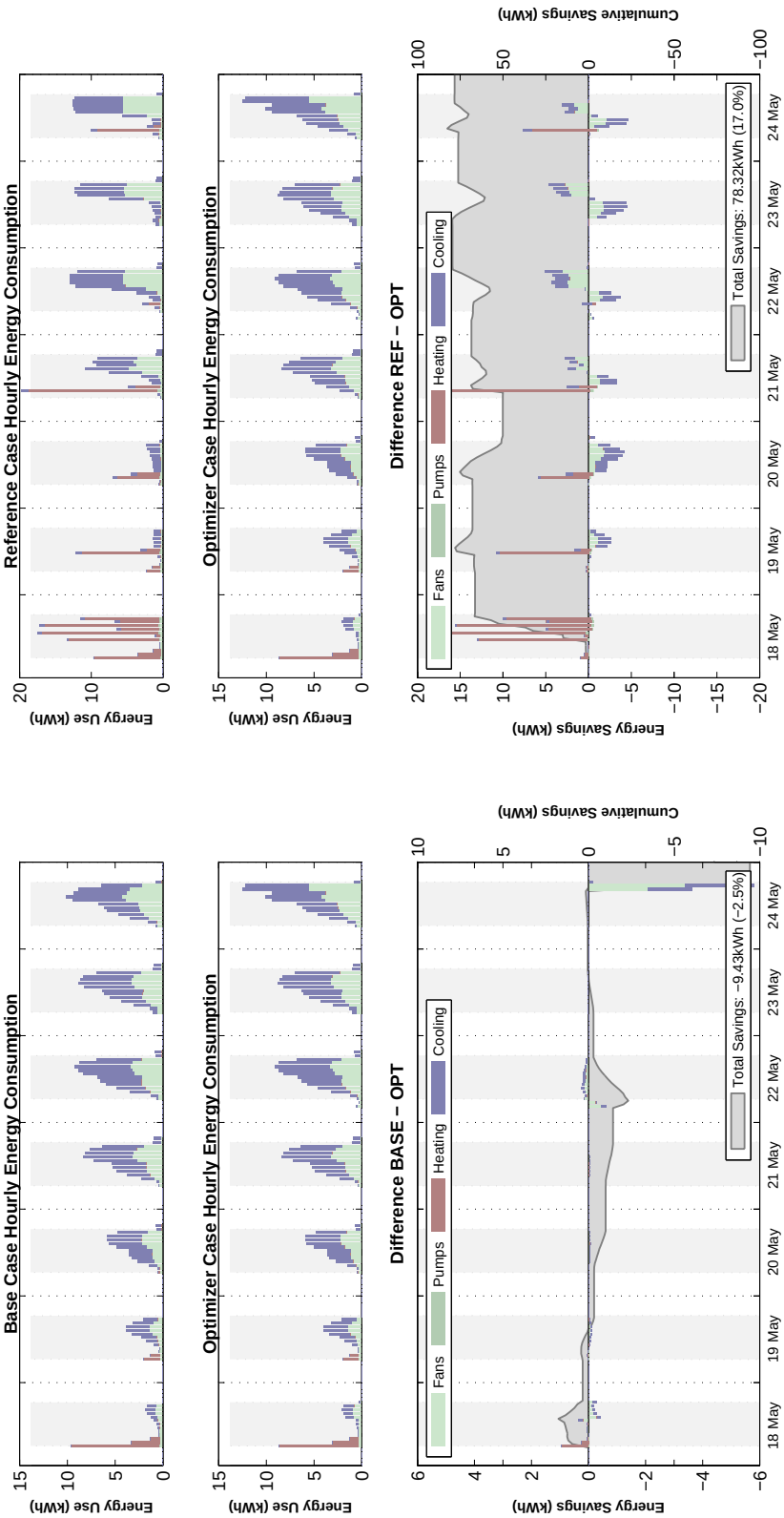
	Swing Season			Cooling Season		
	HVAC Electric	HVAC Gas Use (kWh)	Comfort Violations	HVAC Electric	HVAC Gas Use (kWh)	Comfort Violations
Base Case	356	16	11	601	0	0
Reference Case	319	138	9	648	0	0
Optimal Case						
55 static	367	15	9	560	0	0
55 adaptive	296	98	3	485	0	0
15251 adaptive	331	60	0	407	0	0



(a) Solution

(b) Comfort

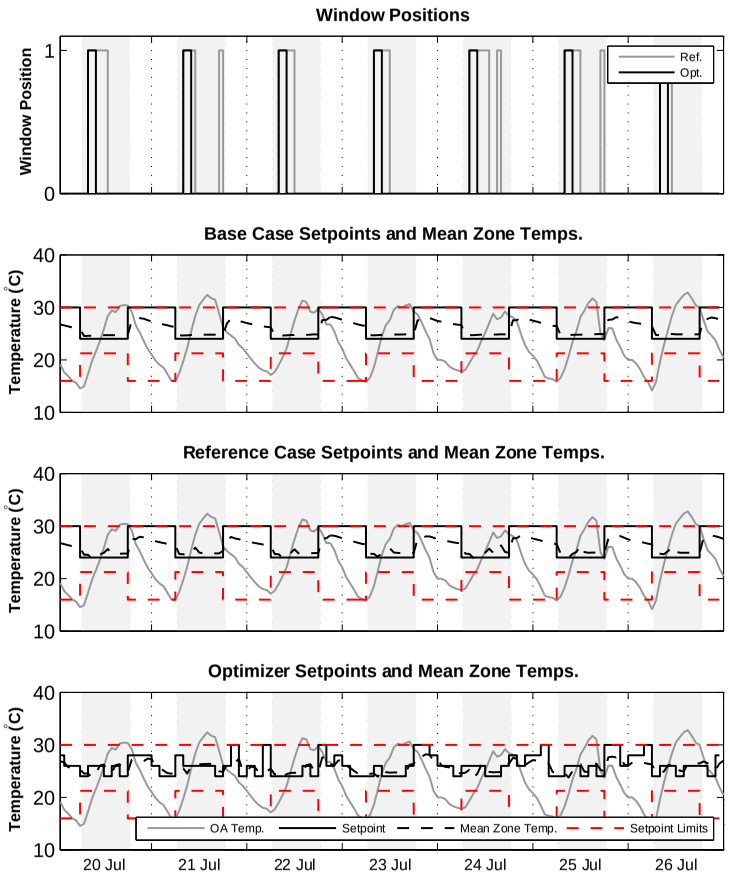
Figure D.1: Solution and comfort: MM1, swing season, ASHRAE 55 static comfort penalty



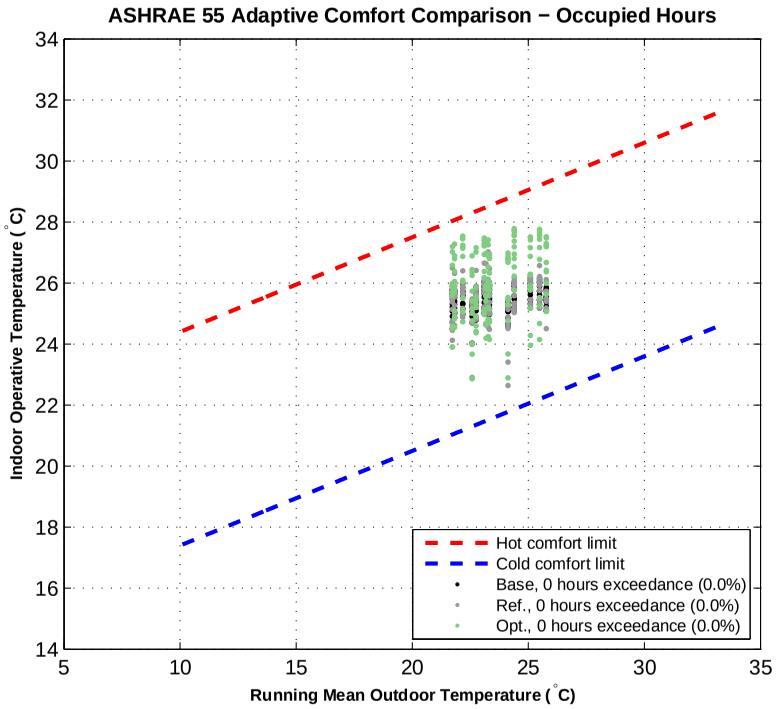
(a) Base vs. optimal

(b) Reference vs. optimal

Figure D.2: Savings: MM1, swing season, ASHRAE 55 static comfort penalty

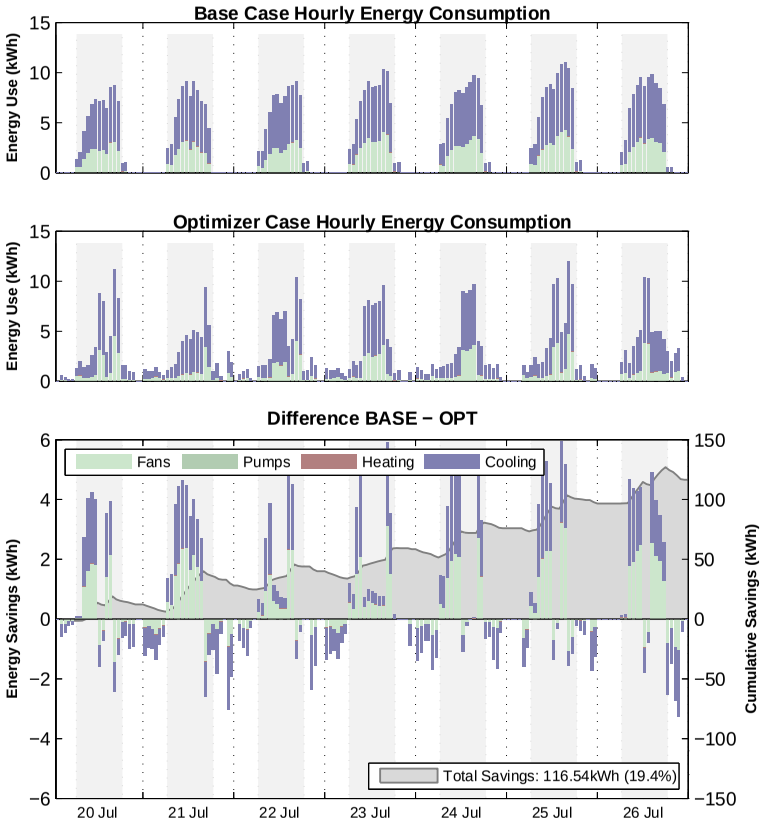


(a) Solution

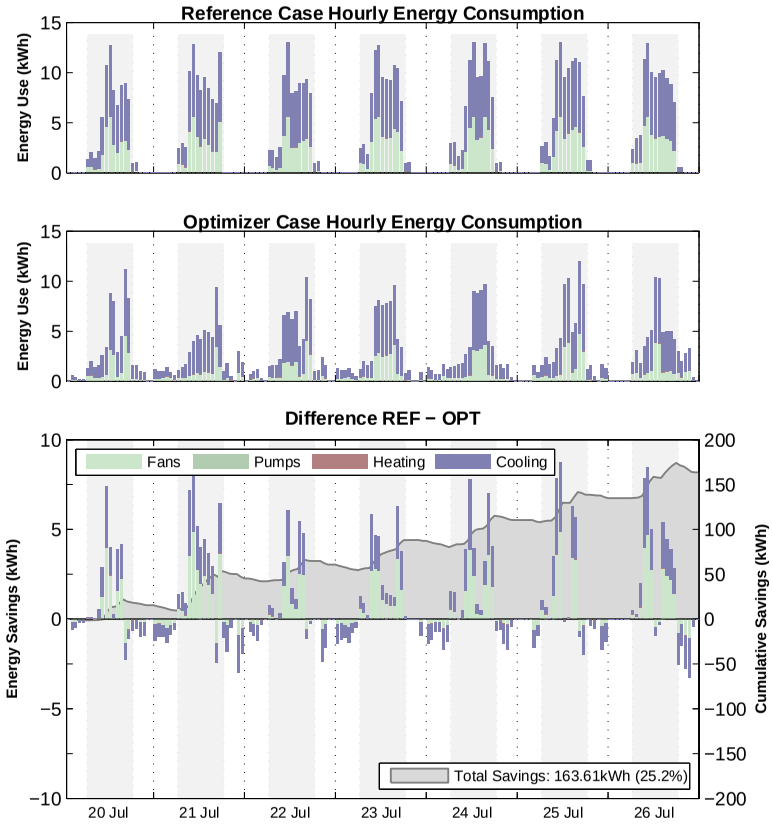


(b) Comfort

Figure D.3: Solution and comfort: MM1, cooling season, ASHRAE 55 adaptive comfort penalty



(a) Base vs. optimal



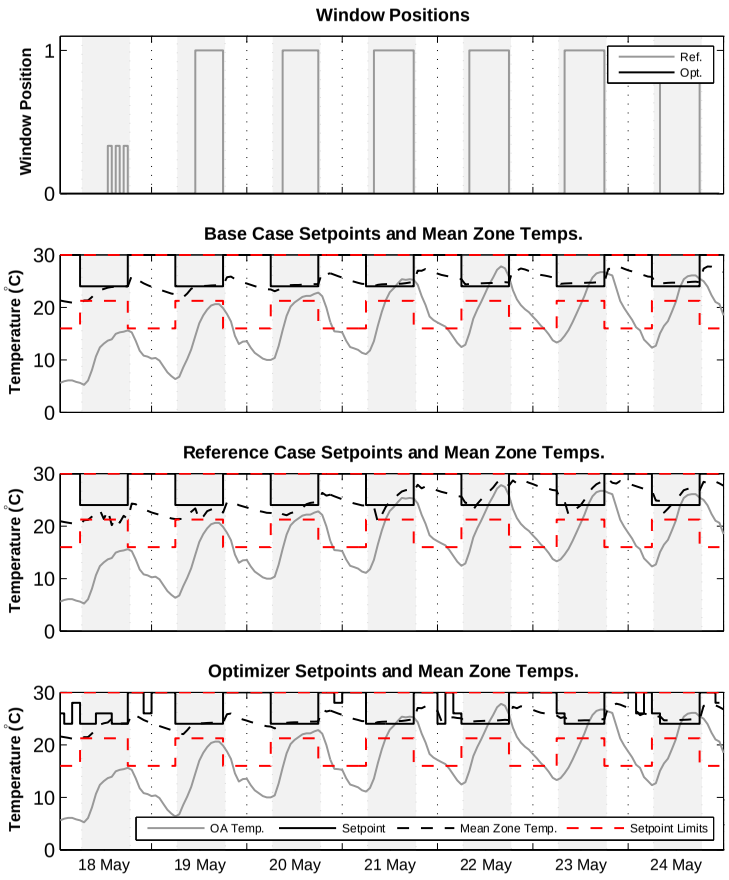
(b) Reference vs. optimal

Figure D.4: Savings: MM1, cooling season, ASHRAE 55 adaptive comfort penalty

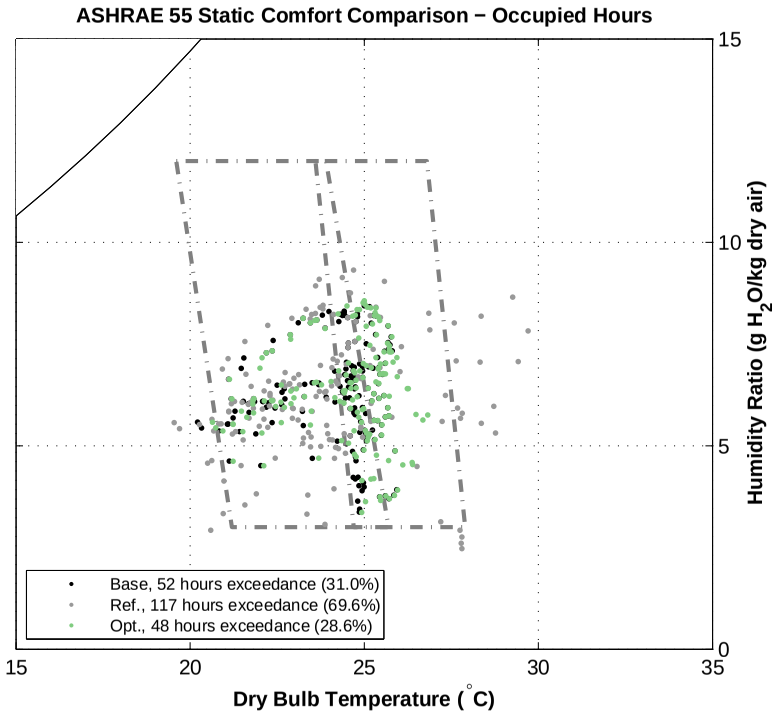
D.2 MM1 Changeover Results

Table D.2: MM1 Energy and Comfort Solution Summary

	Swing Season			Cooling Season		
	HVAC Electric	HVAC Gas Use (kWh)	Comfort Violations	HVAC Electric	HVAC Gas Use (kWh)	Comfort Violations
Base Case	356	16	11	601	0	0
Reference Case	32	30	28	475	0	9
Optimal Case						
55 static	360	15	9	576	0	0
55 adaptive	54	27	27	424	0	1
15251 adaptive	118	26	0	422	0	0

