

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Using Building Simulation and Optimization to Calculate Lookup Tables for Control

Permalink

<https://escholarship.org/uc/item/60h9c8gs>

Author

Coffey, Brian

Publication Date

2011

Peer reviewed|Thesis/dissertation

Using Building Simulation and Optimization to Calculate
Lookup Tables for Control

by

Brian Coffey

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Doctor of Philosophy

in

Architecture

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Gail Brager, Chair

Professor Edward Arens

Professor Francesco Borrelli

Professor Philip Haves

Fall 2011

Abstract

Using Building Simulation and Optimization to Calculate
Lookup Tables for Control

by

Brian Coffey

Doctor of Philosophy in Architecture

University of California, Berkeley

Professor Gail Brager, Chair

There is a growing demand for more energy efficient buildings. Integrated systems with more intelligent controls are an important part of meeting this demand. Model predictive control (MPC) is an established control technique in other fields and holds promise for improved supervisory control in buildings. It has been receiving increasing attention in buildings research but has yet to find its way into common practice. This is due, at least in part, to a mismatch between the tools and techniques used in most MPC development and the tools, skills and processes commonly found in building design and operation. This dissertation investigates an approach to optimization-based control that uses common building simulation tools and could fit more readily into building design and operation practices. Instead of solving optimization problems in real-time to determine control set-points given current states and predicted disturbances, the optimal set-points are pre-computed offline over a grid of possible conditions and the resulting lookup table is used with linear interpolation for control. The feasibility and range of applicability of this approach are evaluated, including analyses of the performance impacts of grid spacing and techniques for problem dimensionality reduction. Three abstract case studies and two detailed case studies are presented. The approach is found to be feasible for supervisory control problems that can be effectively simplified to functions of roughly 5-6 conditions variables, and the case studies show good performance relative to online MPC. The benefits for ease of implementation are significant, but the most useful aspect is likely the feedback it can provide to the design process.

Dedication

To Philip Haves for the inspiration, to Corinne Benedek for the patience,
and to both of them for their support.

Acknowledgements

The research described herein was made possible by fellowships from the National Science and Engineering Research Council of Canada and the American Society of Heating Refrigeration and Air conditioning Engineers, and through research projects at the Lawrence Berkeley National Laboratory. In addition to all of the dissertation committee members, the author would like to thank Michael Wetter, Gregor Henze, Peter May-Ostendorp, Meli Stylianou, Edward Kutrowski, Edward Morofsky and Eleanor Lee for their assistance and feedback.

Contents

1	Background	1
1.1	Introduction	1
1.2	Context	2
1.2.1	Trends in high-performance building design	2
1.2.2	State of the art in building control systems	2
1.2.3	Existing design processes	3
1.2.4	Existing building energy simulation tools	4
1.2.5	Trends in building simulation research and development	6
1.2.6	Summary	9
1.3	Control optimization with building simulation tools	10
1.3.1	Model Predictive Control (MPC)	10
1.3.2	Offline approximation to MPC	12
1.4	Statement of the Problem	13
2	Objectives and Overview	14
2.1	Essential concepts and terms	14
2.2	Research objectives	15
2.3	Dissertation overview	16
3	Methods	17
3.1	Approaches: Rule-based control, online model predictive control, and lookup control with offline optimization	17
3.2	Range of computational feasibility	18
3.3	Conditions parametrization	20
3.4	Problem decomposition	23
3.5	Solution methods for the conditions grid	26
3.6	Lookup table visualization	27
3.7	Simulated implementation	29
3.8	Physical implementation	30
3.9	Open-source software	31
3.10	Case studies	32
4	Abstract Case Study #1: Active Facade, Grid Spacing	33
4.1	Case study: Solar shading and natural ventilation	33
4.1.1	Case description	33
4.1.2	Model description	34