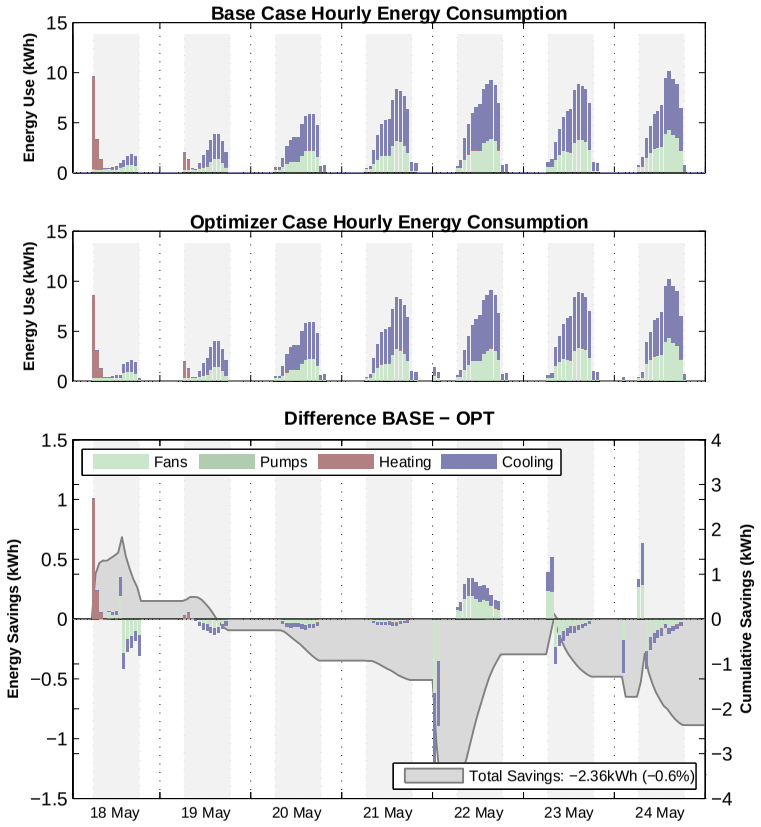


(a) Solution

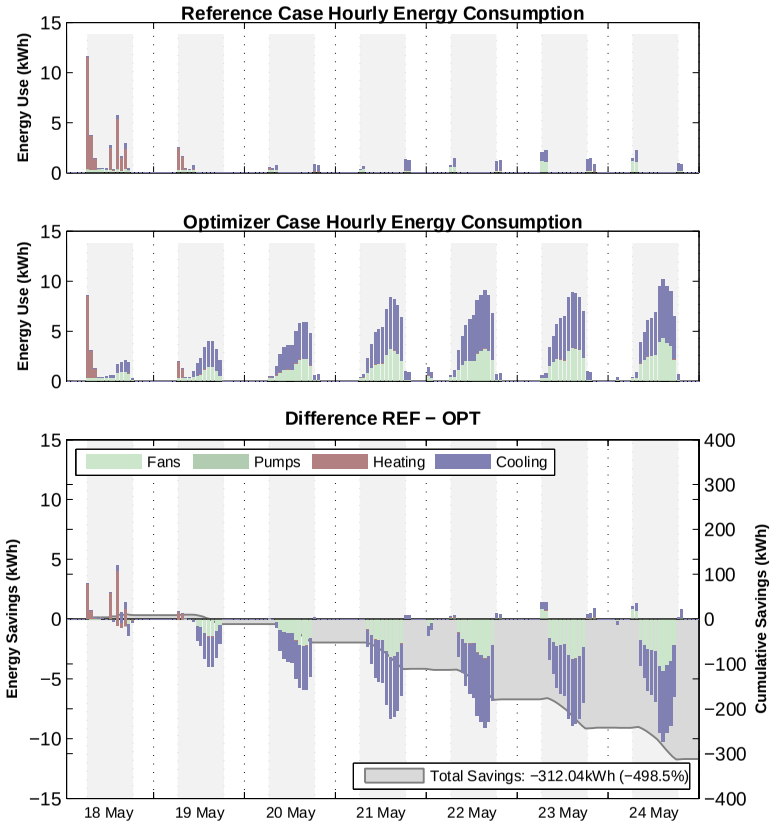


(b) Comfort

Figure D.5: Solution and comfort: MM1, swing season, ASHRAE 55 static comfort penalty

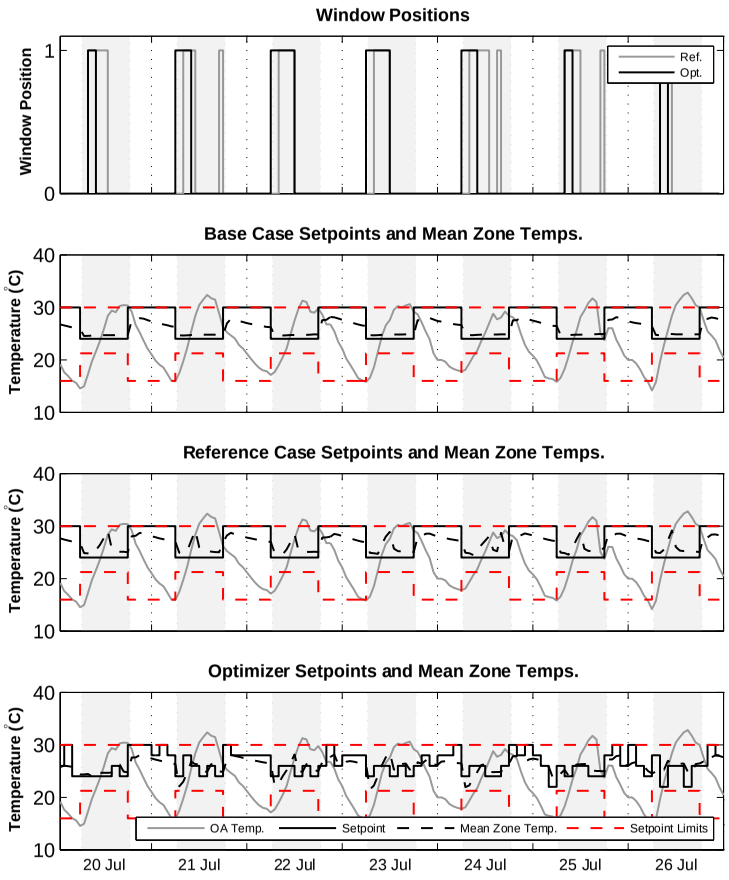


(a) Base vs. optimal

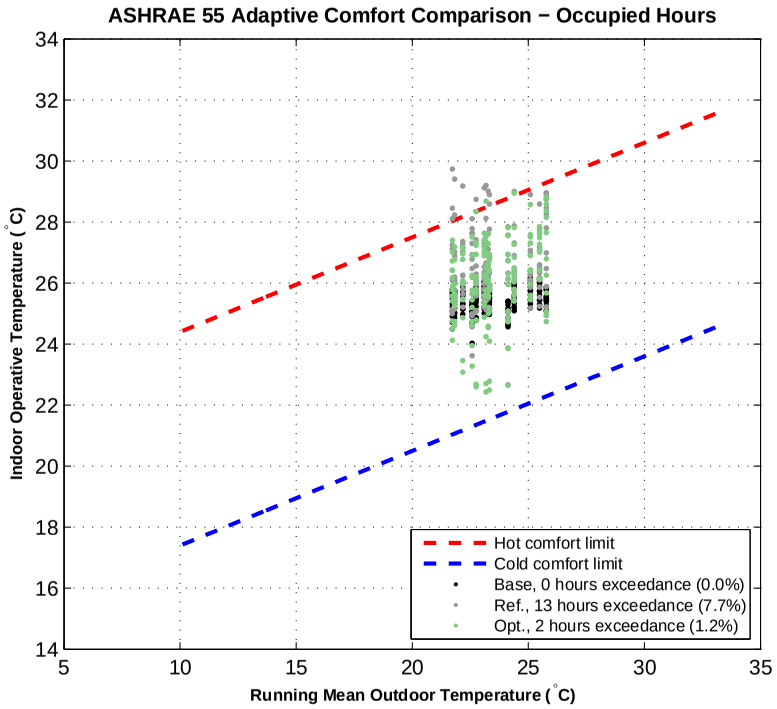


(b) Reference vs. optimal

Figure D.6: Savings: MM1, swing season, ASHRAE 55 static comfort penalty

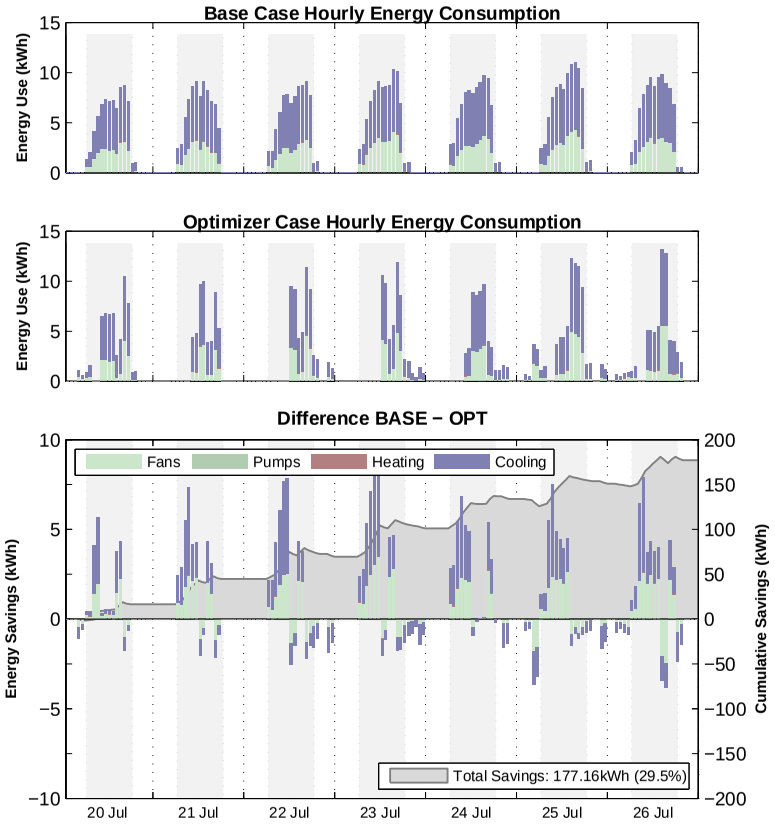


(a) Solution

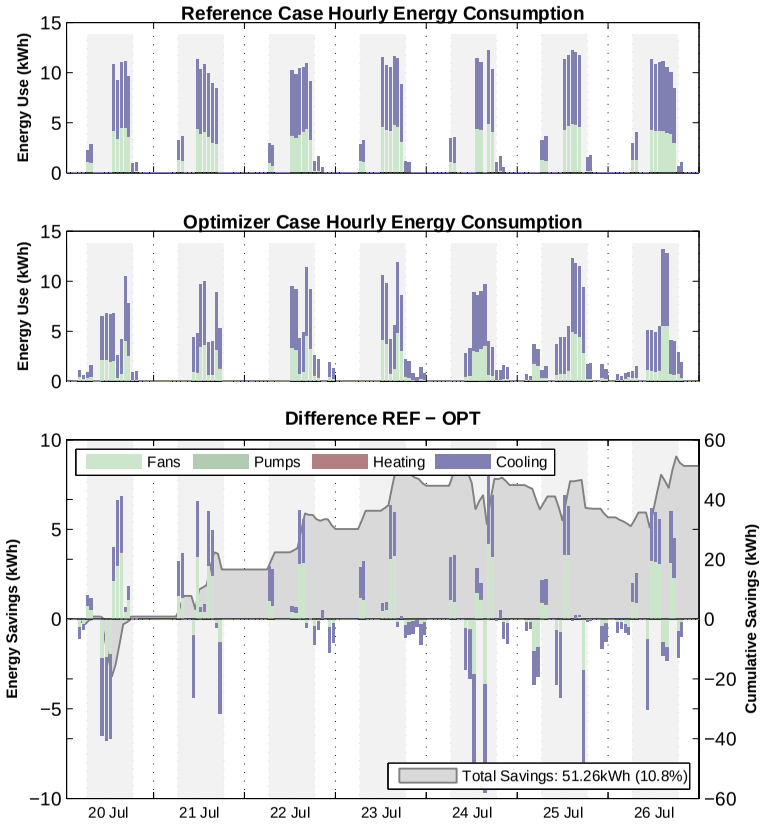


(b) Comfort

Figure D.7: Solution and comfort: MM1, cooling season, ASHRAE 55 adaptive comfort penalty



(a) Base vs. optimal



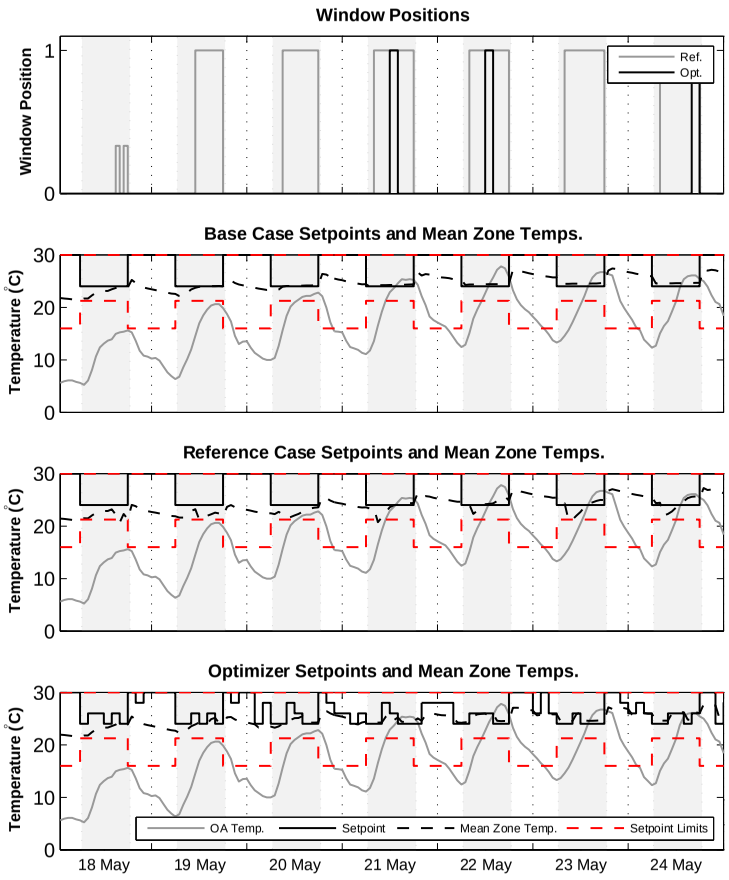
(b) Reference vs. optimal

Figure D.8: Savings: MM1, cooling season, ASHRAE 55 adaptive comfort penalty

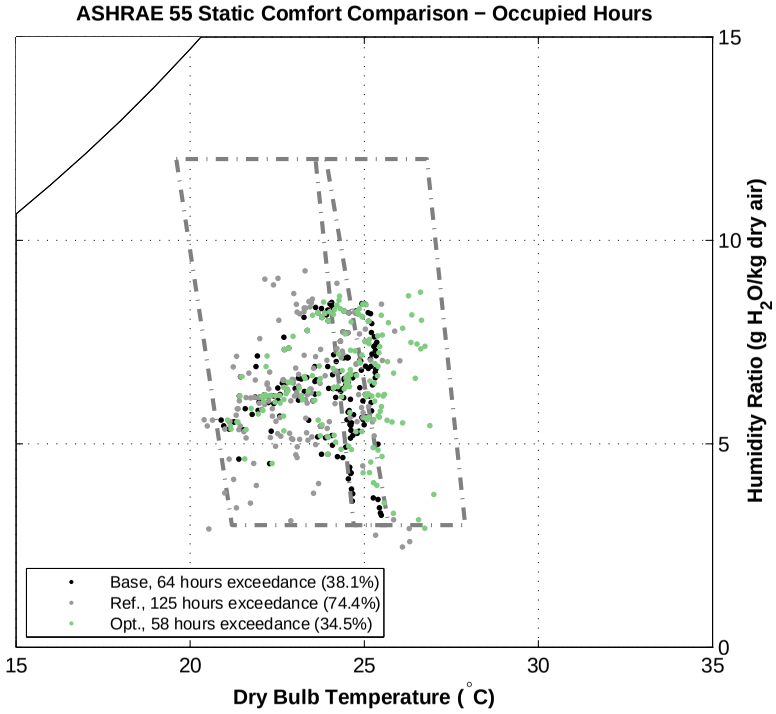
D.3 MM2 Partial Changeover Results

Table D.3: MM2 Energy and Comfort Solution Summary

	Swing Season			Cooling Season		
	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)
Base Case	234	14	15	532	0	0
Reference Case	261	118	1	581	0	0
Optimal Case						
55 static	219	14	15	473	0	0
55 adaptive	231	17	0	337	8	0
15251 adaptive	209	30	0	330	0	0

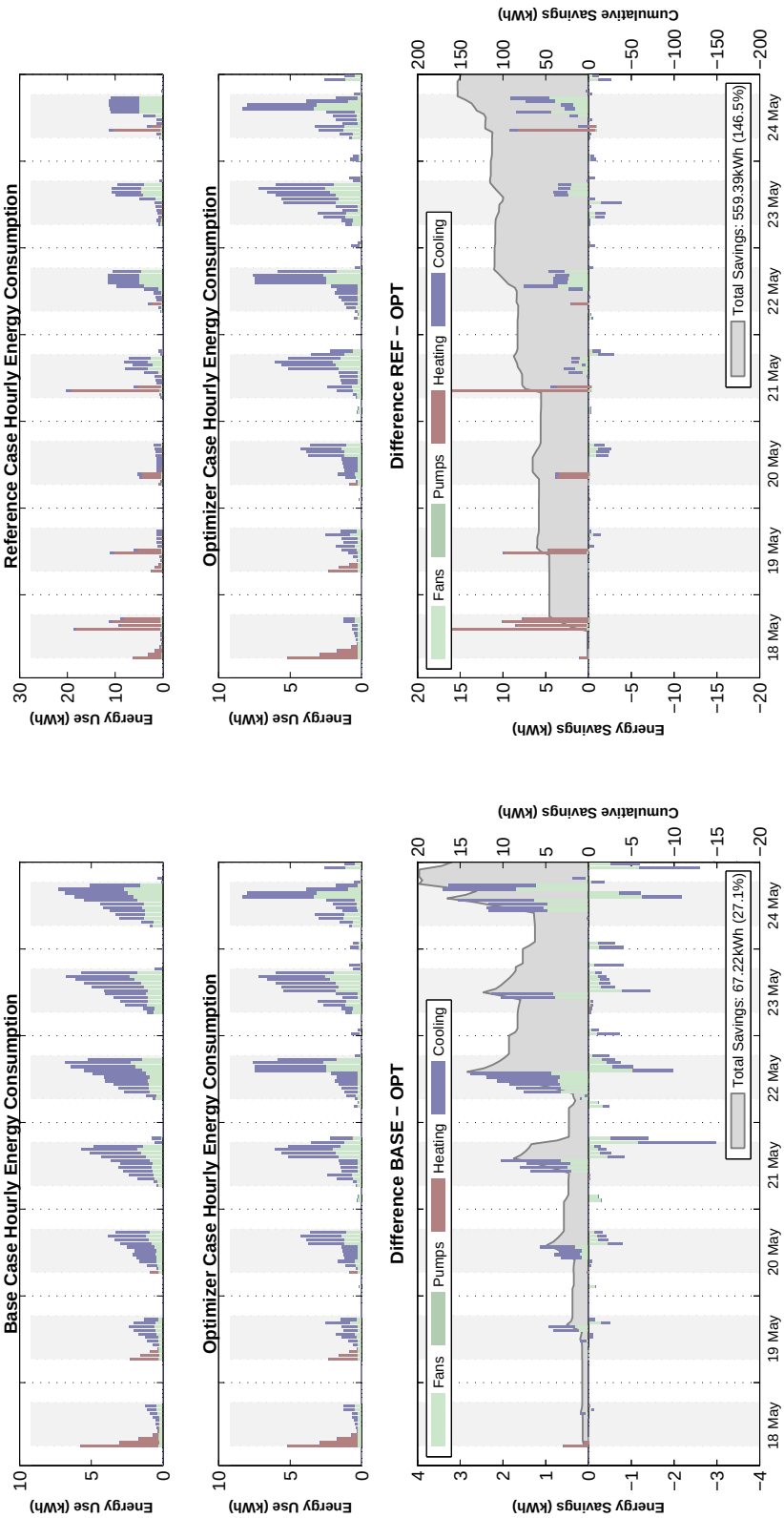


(a) Solution



(b) Comfort

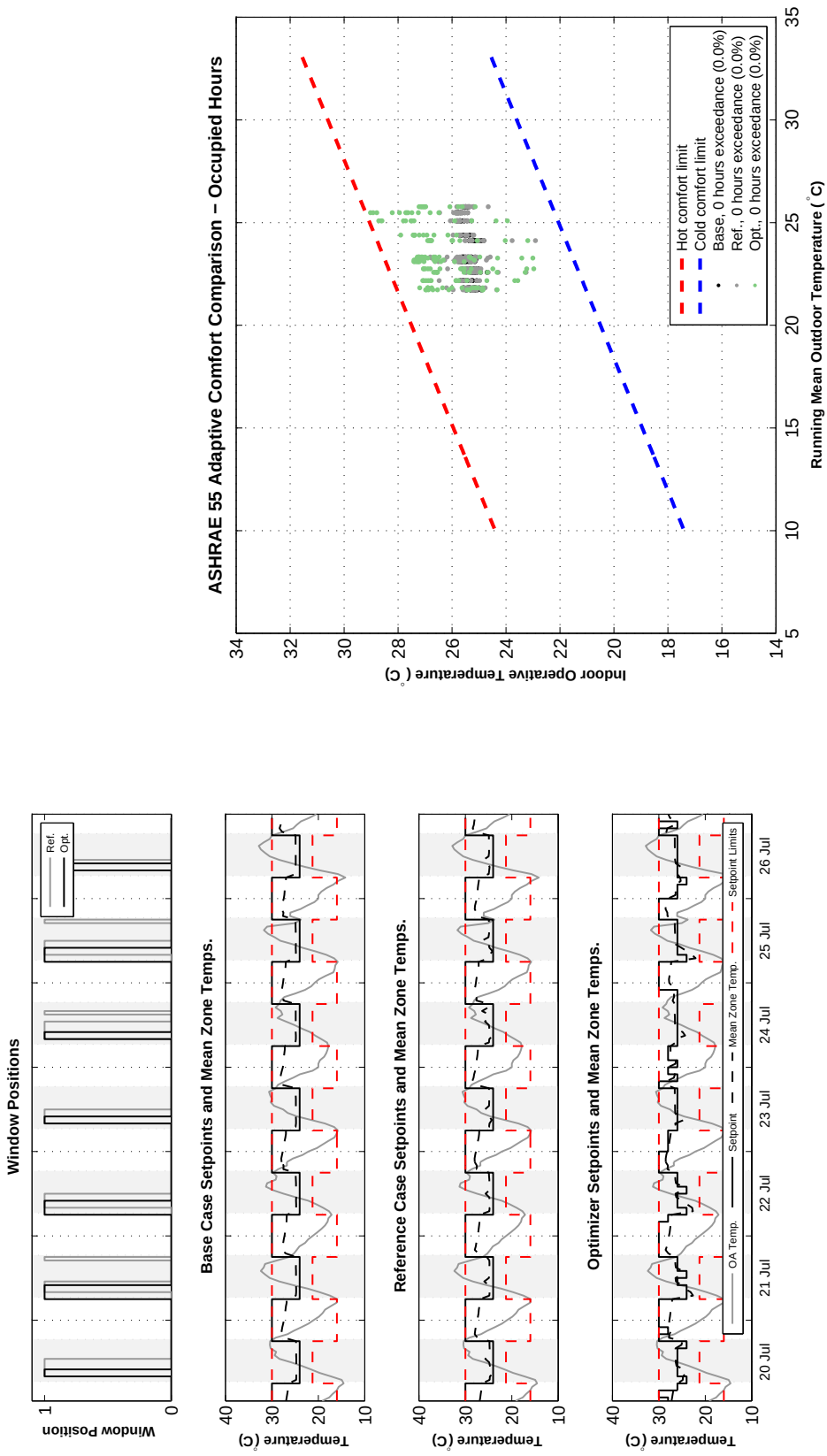
Figure D.9: Solution and comfort: MM2, swing season, ASHRAE 55 static comfort penalty



(a) Base vs. optimal

(b) Reference vs. optimal

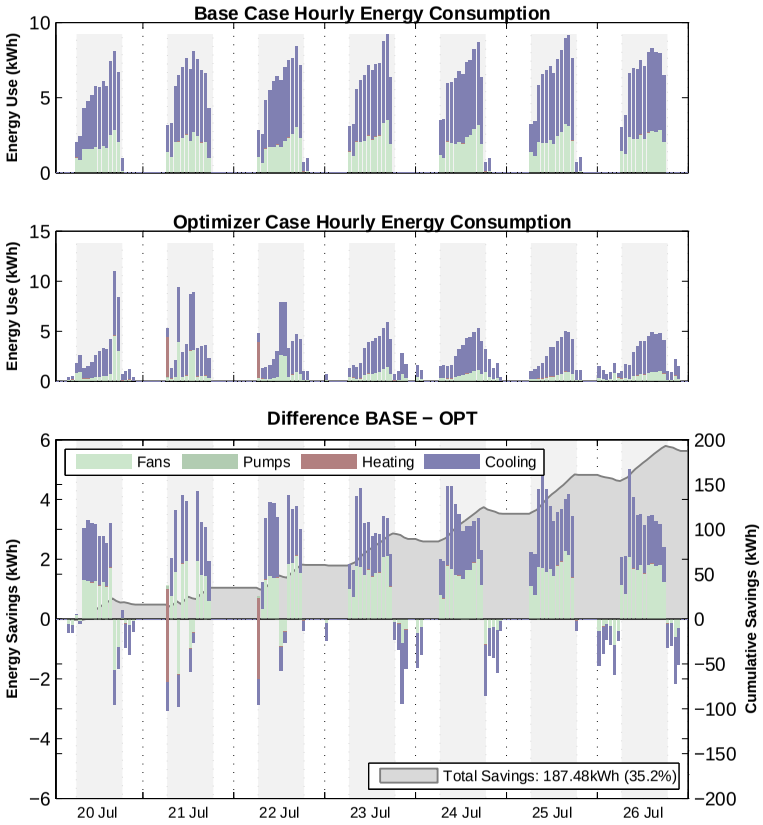
Figure D.10: Savings: MM2, swing season, ASHRAE 55 static comfort penalty



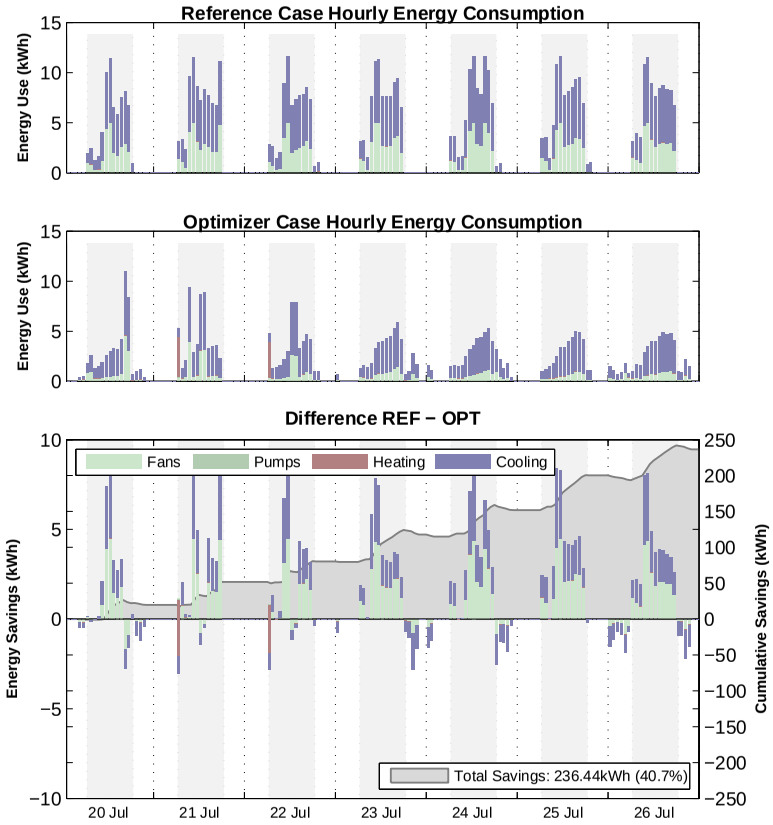
(a) Solution

(b) Comfort

Figure D.11: Solution and comfort: MM2, cooling season, ASHRAE 55 adaptive comfort penalty



(a) Base vs. optimal



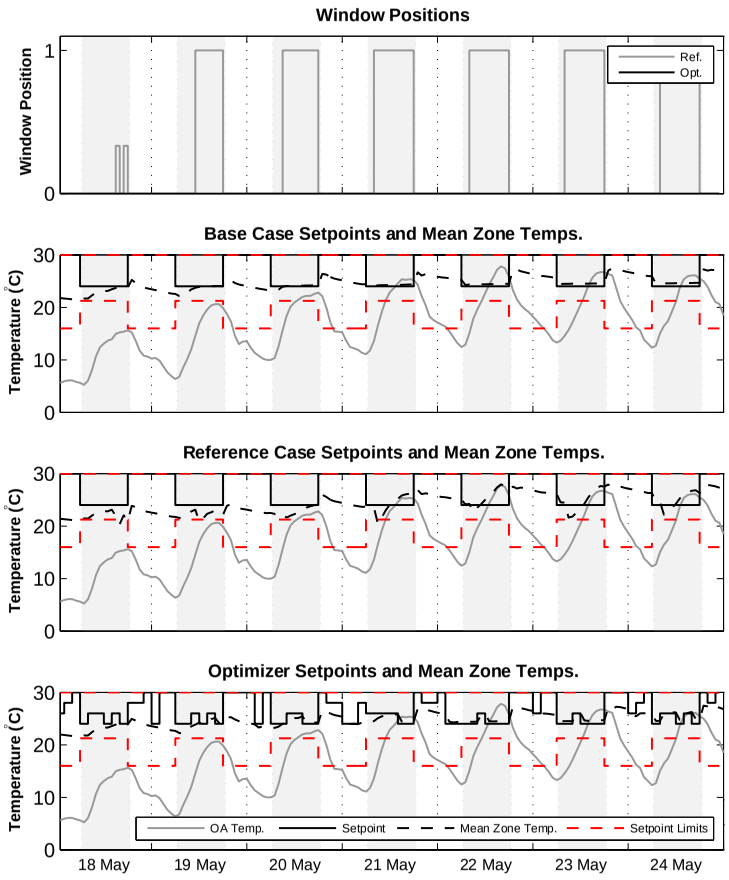
(b) Reference vs. optimal

Figure D.12: Savings: MM2, cooling season, ASHRAE 55 adaptive comfort penalty

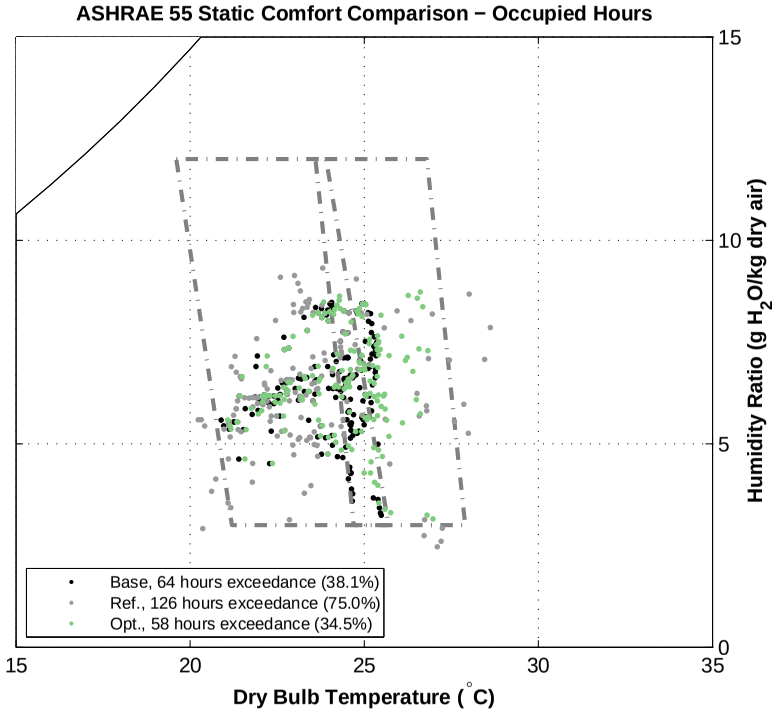
D.4 MM2 Changeover Results

Table D.4: MM2 Energy and Comfort Solution Summary

	Swing Season			Cooling Season		
	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)
Base Case	234	14	15	532	0	0
Reference Case	23	20	18	440	0	6
Optimal Case						
55 static	222	14	15	478	0	0
55 adaptive	31	20	18	266	0	1
15251 adaptive	37	15	0	337	0	0