4.1.3	Optimization configuration	36
4.1.4	Lookup tables results	36
4.1.5	Annual simulations	39
4.2	Analysis: Effects of grid spacing on control performance	49
4.2.1	Controller performance relative to grid spacing	49
4.3	Discussion	51
4.3.1	Process notes	52
4.3.2	For future consideration	52
5	Abstract Case Study #2:	
	Slab Pre-Cooling, Conditions Parametrization	54
5.1	Case study	54
5.1.1	Case description	54
5.1.2	Model description	54
5.1.3	Conditions parametrization: disturbances	55
5.1.4	Conditions parametrization: initial states	61
5.1.5	Optimization configuration	63
5.1.6	Lookup table results	63
5.1.7	Annual simulations	67
5.2	Analysis: Effects of conditions parametrization on performance	71
5.2.1	Full online MPC configurations for comparison	71
5.2.2	Results	71
5.3	Discussion	75
6	Abstract Case Study #3:	
	Cogeneration Dispatch, Problem Decomposition	76
6.1	Case study	76
6.1.1	Case description	76
6.1.2	Problem decomposition	77
6.1.3	Subproblem model description	77
6.1.4	Optimization configuration	79
6.1.5	Subproblem lookup table results	79
6.1.6	Annual simulations	82
6.2	Discussion	85
6.2.1	Irregular grid and/or heuristics to deal with control switch points	85
6.2.2	Computational infeasibility of a full online solution	86

7	Detailed Case Study #1:	
	Extension of the UC Merced chilled water system MPC study	87
7.1	Case description	87
7.1.1	Background	87
7.1.2	Description of campus chilled water system	87
7.1.3	Problem definition	88
7.1.4	System component models	89
7.2	Problem decomposition	90
7.2.1	Subproblem definition	91
7.2.2	Main problem definition	92
7.3	Conditions parametrization	92
7.4	Subproblem: Optimization configuration	95
7.4.1	Grid definition	95
7.4.2	Computation methods	95
7.5	Subproblem: Lookup table results	95
7.6	Main problem: Online solution using the subproblem lookup table . .	101
7.7	Annual simulations	101
7.8	Packaged software for use by operators	108
7.9	Discussion	108
7.9.1	Lessons learned in previous iterations	108
7.9.2	Possible extensions in future research	109
8	Detailed Case Study #2:	
	Integrated Facade and HVAC Control	111
8.1	Case description	111
8.1.1	Background	111
8.1.2	Problem definition	111
8.2	Model description	112
8.2.1	Assumptions and overview	114
8.2.2	Model configuration for control optimization	114
8.3	Conditions parametrization	115
8.4	Optimization configuration	118
8.5	Lookup table results	118
8.5.1	Interior blind case	118
8.5.2	Exterior blind case	121
8.6	Annual simulations	123
8.6.1	Annual simulation configuration	123
8.6.2	Annual simulation results: Interior blind case	124

8.6.3	Annual simulation results: Exterior blind case	127
8.7	Discussion	128
9	Discussion	129
9.1	Lessons learned from the case studies	129
9.2	Evaluation of the approach	130
9.2.1	Range of feasibility and suitability	130
9.2.2	Benefits	130
9.2.3	Challenges	132
9.2.4	Prospects for market uptake	132
9.3	Areas to consider in future research	132
10	Conclusions	134
	References	136
A	Appendix: Campus Load and Return Temperature Models	141
A.1	Overview	141
A.2	Campus load model	142
A.2.1	Model description	142
A.2.2	Model calibration	145
A.3	Campus return temperature model	146
A.3.1	Model description	146
A.3.2	Model calibration	150
A.3.3	Model predictions with warmer supply water temperatures . .	151
A.3.4	Experimental results	152
A.3.5	Discussion	153