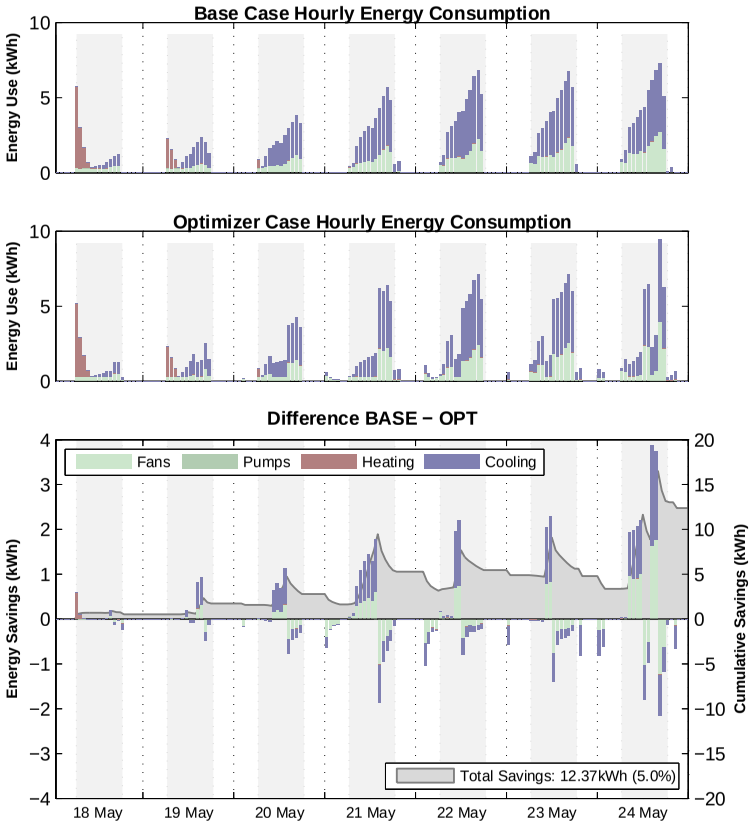


(a) Solution

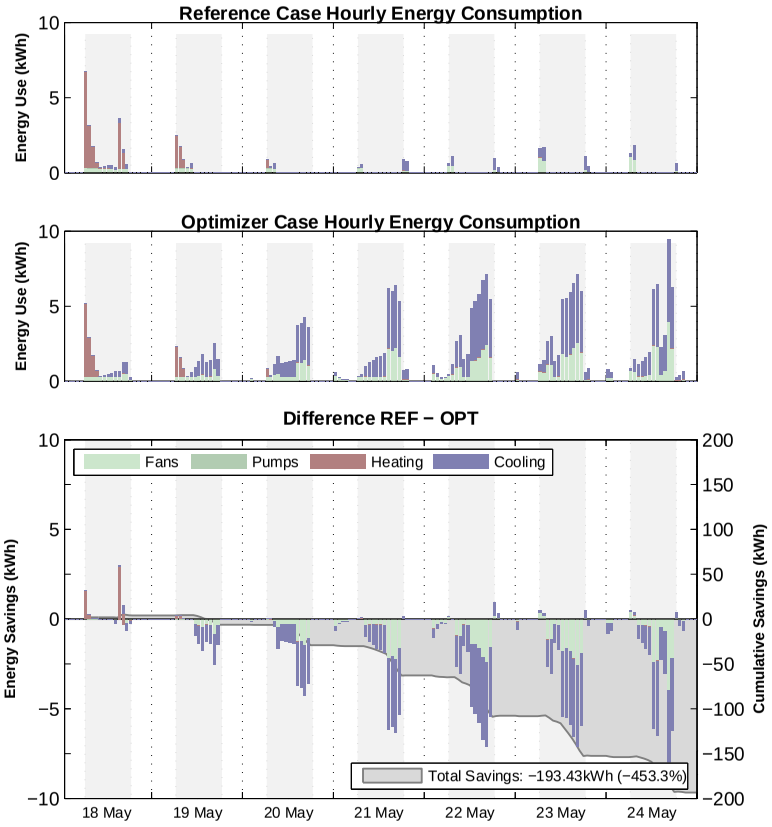


(b) Comfort

Figure D.13: Solution and comfort: MM2, swing season, ASHRAE 55 static comfort penalty

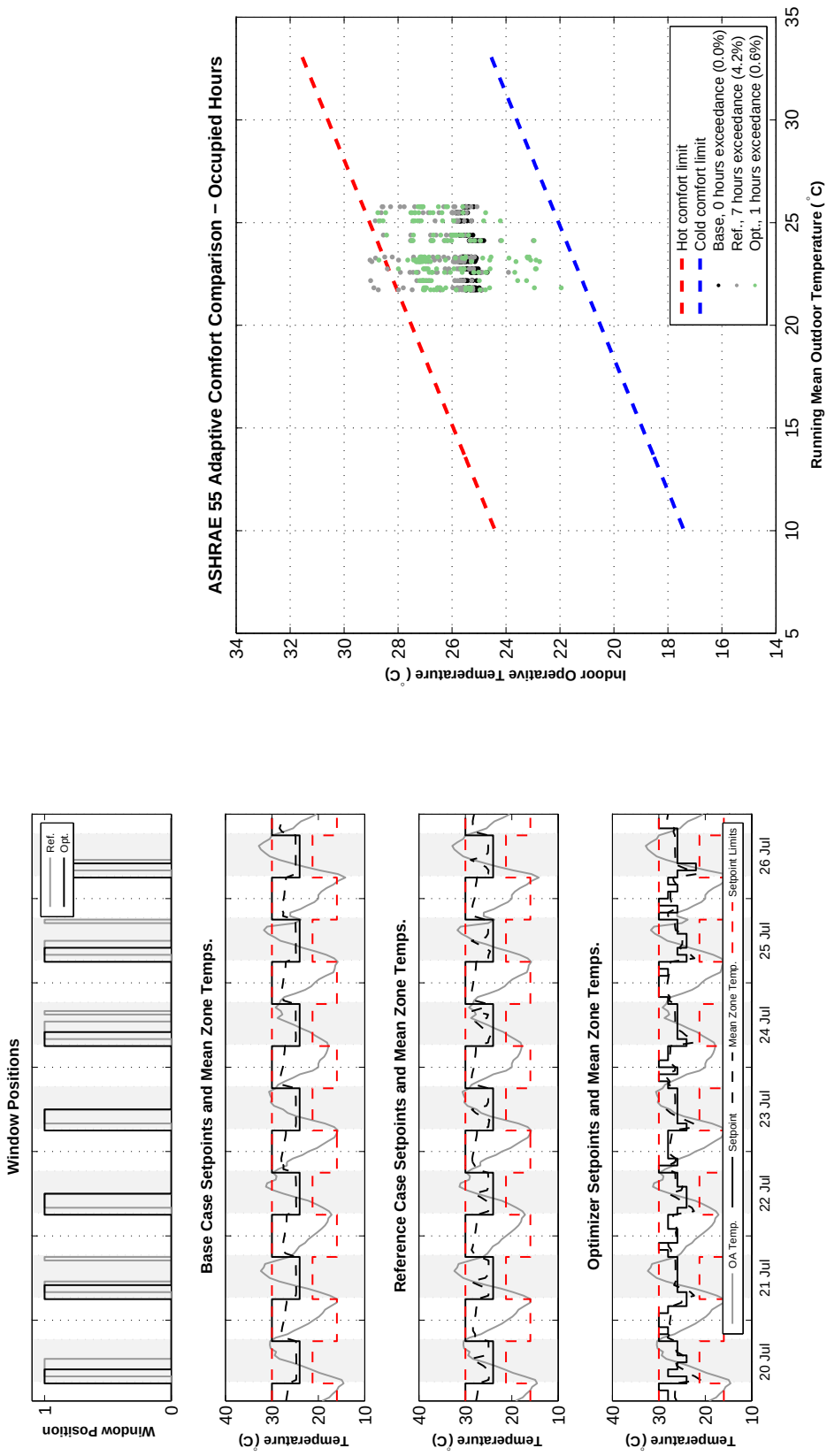


(a) Base vs. optimal



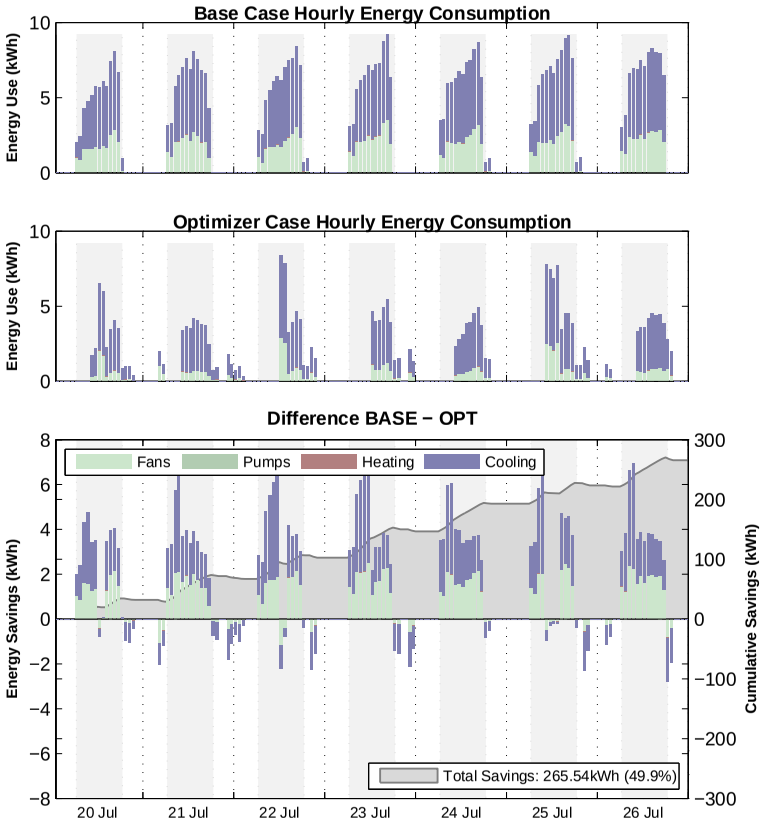
(b) Reference vs. optimal

Figure D.14: Savings: MM2, swing season, ASHRAE 55 static comfort penalty

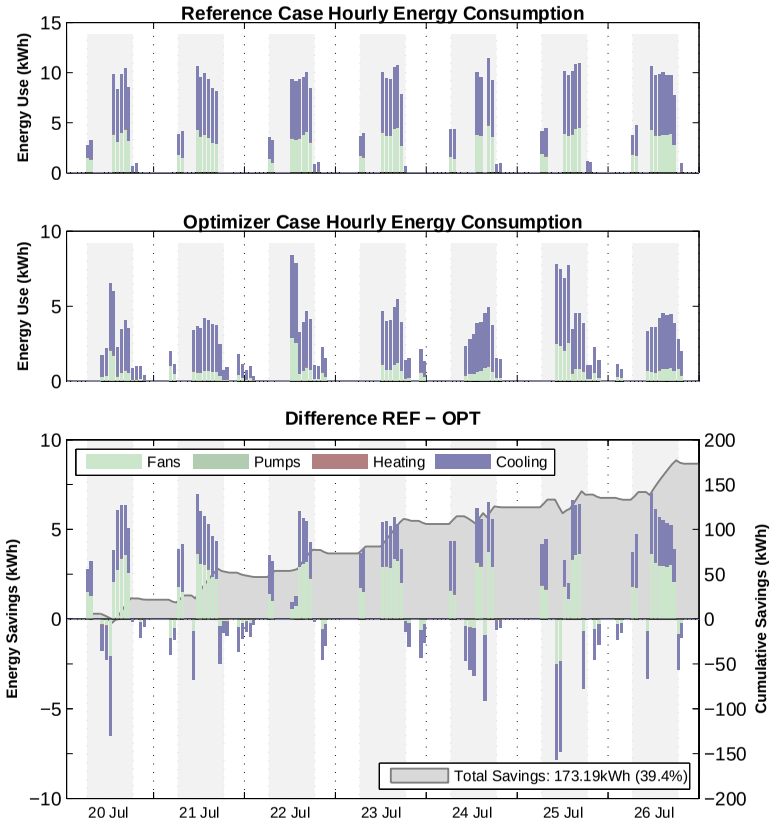


(a) Solution (b) Comfort

Figure D.15: Solution and comfort: MM2, cooling season, ASHRAE 55 adaptive comfort penalty



(a) Base vs. optimal



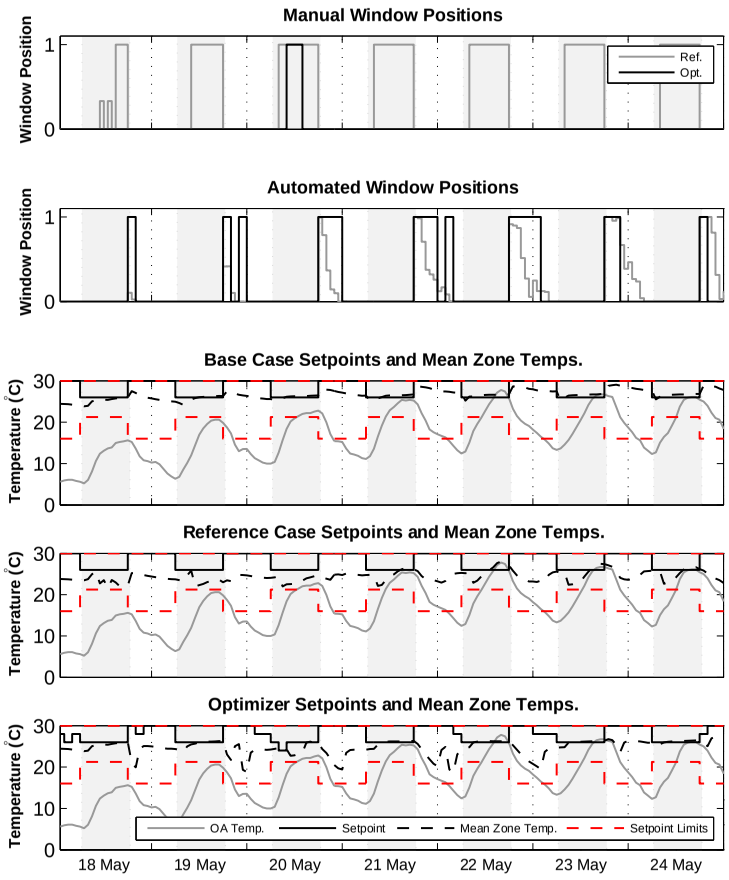
(b) Reference vs. optimal

Figure D.16: Savings: MM2, cooling season, ASHRAE 55 adaptive comfort penalty

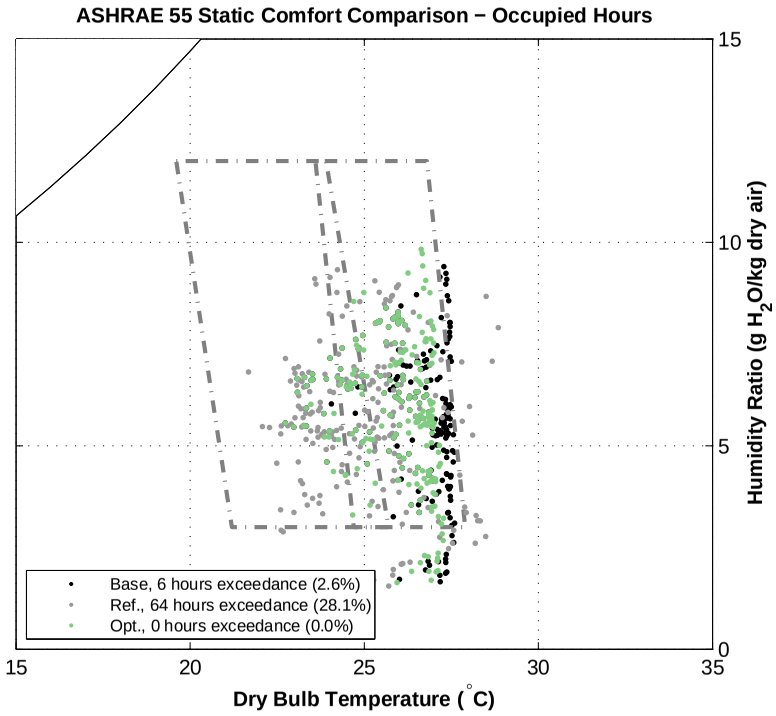
D.5 MM3 Partial Changeover Results

Table D.5: MM3 Energy and Comfort Solution Summary

	Swing Season			Cooling Season		
	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)
Base Case	167	17	6	372	0	50
Reference Case	5	440	30	188	1	4
Optimal Case						
55 static	105	26	0	211	1	0
55 adaptive	121	67	21	194	5	0
15251 adaptive	109	27	0	190	2	0

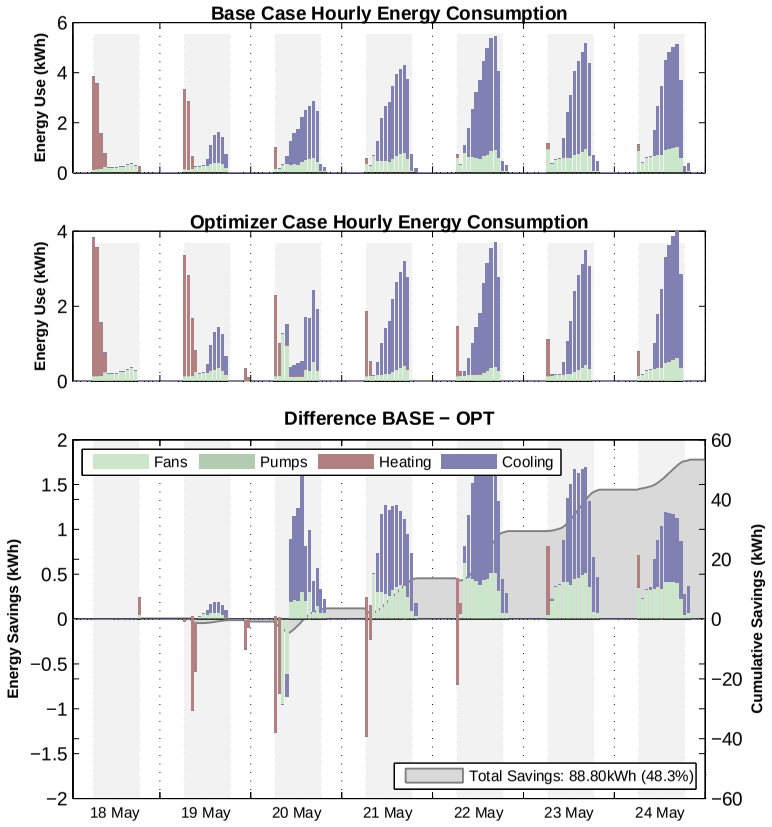


(a) Solution

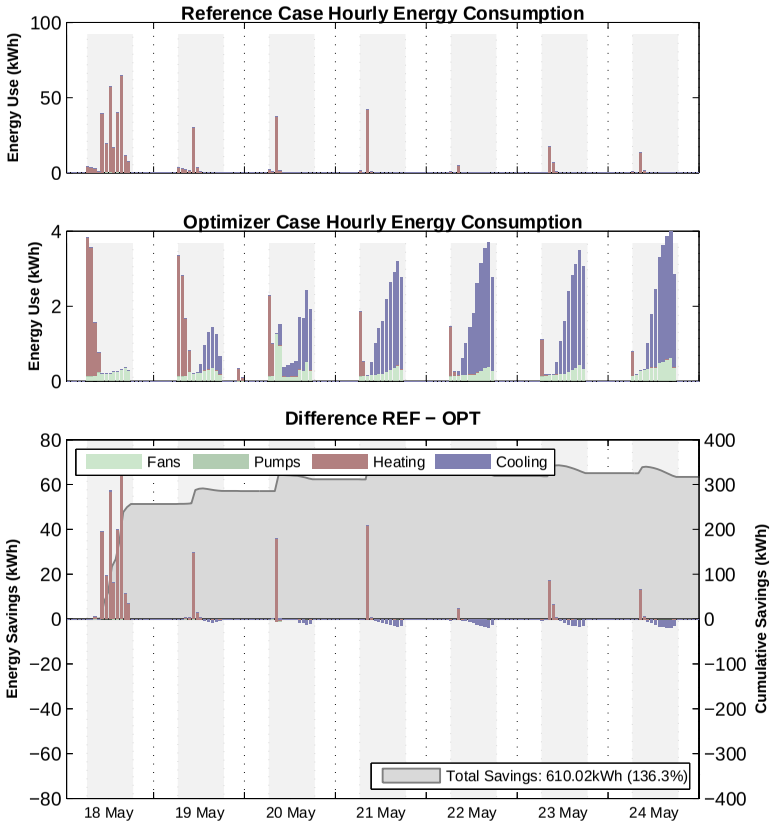


(b) Comfort

Figure D.17: Solution and comfort: MM3, swing season, ASHRAE 55 static comfort penalty