List of Figures

1	Receding horizon strategy	10
2	Controller configurations: rule-based control versus online MPC . . .	17
3	Controller configuration for lookup control using offline-optimization	18
4	Range of feasibility of lookup table approach	19
5	Example parametrizations	21
6	Controller configuration for lookup control using offline-optimization	23
7	An example lookup table slice shown in two different ways	28
8	Example lookup table scatter plot	29
9	Simulated implementation with BCVTB	30
10	Example human-in-the-loop interface for lookup table controller . . .	31
11	Solar shading and natural ventilation case study illustration	33
12	Massless zone model	34
13	Optimal shading and natural ventilation levels	37
14	Optimal shading levels versus direct and diffuse solar gain	38
15	Weather disturbances used in the annual simulation	39
16	Weekly schedule for waste heat from people and plugs	40
17	Control decisions, lookup table control versus base case, annual . . .	41
18	Control decisions, March 1-14	42
19	Control decisions versus ambient temperature	43
20	Hourly energy consumption, lookup table control vs base case, annual	44
21	Energy consumption outputs, March 1-14	45
22	Total energy consumption outputs, lookup table control versus base case	46
23	Lookup control energy savings over base case	46
24	Lookup control energy minus online MPC energy	47
25	Total energy consumption outputs	48
26	Lookup control energy minus online MPC energy	50
27	Energy savings vs direct and diffuse grid spacing	51
28	An earlier iteration: optimal shading and natural ventilation levels .	53
29	An earlier iteration: optimal total energy	53
30	Chilled slab case study illustration	54
31	Building layout image from the RadLoTempHydrCoolTower.idf file .	55
32	Ambient temperatures - daily maximum and range, SF TMY data .	56
33	Ambient temperatures parametrization, based on SF TMY data . . .	56
34	Global horizontal solar parametrization, based on SF TMY data . .	57
35	Direct normal solar parametrization, based on SF TMY data	58

36	Diffuse horizontal solar parametrization, based on SF TMY data . . .	59
37	Relative humidity parametrization, based on SF TMY data	60
38	Floor temperature as a function of pre-charge length and heat burst .	62
39	Zone temp minus floor temp, as a function of heat burst and pre-charge length	62
40	Optimal charge length	64
41	Surface plots of optimal start time and charging length	65
42	Optimal average PPD	66
43	Disturbances, annual	67
44	Control decisions versus next day max ambient temperature	67
45	Control values, summer - top, June only - bottom	68
46	State values, summer - top, June only - bottom	69
47	PPD values, summer	70
48	Control performance versus next day max ambient temperature . . .	70
49	Control values with MPC, summer	72
50	State values with MPC, summer	73
51	PPD values with MPC, summer	74
52	Fuels to loads diagram of the cogeneration plant considered in this study	76
53	TRNSYS model of cogeneration system	78
54	Optimal generator level vs elec price and elec demand	80
55	Optimal generator heating level vs elec price and heating demand . .	80
56	Optimal generator and chiller levels vs elec price and cooling demand	81
57	Loads and prices for annual simulation	82
58	Optimal monthly thresholds	83
59	Annual hourly electric grid purchases	84
60	Chilled water storage system	88
61	Overnight T_{wb} minimum and range over a typical year	93
62	Normalized overnight T_{wb} values	93
63	Overnight campus return temperatures and flow rates	94
64	Annual 10pm temperatures of the storage tank hot node	94
65	Scatter plot of optimal start time versus charge length	96
66	Detailed slices of optimal start time versus charge length and tank temp	97
67	Scatter plot of optimal condenser temp versus minium wet bulb temp	98
68	Detailed slices of optimal condenser temp versus T_{wbMin} and T_{a10pm} .	99
69	Detailed slices of optimal plant COP versus T_{wbMin} and T_{a10pm} . . .	100
70	Disturbances, Merced	102