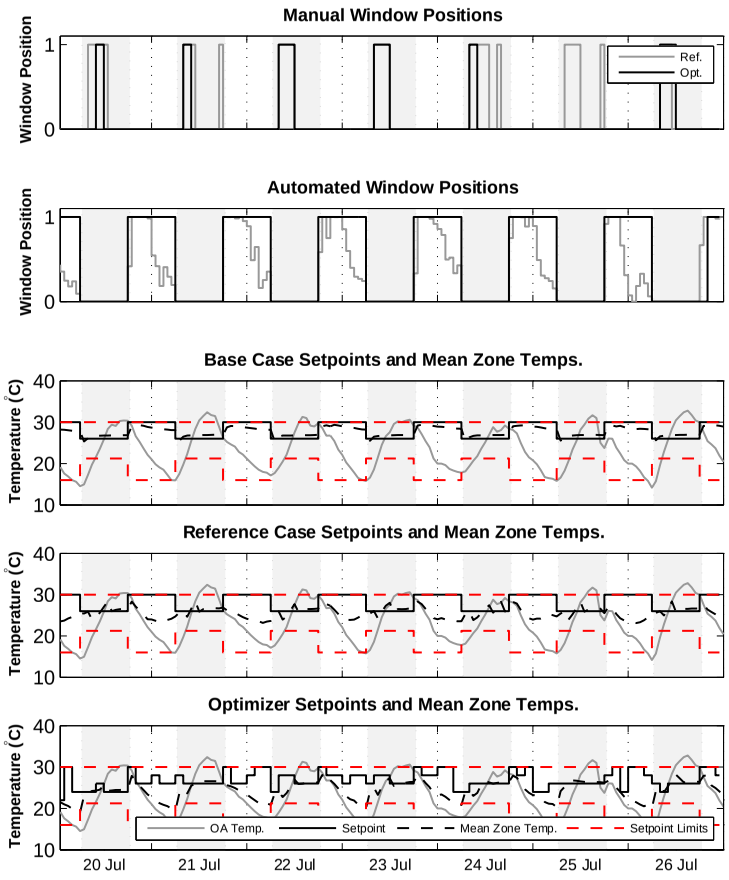


(a) Base vs. optimal

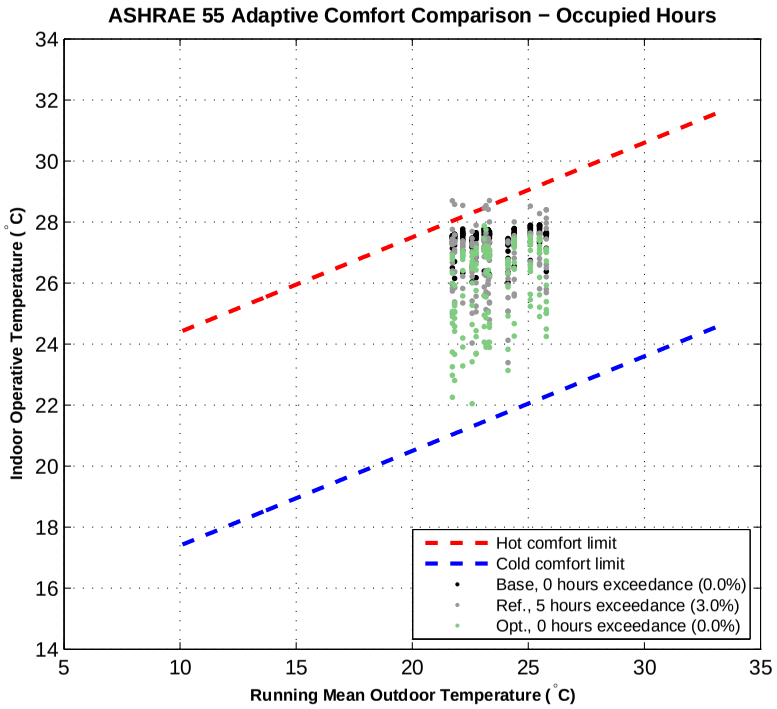


(b) Reference vs. optimal

Figure D.18: Savings: MM3, swing season, ASHRAE 55 static comfort penalty

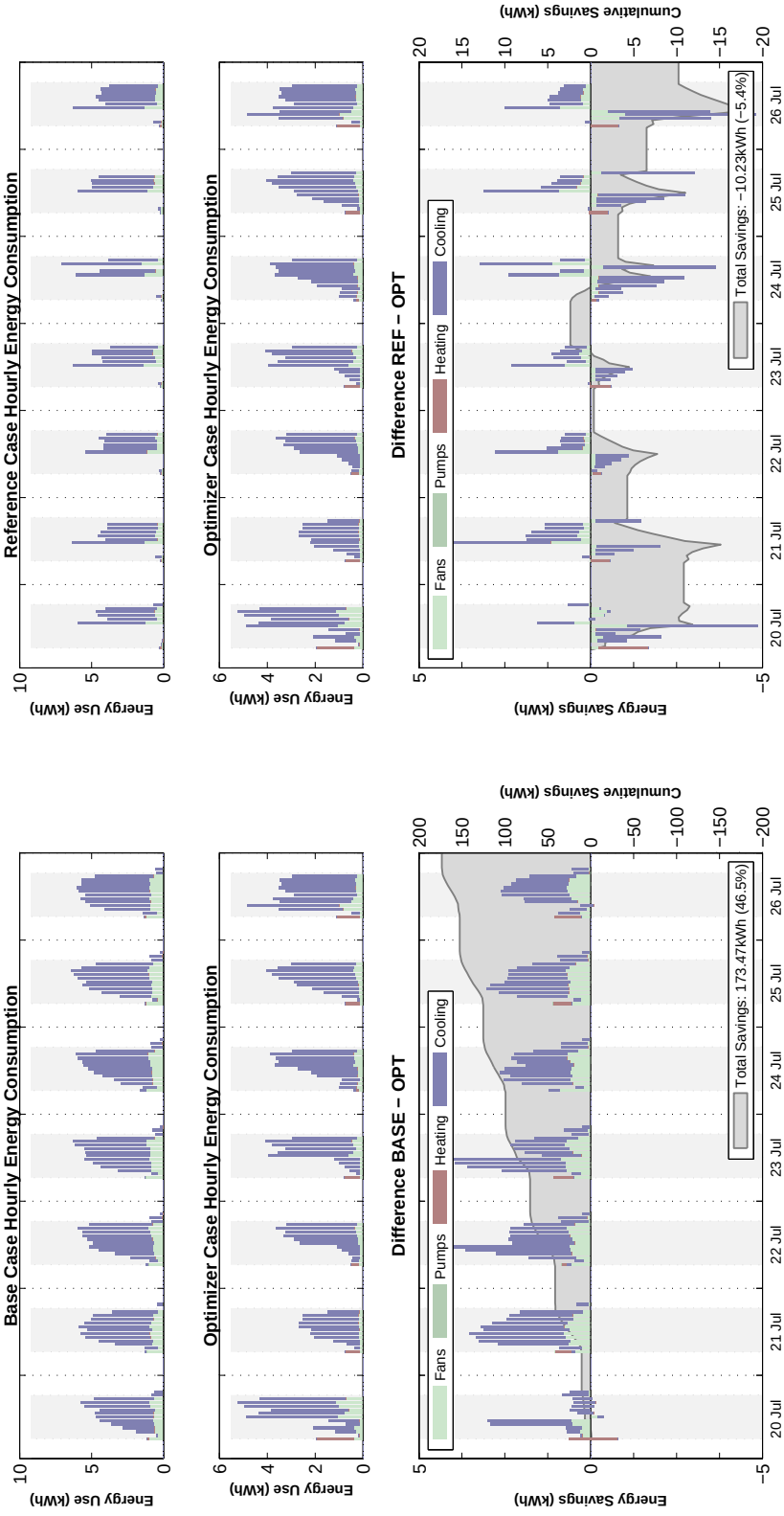


(a) Solution



(b) Comfort

Figure D.19: Solution and comfort: MM3, cooling season, ASHRAE 55 adaptive comfort penalty



(a) Base vs. optimal

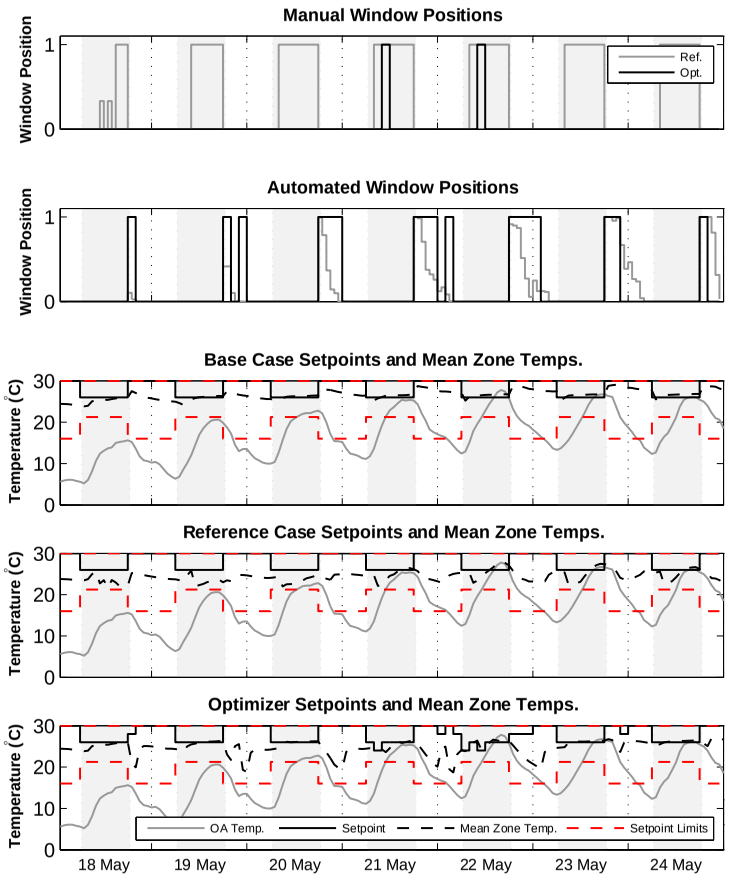
(b) Reference vs. optimal

Figure D.20: Savings: MM3, cooling season, ASHRAE 55 adaptive comfort penalty

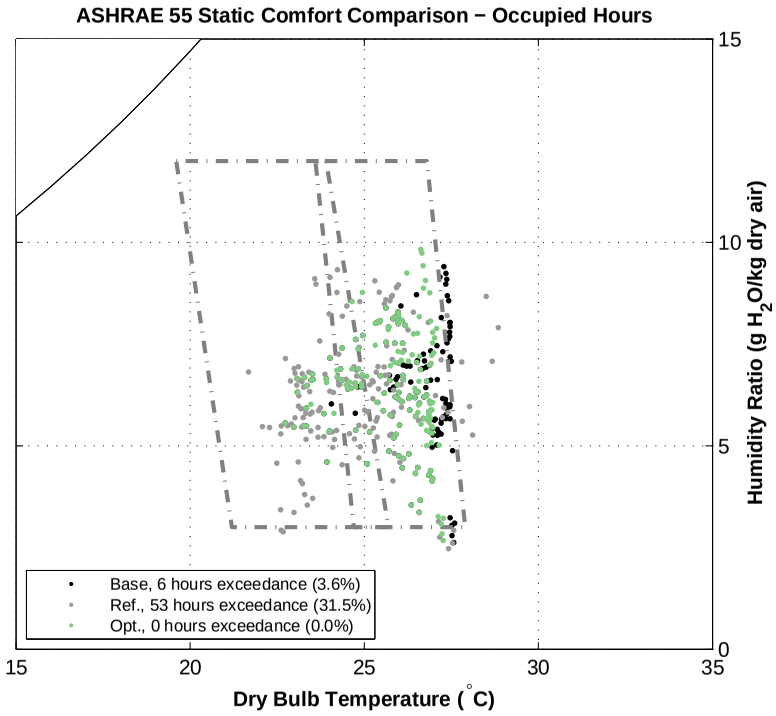
D.6 MM3 Changeover Results

Table D.6: MM3 Energy and Comfort Solution Summary

	Swing Season			Cooling Season		
	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)
Base Case	167	17	6	372	0	50
Reference Case	5	440	30	189	0	4
Optimal Case						
55 static	103	24	0	211	3	0
55 adaptive	57	61	18	177	6	0
15251 adaptive	44	26	0	173	2	0

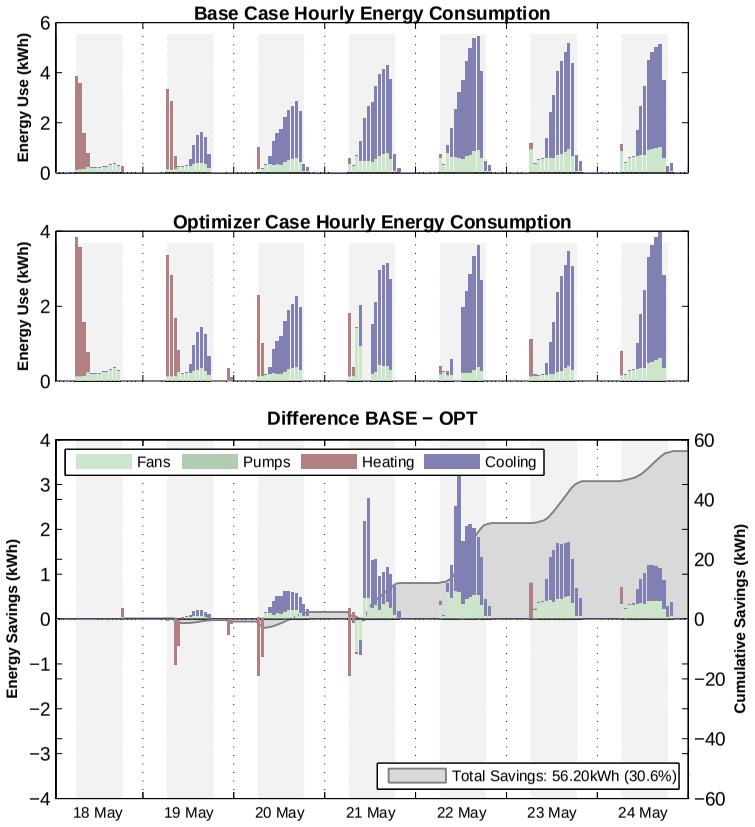


(a) Solution

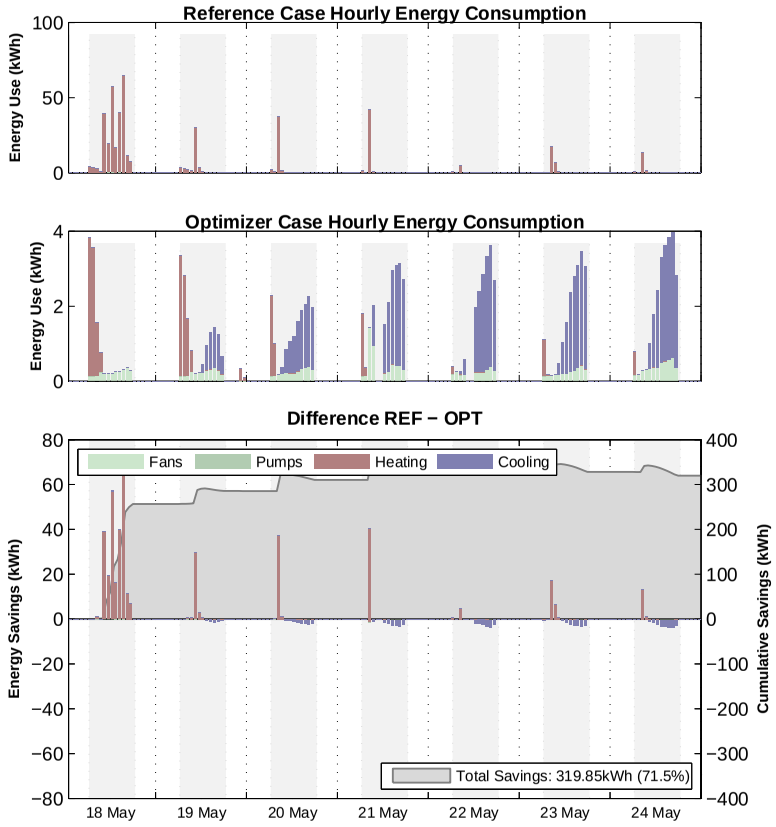


(b) Comfort

Figure D.21: Solution and comfort: MM3, swing season, ASHRAE 55 static comfort penalty

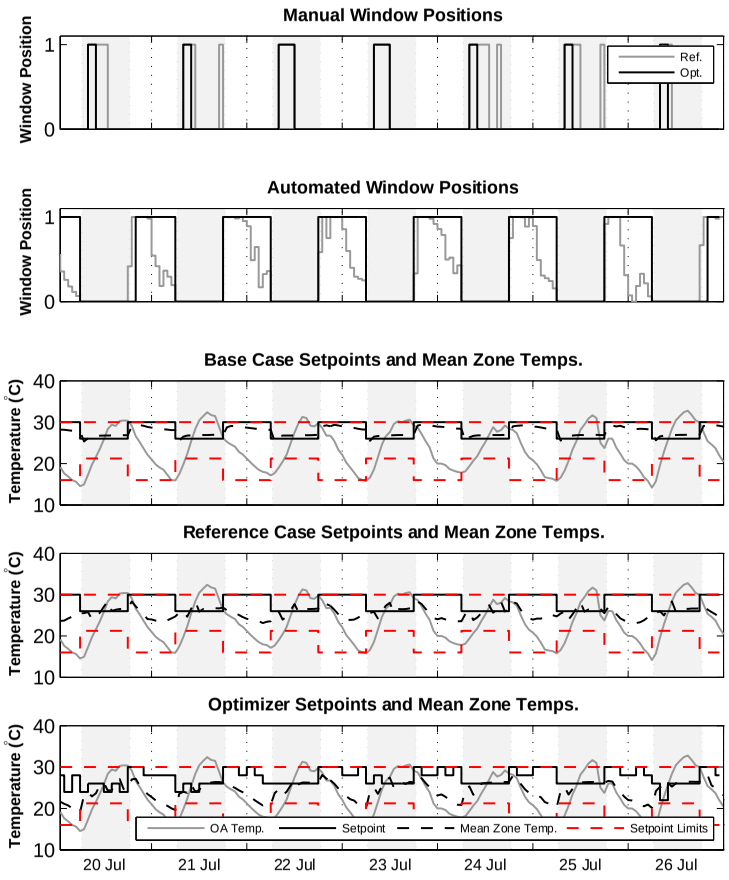


(a) Base vs. optimal

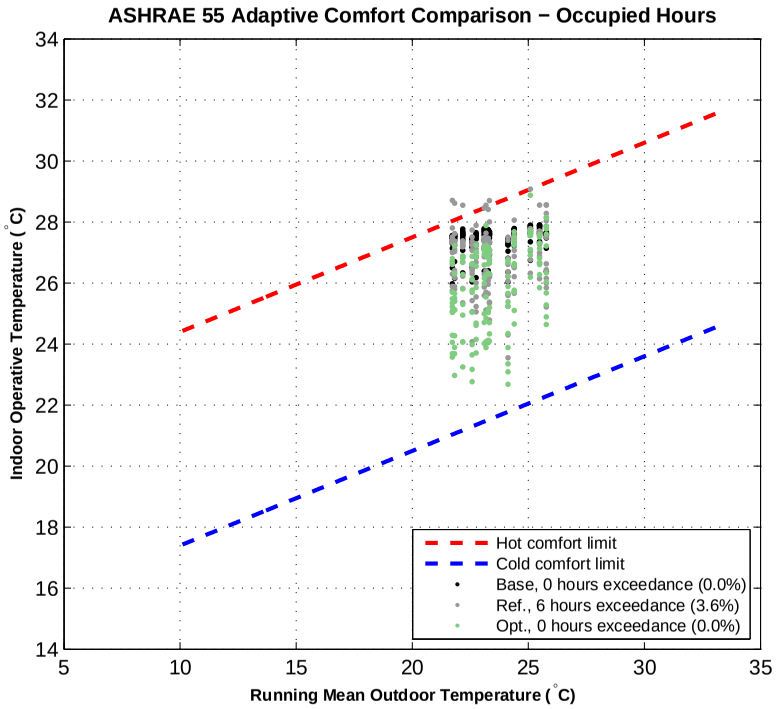


(b) Reference vs. optimal

Figure D.22: Savings: MM3, swing season, ASHRAE 55 static comfort penalty

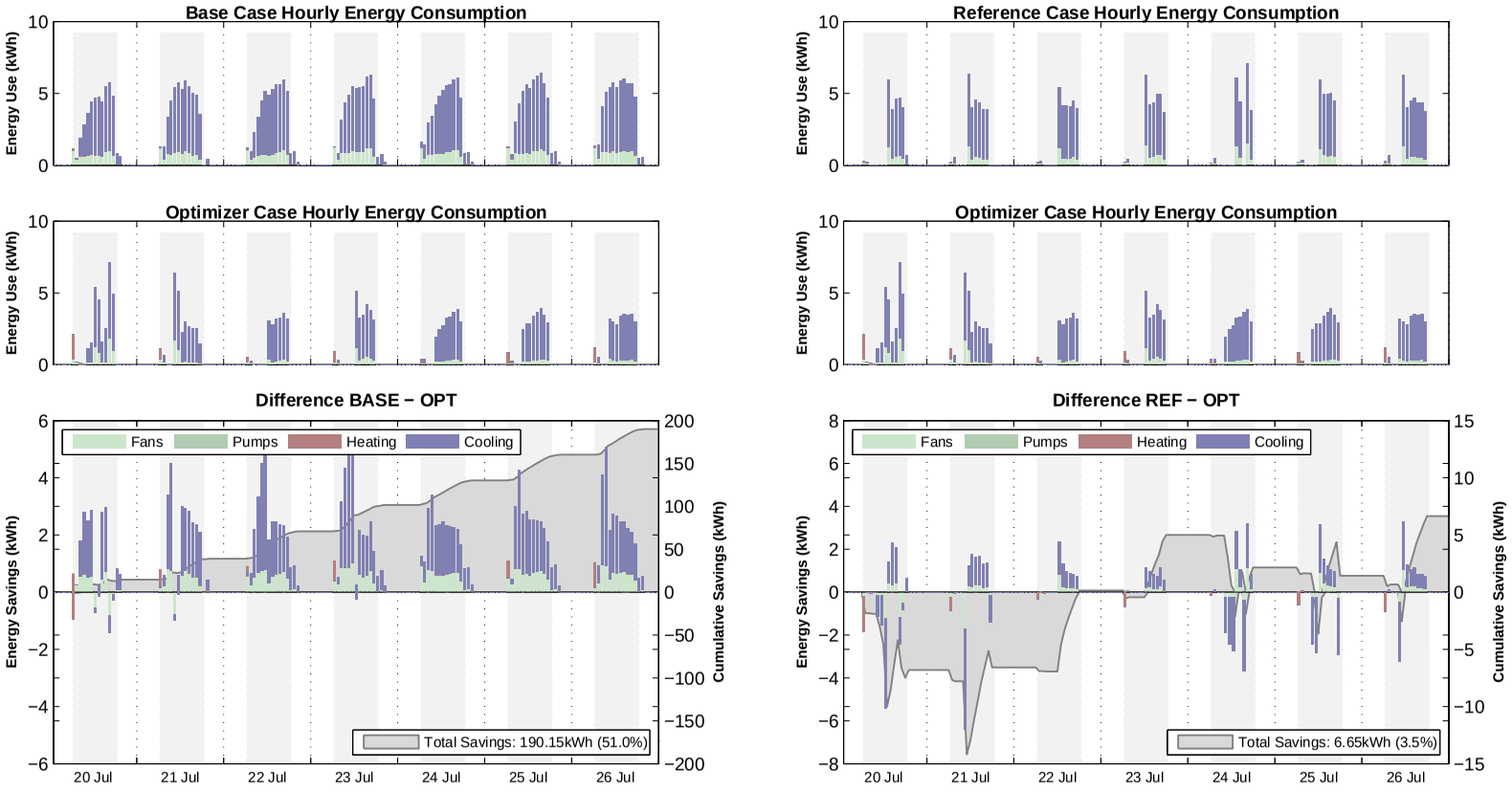


(a) Solution



(b) Comfort

Figure D.23: Solution and comfort: MM3, cooling season, ASHRAE 55 adaptive comfort penalty



(a) Base vs. optimal

(b) Reference vs. optimal

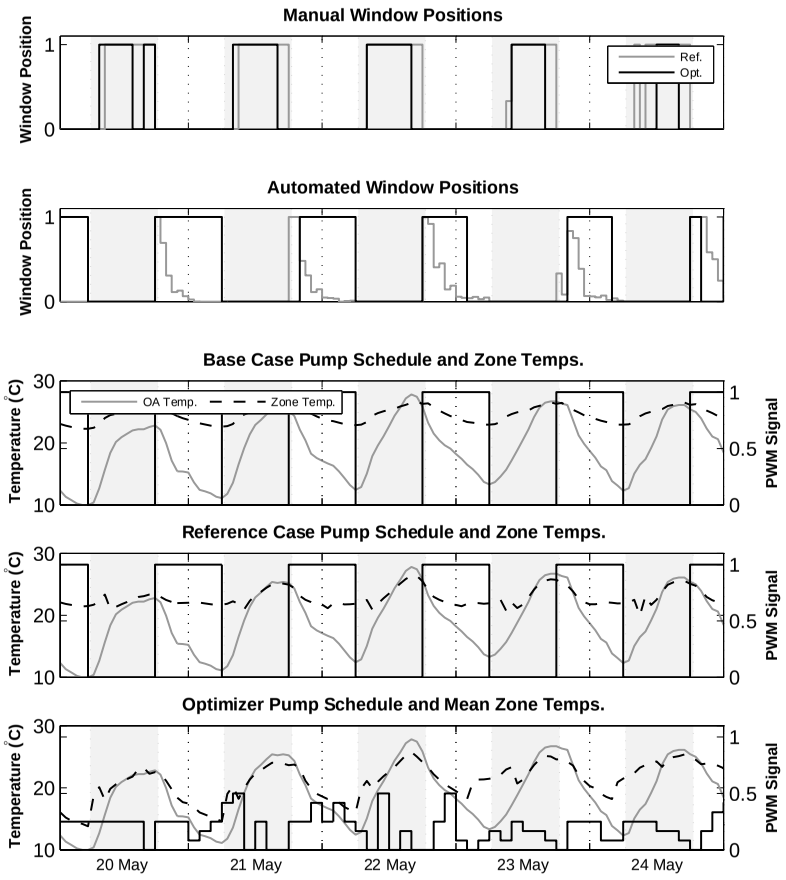
Figure D.24: Savings: MM3, cooling season, ASHRAE 55 adaptive comfort penalty

D.7 MM4 Partial Changeover Results

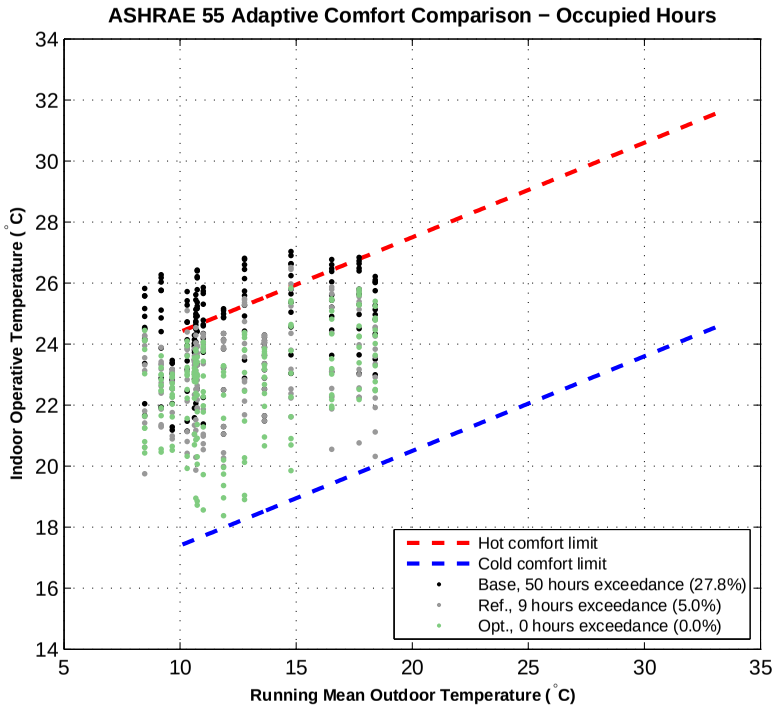
Due to the higher mass and longer warmup periods associated with MM4, the first two days of the swing season period have been clipped to eliminate lingering portions of the solution exhibiting warmup behavior.

Table D.7: MM4 Energy and Comfort Solution Summary

	Swing Season			Cooling Season		
	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)
Base Case	88	0	16	267	0	14
Reference Case	33	0	10	241	0	0
Optimal Case						
55 static	77	0	13	247	0	10
55 adaptive	70	0	3	124	0	10
15251 adaptive	70	0	0	124	0	0

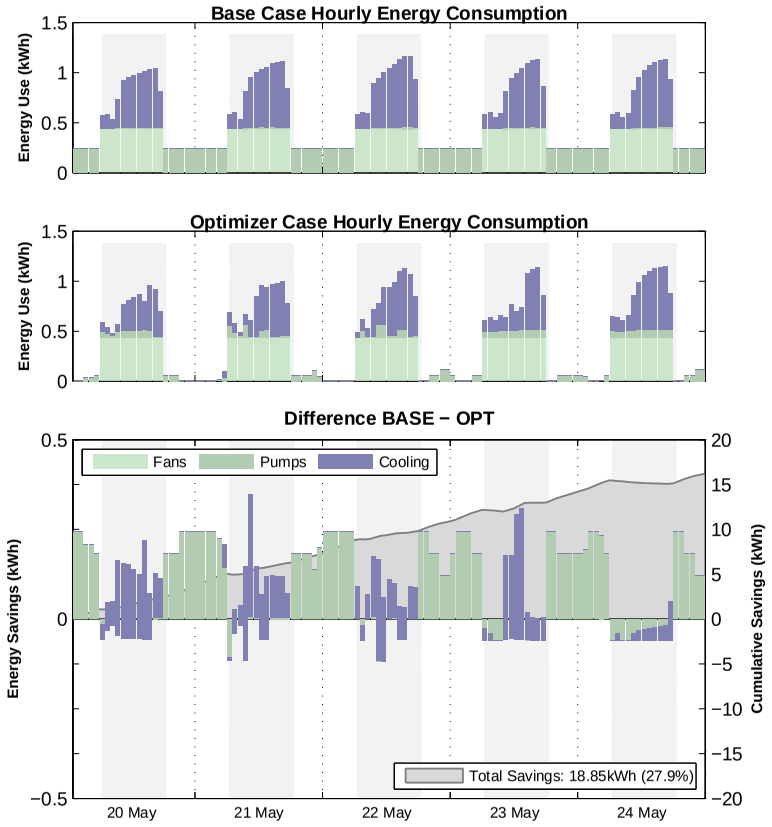


(a) Solution

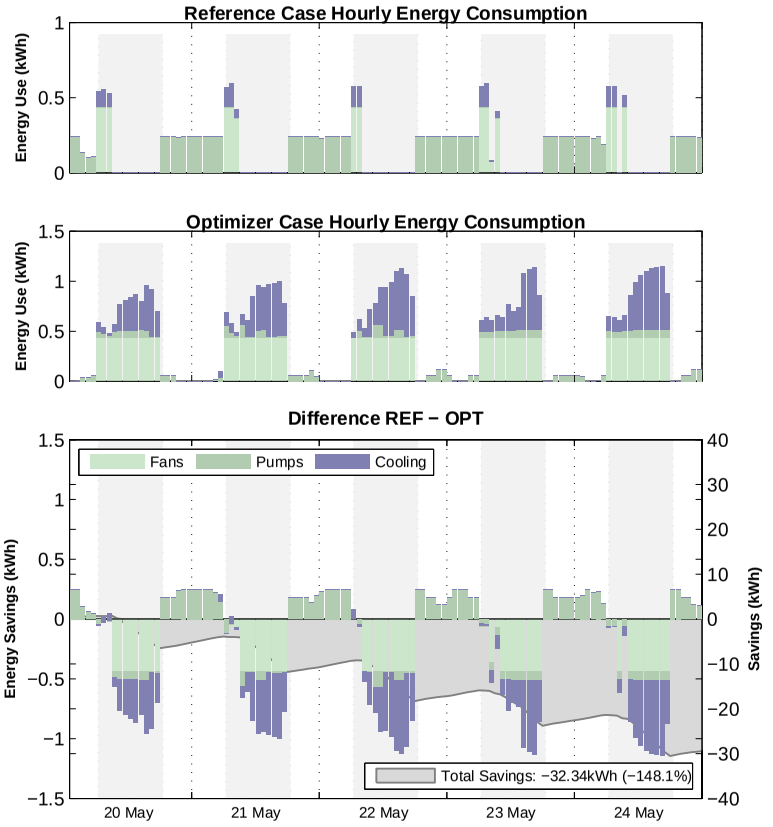


(b) Comfort

Figure D.25: Solution and comfort: MM4, swing season, ASHRAE 55 static comfort penalty