71	Daily control decisions, Merced	103
72	System states, Merced	104
73	Daily charge length, Merced	104
74	Charge length versus disturbances and states, Merced	105
75	Charge length versus disturbances and states, averaged values, Merced	106
76	Monthly average wet bulb temperature and tank temperature during charging, Merced	107
77	Monthly plant COP and energy consumption, Merced	107
78	User interface for approximated MPC implementation	108
79	Optimal start time vs charge amount, assuming constant return temp	109
80	Shading and HVAC control variables of interest	111
81	EnergyPlus model for case study	112
82	Annual energy basecase, end-use breakdowns	113
83	Screenshot of hourly weather parametrization, horizontal infrared ra- diation	116
84	Horizontal infrared radiation versus ambient temperature	116
85	Screenshot of hourly weather parametrization, wind direction	117
86	Interior blind case: Optimal supply temperature versus ambient temp.	119
87	Interior blind case: Optimal blind angle versus ambient temperature	120
88	Interior blind case: Optimal blind angle versus direct and diffuse solar values	120
89	Exterior blind case: Optimal supply temperature versus ambient tem- perature	121
90	Exterior blind case: Optimal blind angle versus ambient temperature	122
91	Exterior blind case: Optimal blind angle versus direct and diffuse solar values	122
92	Annual weather variables	123
93	Annual control outputs	124
94	March control outputs	125
95	Hourly control outputs vs ambient temperature	125
96	Annual energy savings, end-use breakdowns	126
97	Hourly energy consumption vs ambient temperature	127
98	Campus load and return temperature models	141
99	Building load model in Modelica, showing the two main sub-components	142
100	Thermal resistance and capacitance model	144
101	Campus cooling load, May 16 - Jun 2, 2009	145
102	Block diagram of the heat exchange model	146
103	Measured return temperatures for (a) C&O Building, (b) entire campus	147

104	(a) UA as a function of m_w and m_a (Equation 38), (b) Performance of model with the variable UA , Equations 34-39	148
105	(a) UA varied by Equations 38-42, (b) Model performance, Equations 34-42	150
106	(a) Maintaining a LMTD, (b) a new convergence point with a higher T_{wi}	151
107	Predicted return temperature values with a higher supply temperature	152
108	Measured effect of increased T_{wi} on (a) C&O Building, (b) entire campus	153

List of Tables

1	Model parameters	34
2	Disturbances grid	36
3	Energy savings over base case for full MPC test periods	47
4	Disturbances grids considered	49
5	Conditions grid for the massive slab cooling case study	63
6	Results for the massive slab cooling case study: basecase vs lookup	71
7	Results for the massive slab cooling case study: lookup vs MPC	74
8	Key model parameters	78
9	Disturbances grid	79
10	Annual energy costs comparisons	84
11	Grid of conditions for subproblem solution lookup table	95
12	Some key model parameter values	112
13	Base case annual energy consumption by end use, W/m^2	113
14	Parameter correlations	117
15	Conditions grid	118
16	Interior blind case: Annual energy consumption, W/m^2	126
17	Exterior blind case: Annual energy consumption, W/m^2	128
18	Gains component parameters	143
19	Campus load model calibrated parameter values	145

1 Background

1.1 Introduction

Building designers and operators are currently being challenged by two opposing trends. The 20th century saw the introduction of mechanical cooling, large-scale window glass manufacturing and cheap energy sources, and building occupants have come to expect tighter temperature control, more daylighting and more expansive views. However, by the end of the 20th century it also became clear that fossil fuel consumption would eventually be forced to decline because of the economics of finite resources, and would likely have to decrease much more quickly than that if the risk of disruptive anthropogenic climate change is to be kept low. In the United States, the residential and commercial sectors combine to account for 40% of national energy consumption (USDOE, 2011a). Much of that is not directly attributable to the building itself as much as it is to energy uses that occur within the building, such as cooking and plug loads, but roughly half of it is from space heating (27% in residential, 14% in commercial), space cooling and ventilation (16% in residential, 19% in commercial), and lighting (10% in residential, 17% in commercial) (USDOE, 2011a). This very significant portion of national energy consumption attributable to buildings is also considered to be one of the least expensive and fastest areas of the economy in which to improve energy efficiency (Metz et al., 2007). Building designers and operators are thus being asked to produce much lower energy consumption rates while still meeting contemporary expectations for comfort and amenities.