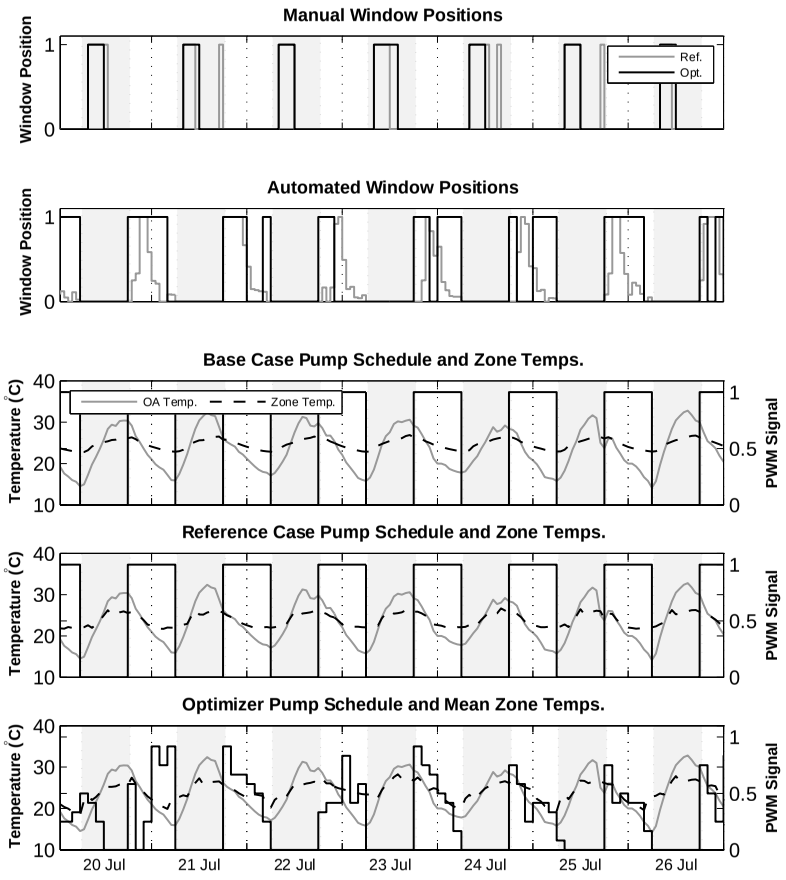


(a) Base vs. optimal

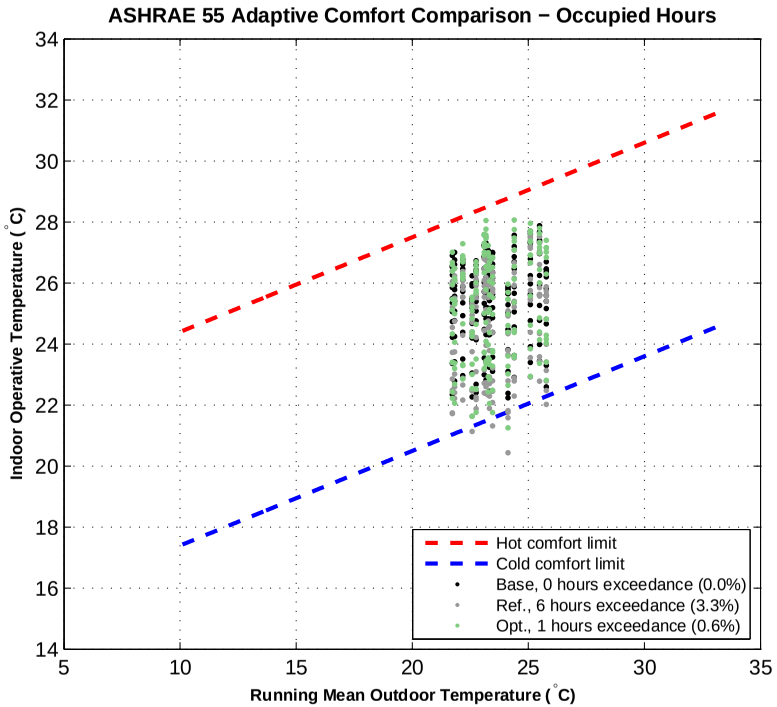


(b) Reference vs. optimal

Figure D.26: Savings: MM4, swing season, ASHRAE 55 static comfort penalty

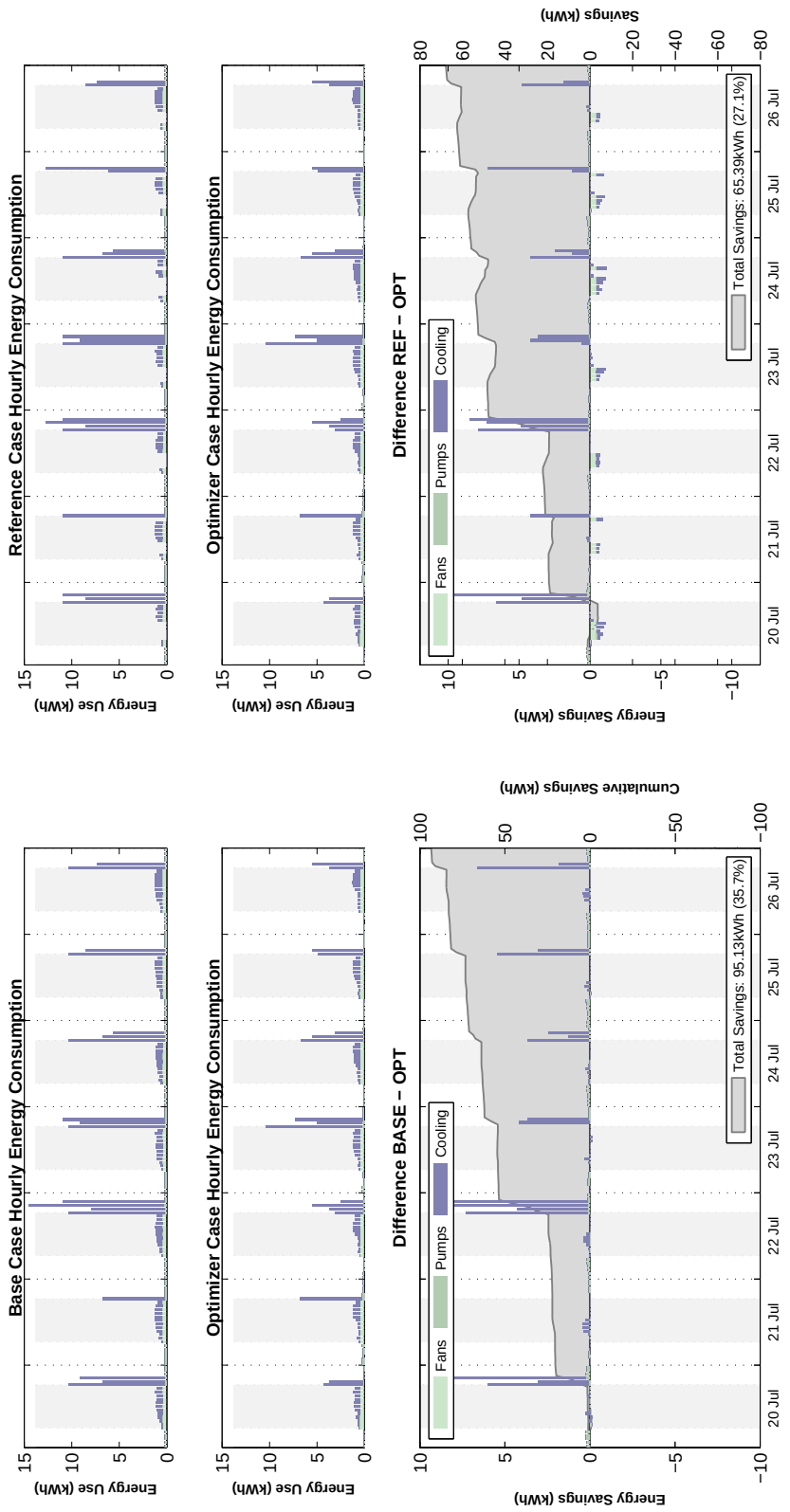


(a) Solution



(b) Comfort

Figure D.27: Solution and comfort: MM4, cooling season, ASHRAE 55 adaptive comfort penalty



(a) Base vs. optimal

(b) Reference vs. optimal

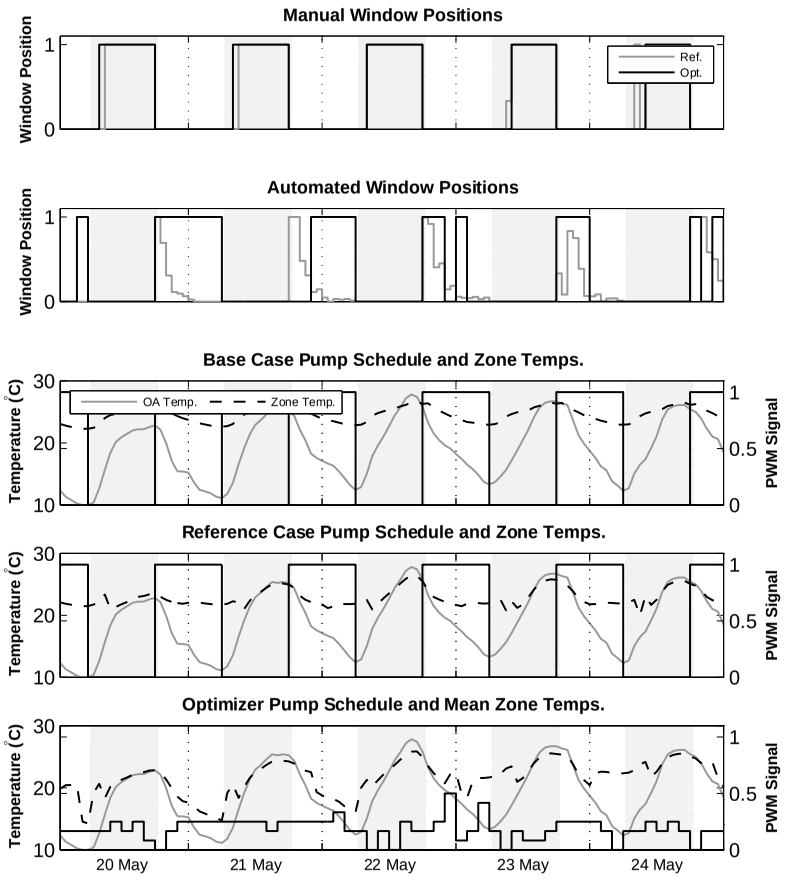
Figure D.28: Savings: MM4, cooling season, ASHRAE 55 adaptive comfort penalty

D.8 MM4 Changeover Results

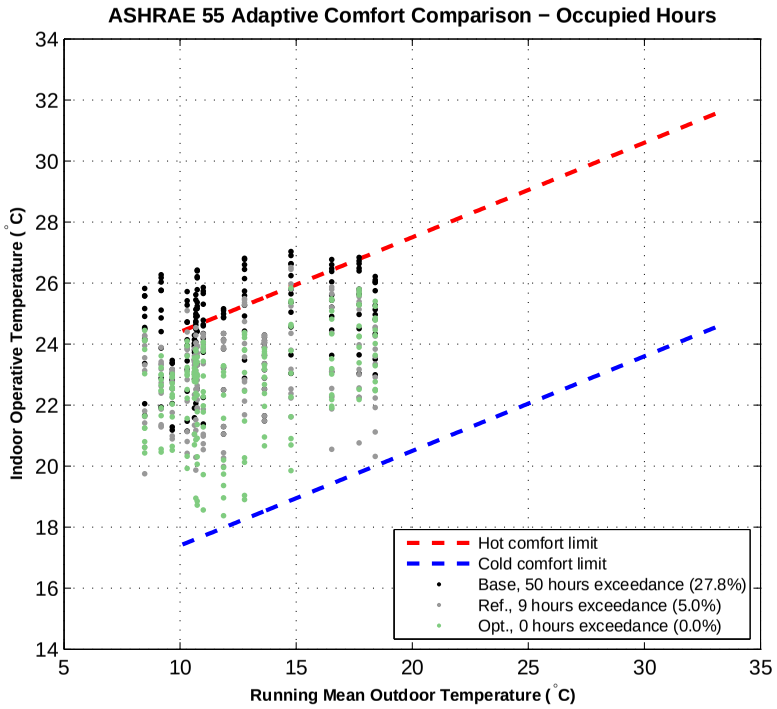
As with partial changeover solutions for MM4, the first two days of the swing season period have been clipped to avoid portions of the solution still exhibiting warmup behavior.

Table D.8: MM4 Energy and Comfort Solution Summary

	Swing Season			Cooling Season		
	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)	HVAC Electric Use (kWh)	HVAC Gas Use (kWh)	Comfort Violations (h)
Base Case	88	0	16	267	0	14
Reference Case	31	0	11	241	0	0
Optimal Case						
55 static	57	0	14	284	0	10
55 adaptive	33	0	8	64	0	7
15251 adaptive	33	0	0	54	0	0