There is much to be learned from low-energy buildings designed before mechanical cooling, float glass and cheap energy, but these designs cannot be simply replicated now in pursuit of decreased energy consumption. Contemporary comfort expectations and the spirit of architectural innovation both demand an approach that learns from historical low-energy precedents while devising new forms of energy efficient design and operation. The digital revolution has also pressed this agenda in two significant ways: (1) waste heat from plug loads makes it even more untenable to simply replicate pre-air-conditioning designs, and (2) information and controls technologies promise to change the building stock just as they have so much else in contemporary society. This dissertation is a small part of a growing body of research and practice that looks to harness these new-found computational powers to both decrease energy consumption and improve occupant comfort.

1.2 Context

The research presented herein considers a novel approach to creating near-optimal control strategies for innovative low-energy buildings. The approach grows out of the context of current design and operations practice, and if it is to be successful then it must also fit well within this context. Significant aspects of current practice, from the perspective of new controls development, are thus discussed below, before considering research precedents in control optimization for buildings.

1.2.1 Trends in high-performance building design

Design strategies that decrease solar, lighting and envelope loads are being pursued not just for their own incremental energy savings, but also because low enough thermal loads can allow for the use of more efficient, but lower capacity, HVAC systems, such as radiant slabs and natural ventilation. These lower capacity HVAC systems often operate at temperatures closer to ambient, which increases the possibility of using site and program peculiarities, such as local heat sources and sinks, to further increase building efficiency. Thermal energy storage is becoming more common, both for increased efficiency and for load shifting to avoid peak energy charges. In addition, some new designs are also integrating previously independent systems, such as HVAC and solar shading, in order to draw out even more efficiency improvements.

Intelligent operation of these more complex and integrated systems is often crucial to their success. However, designing good controls for such systems is challenging, and is rarely considered properly during the system design process.

1.2.2 State of the art in building control systems

The rate of new developments in the fields of electronics, controls and computer technology over the past four decades has been much faster than the turnover rate of the building stock, or even that of most buildings' HVAC equipment. Because of this, and perhaps also because of how relatively small a concern energy expenditure is for most firms, existing buildings sport a patchwork quilt of pneumatic, analog, digital and wireless systems implementing a wide array of control and interfacing strategies, ranging from archaic to reasonably advanced. Simple thermostats are used in most commercial buildings, with more complex central energy management systems installed in only about one third of the commercial buildings stock, more often in larger buildings than in smaller ones (Brambley et al., 2005). These central energy management systems are used by building operators primarily to identify potential operations problems (e.g. zones not meeting set-point), and only occasionally

to monitor or analyze energy consumption. Their user interfaces tend to be rudimentary, and closed proprietary systems make it difficult for people other than the control system vendors to make any major adjustments. Lighting control systems are becoming more common, allowing for daylighting or occupancy-based controls, but, when they are used, they are almost always a completely separate control system from the HVAC control system, with no communication between them.

From the perspective of energy efficiency, the interest in building controls is generally not for tighter set-point tracking or greater stability (as is often the case in other controls fields), but rather for devising ways to reduce energy use while still maintaining occupant comfort. Various supervisory control strategies have been developed for use in building energy management systems, such as night temperature set-backs and ‘economizer’ controls for outdoor air flow rates. These control strategies are generally describable by decision trees - based on a collection of current conditions (e.g. time of day, outdoor air temperature), decision rules are used to determine the set points for each system component. So when building system designers and controls designers consider new control strategies for new systems, it is through this lens that they tend to look. But as systems become more complex, such decision trees become harder to define, harder to manage, and harder to optimize.