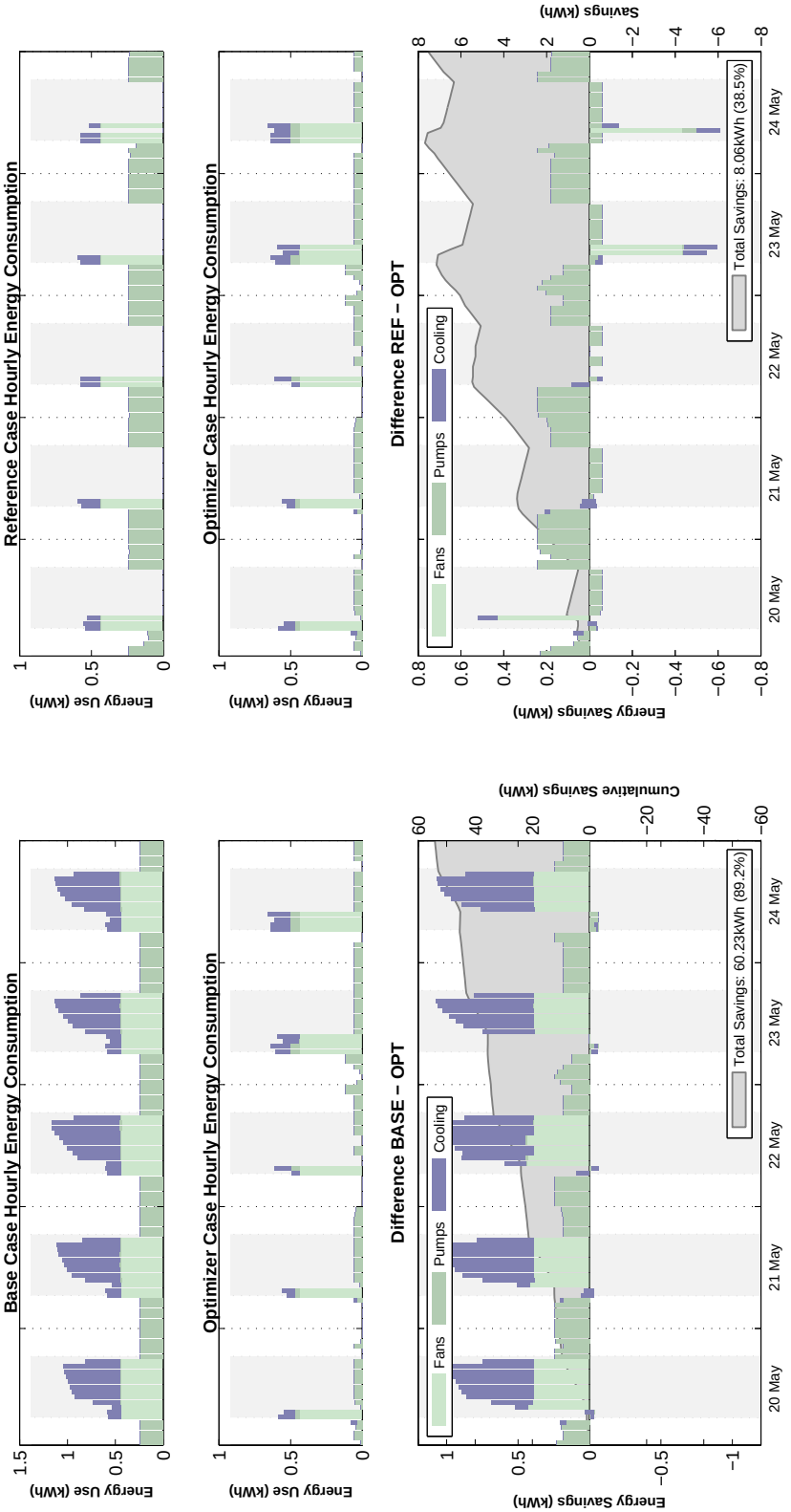


(a) Solution



(b) Comfort

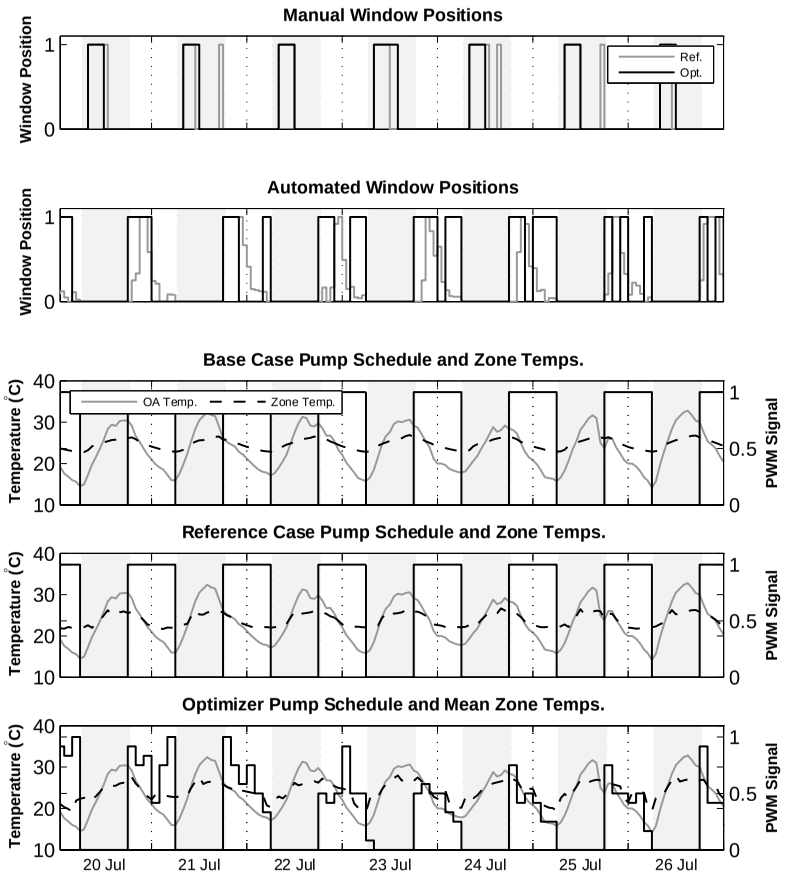
Figure D.29: Solution and comfort: MM4, swing season, ASHRAE 55 static comfort penalty



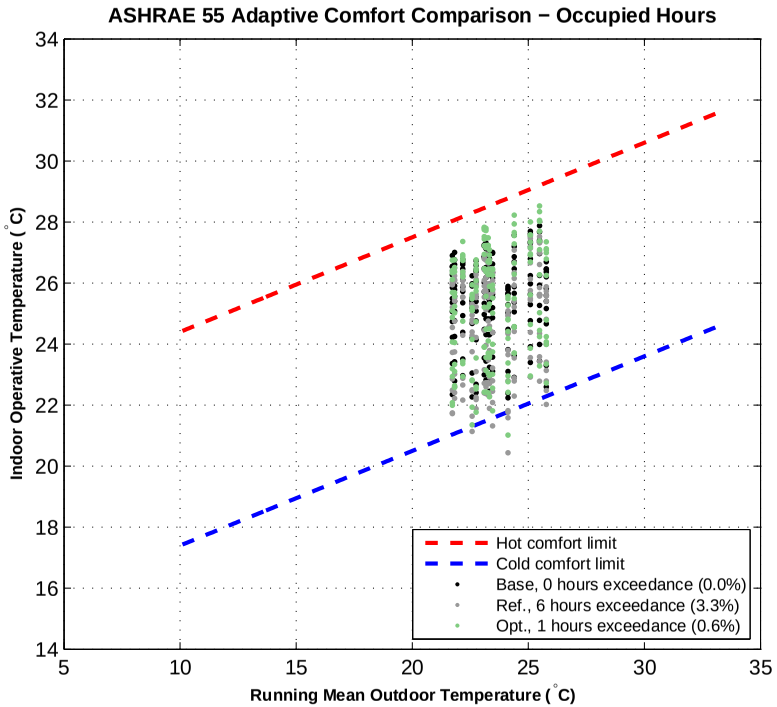
(a) Base vs. optimal

(b) Reference vs. optimal

Figure D.30: Savings: MM4, swing season, ASHRAE 55 static comfort penalty

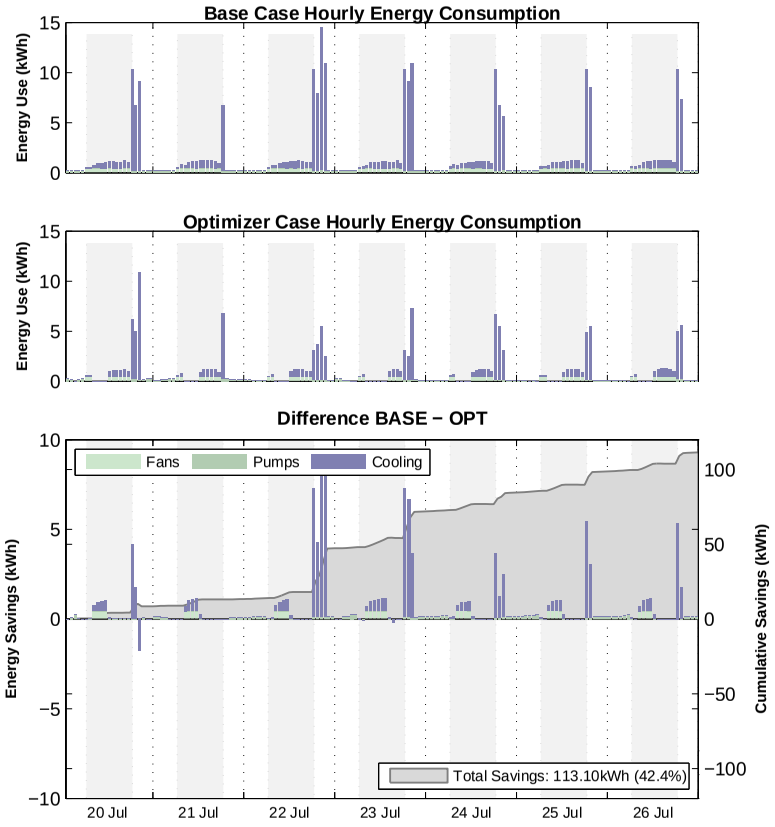


(a) Solution

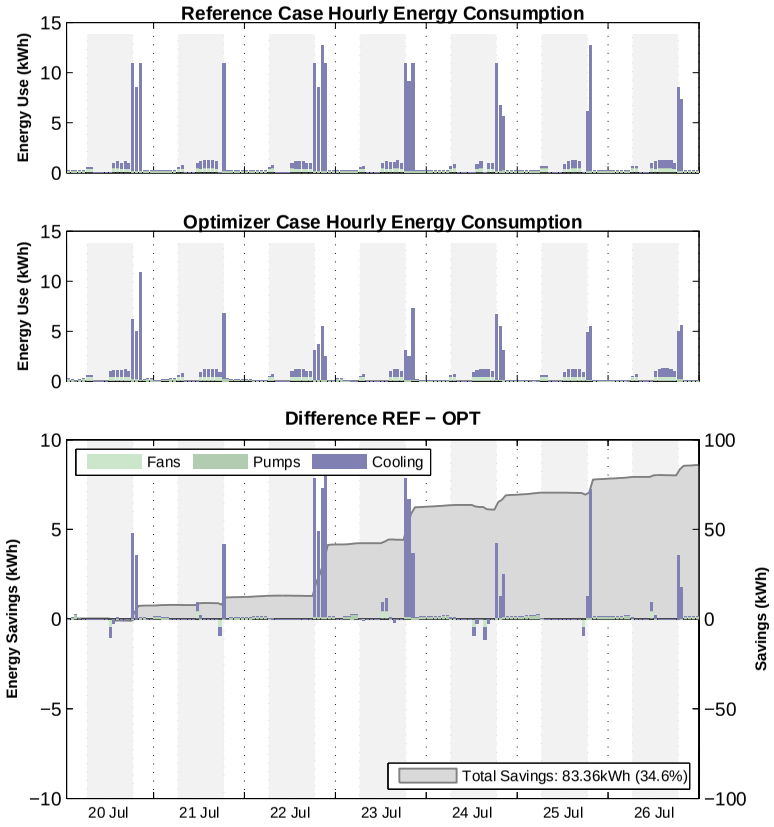


(b) Comfort

Figure D.31: Solution and comfort: MM4, cooling season, ASHRAE 55 adaptive comfort penalty



(a) Base vs. optimal



(b) Reference vs. optimal

Figure D.32: Savings: MM4, cooling season, ASHRAE 55 adaptive comfort penalty

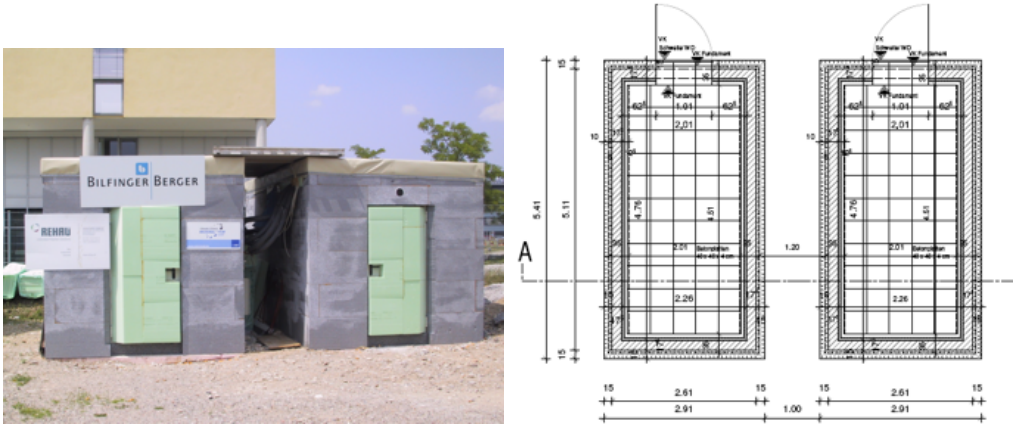
Appendix E

Fraunhofer ISE Experimental Setup Details

E.1 Facility and Equipment

Colleagues at the Fraunhofer Institute for Solar Energy Systems (ISE) in Freiburg, Germany made available two fully controllable and heavily instrumented test cells. These "garages" have a useful floor area of 10 m^2 (a common single office size) and are primarily used to test control strategies for thermally activated building structures (TABS). They are heavily insulated concrete boxes, with internal gains simulators (Figure E.2) and wall transmission gains simulators and are connected to a water heating/cooling apparatus that can provide precisely controlled process water for either heating or cooling the slabs. They also possess fans which can be used to simulate night ventilation cooling schemes at air change rates of about 4 ACH (Figure E.2). An exterior image and floor plan are provided in Figure E.1, with basic construction properties displayed in Table E.1.

A variety of measurements were possible within the cell using installed and calibrated equipment. All sensors were connected to networked data acquisition cards. Temperature measurements via thermocouples are available for room air (at three heights), hydronic loops (supply and return), ceiling surface (at multiple points on a grid), and at intervals within the concrete. Globe/operative temperature measurements were available via a globe thermometer positioned in the center of the room. Flow measurements were provided by a magnetic-induction flow sensor. Power measurements for either the internal or transmission gains simulators were provided by digital multimeters (not networked). Weather readings for the time period—including global horizontal irradiation,



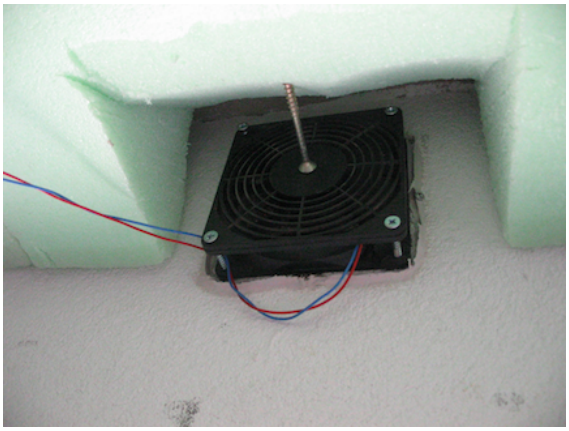
(a) Test garages (b) Floor plan

Figure E.1: TABS test cells at Fraunhofer ISE. Photos courtesy Fraunhofer ISE.

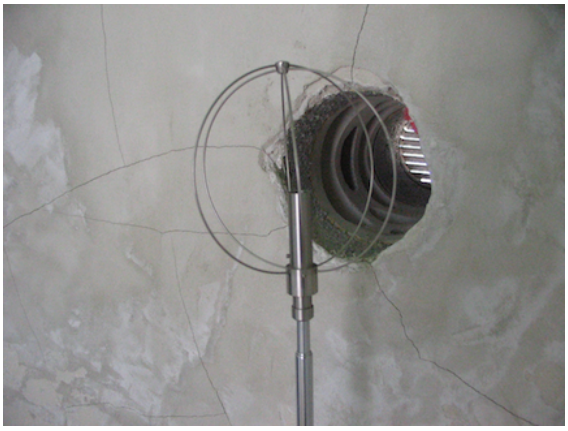
Table E.1: Test Cell Material Thermal Properties

	Layer	Material	d (cm)	λ (W/m-K)	ρ (kg/m^3)	c_p (J/kg-K)
Walls	1	Foam insulation	15	0.032	17	900
	2	Concrete block	17.5	0.8	1400	900
	3	Foam insulation	5.5	0.032	17	900
	4	Screed	0.2	0.35	1200	1000
Roof	1	Foam insulation	16	0.038	33	900
	2	Mid-weight concrete	12	1.8	2200	1000
	-	TABS	-	-	-	-
	3	Mid-weight concrete	6	1.8	2200	1000
Floor	1	Mid-weight concrete	5	1.8	2200	1000
	2	Foam insulation	16	0.038	33	900
	3	Chipboard	0.4	0.13	800	1000

(a) Load simulator



(b) Ventilation fans



(c) Digital anemometer

Figure E.2: Assorted TABS experimental equipment.

dry bulb temperature, and relative humidity—were provided through a separate weather station located in the vicinity of the test building at the ISE. Table E.2 provides descriptions and precision values for select sensors.

The data acquisition system also functioned as the automation system for the experiments and was capable of executing simple scripts. A simple script used in the experiment to calculate the Oleson supply water temperature reset schedule is provided below. Control actions can be made when the module becomes active (i.e. when the logical variable `TABS_ON` becomes true), is deactivated, or while the module is active.

```
Module calc_Tswt (TABS_ON)
{
    OnActivation
    { %Do nothing
    }

    ActiveOutputs
    {
        AT = OATemp;
        targ_temp_1 = 0.35*(18 - AT) + 18;
        targ_temp_a = if( targ_temp_1 < 16.0, 16.0, targ_temp_1);
    }
    OnDeactivation
    { %Do nothing
    }
}
```

Table E.2: Select Test Cell Measurement Equipment

Measurement Variable	Sensor (Type [Manufacturer])	Precision
Air temperature	PT100, 4-lead thermocouple [Fa. S + S Regeltechnik]	± 0.2K
Operative/globe temperature	PT100 4-lead globe thermometer [Fa. S + S Regeltechnik]	± 0.2K
Relative humidity	Capacitive humidity sensor [Fa. Heinz Messtechnik]	± 3% RH in range of 25 - 95%
Concrete core temperature	PT100, 4-lead thermocouple [Fa. S + S Regeltechnik]	± 0.1K
Water temperature	PT100, 4-lead, direct contact thermocouple [Fa. S + S Regeltechnik]	± 0.1K
Volume flow	Magnetic-inductive flow sensor [Fa. Krohne]	± 1.3% in 100 l/h flow range
Electrical power	True power meter [Fa. Saia-Burgess]	IEC 62053-21 precision class 1

E.2 Uncertainty Analysis

Since heat flux measurements form the main basis for comparing the energy performance of the TABS test cell versus offline MPC optimizations, it was necessary to conduct an error propagation analysis for heat flux calculations. Overall, error was found to be small (less than ± 2W). The cooling capacity of the TABS circuit, $\dot{Q}$, is expressed as a simple $\dot{m}c_p(T_o - T_i)$ heat flux calculation. Using Kline-McClintock methods for error propagation, the total error is expressed as the sum in quadrature of its three error components, namely:

$$w_{\dot{Q}} = \left[\sum_{i=1}^{3} \left(w_{x_i} \frac{\partial \dot{Q}}{\partial x_i} \right)^2 \right]^{1/2} \tag{E.1}$$

The partial derivatives for the various variables are provided below:

$$\frac{\partial \dot{Q}}{\partial T_o} = \dot{m}c_p \tag{E.2}$$

This is the same for T_i, except with opposite sign.

$$\frac{\partial \dot{Q}}{\partial \dot{m}} = c_p(T_o - T_i) \tag{E.3}$$

Given the high accuracy of the temperature sensors, heat flux measurements are fairly precise, with the greatest error contributions coming from the magnetic-induction flow sensor.