1.2.3 Existing design processes

Building design is usually a very time-consuming process, but controls design tends to get very little of this time. Standardized control strategies are used for common HVAC and lighting systems and lighting control systems. Controls vendors usually have their own standard rule sets, the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) publishes standard control sequences, and some HVAC design firms may develop and specify their own for particular system types that they commonly use. More innovative building systems often require custom-made supervisory control rules, which usually rely very heavily on the engineering judgement of the designers. As an example of custom-made control rules, consider the use of ‘red light green light’ systems for mixed mode systems (signaling to occupants when they can open their window or not), which is too new a technology to have been standardized. Ackerly and Brager (2011) found that a wide variety of control strategies have been used in practice in such buildings, even though, in the cases considered, the building systems themselves were very similar. This suggests that relying entirely on engineering judgement may not be the best way to approach the design of controls strategies for innovative systems.

Shallow consideration of control strategies during system design makes it diffi-

cult to evaluate the energy savings potential of design variants whose performance depends on intelligent controls. It also makes it difficult to ensure that the designed system operates as intended, since the controls implementation is often left to someone other than the system designer.

The use of building energy simulation tools in building design is worth noting. The tools are generally intended to provide the designers with feedback on the potential operational energy consumption of building designs, to help guide the design process towards more efficient buildings. However, most buildings never get simulated before being built, and those that do are usually just simulated for comparison against a pre-defined base case, for the purposes of code compliance (e.g. Title 24 in California) or green-building certification (e.g. LEED). Some more progressive design firms will use building simulation to inform design decisions. The simulation tools used for this are often the same ones that will be used later for compliance / certification modeling, despite sometimes being limited by interfaces better suited to compliance modeling than to systematic analysis of design trade-offs. However, this modeling process by system designers or consultants is usually the point when the most detailed energy efficiency analysis occurs in a building's life cycle. At present, this process rarely involves active development of control strategies, and the models, with their embedded knowledge of design intent and expected system behavior, rarely get used beyond the point of building occupation.

1.2.4 Existing building energy simulation tools

History of building energy simulation

Dynamic building simulation got its start in the 1970s and 1980s, spurred on by the energy crisis and the introduction of the personal computer. The first generation of programs were written primarily in FORTRAN, and include the whole-building simulation program DOE-2 (written primarily at Lawrence Berkeley National Lab (LBNL)), along with BLAST (by the US Department of Defense) and TRNSYS (University of Wisconsin, Madison). DOE-2, primarily in its eQuest / DOE-2.2 form, has become the most commonly used building simulation program in North America. TRNSYS has continued to expand from its initial focus on solar water heating systems to become a modular program with an extensive library of building system components and load models, but it requires software purchase (unlike many of the other tools) and sees more use in Europe and Canada than it does in the United States. BLAST has become essentially obsolete, but some of its principles and code have been absorbed into EnergyPlus, which started its development in the 1990s after DOE-2.1E development ended. EnergyPlus is being developed by the

US Department of Energy through a number of its national labs (including LBNL and the National Renewable Energy Lab (NREL)) and with a number of partners (including the Florida Solar Energy Center).

It is worth noting that all of these programs are still FORTRAN based, and that many of the core algorithms have not changed since the 1970s, despite the vast changes in programming practices since then. Recognizing the benefits of a more modular approach to both code development and tool use, there were a number of attempts in the 1990s to develop more modular simulation programs in languages like C++. SPARK was one such program, developed at LBNL, which allows users to define objects with ports, rather than pre-defined inputs and outputs (as is the case with TRNSYS), and has a solver that determines what the inputs and outputs are within a given case and devises an efficient method to parse and solve the problem. As Augenbroe (2003) has argued, these attempts at object-oriented programs in the 1990s were founded on good ideas, but got caught up in the many challenges associated with developing the modular structures and solvers, and they were overwhelmed by the very large challenge of having to re-write the algorithms of the existing programs, without the funding that the existing programs received. Another attempt at developing a more modular, equation-based, object-oriented building simulation environment is under way, this time taking advantage of underlying structural development that has already been completed (an approach that Augenbroe endorses in his analysis) by developing a buildings library for Modelica (Wetter, 2009b).

This US-centric brief history has not yet noted whole building tools like ESP-r and TAS that are more commonly used in Europe. The commercial software IES-Virtual Environment is important to include, as it is becoming common in Europe and Australia and is gaining popularity in the US. Ecotect is also worth noting (it is very popular among architects, but more for its climate and solar analysis and graphics than for its energy simulation), as are the variety of CFD tools available, the bulk airflow tools like CONTAM, and Radiance and DaySim for light simulation. The US Department of Energy maintains a list of publicly-available (free or commercial) building simulation tools and other tools closely related to building simulation, currently with over 400 tools listed (USDOE, 2011b).

Characteristics of existing building energy simulation tools

The main use case in the original development of most of the major building simulation tools is to aid in design decisions regarding building orientation, massing, HVAC system selection, glazing areas and so forth. Related to this is the more common use for compliance checking for codes and certifications. In both of these

cases, annual simulations with timesteps on the order of 15 minutes to one hour are the norm. Initial state values rarely have a noticeable impact on the annual energy results, and most of the simulation tools have hard-coded ‘warm-up period’ requirements (to get the initial states roughly correct), rather than allowing the user to enter initial state values themselves. Models tend to be of an entire building, with possibly hundreds of thermal zones, rather than of a smaller part of the building. Annual simulation run-times of minutes or even hours are neither uncommon nor particularly problematic for these use cases.

Existing software that extends the use of building simulation tools

Although less common in practice than desired by most researchers and advanced practitioners, parametric analysis with these tools is occasionally carried out in the early design phases to study the sensitivity of the building’s energy consumption to particular design parameters. Numerical optimization is even less common in practice, although it does happen in some consulting firms, and is relatively common in building research. GenOpt (Wetter, 2009a) is an open-source java program designed to facilitate optimization with any building simulation tool that uses human-readable text files as inputs and outputs (as most of the common tools do). It has an extensible library of optimizations algorithms, so the user may choose an existing algorithm or write their own. It is well established as an optimization tool for building simulation (and a more complete user interface would likely increase its use in practice). It is used extensively in the research presented herein.

It is often the case that multiple simulation tools are needed to investigate particular design or research questions, and sometimes these need to be run alongside each other to pass information back and forth at each timestep. The Building Control Virtual Test Bed (BCVTB, (Wetter and Haves, 2008)) is an extension of the Ptolemy II software (of Electrical Engineering and Science, 2011) that facilitates the run-time coupling of various building simulation tools and controls software platforms. It can be used for co-simulation of whole-building energy simulation tools with CFD or detailed daylighting analysis, and also can be used for the coupling of energy simulation tools with controls software or hardware. It is a relatively new piece of software that is quickly becoming common in research use and may eventually make its way into practice. It is also used extensively herein.

1.2.5 Trends in building simulation research and development

Building simulation tool development is currently being shaped in at least four significant ways by the trends in building design and the demands for more efficient

buildings. The first is the demand from users to simulate more complex and efficient HVAC systems, such as chilled beams, variable-refrigerant-flow systems and radiant slab cooling. The second is to make simulation tools more user-friendly, as building codes and labeling schemes are vastly increasing the demand for building simulation for compliance purposes. The third is a growing use of Building Information Models (BIM) in the building design and construction industries in general, and building simulation tools are being pushed to properly integrate with such models. All three of these are of interest in this context, because they mean that at least some of the existing building simulation tools are likely to be further developed to become more information-rich, less taxing on the model developer, potentially more flexible, and more widely-used. In general, they are likely to become ever more prominent in building design.

The fourth notable trend in building simulation development is towards the operations phase of the building life cycle. Although the vast majority of energy analysis occurs during building design, it is during building operation that the energy is actually consumed, and researchers and some practitioners are beginning to find ways of re-using building models from the design phase to improve operations. Three main operations-phase uses of simulation are outlined below: benchmarking and feedback, fault detection and diagnosis, and retrofit analysis.

Benchmarking and occupant / operator feedback

There has been much recent growth in the field of real-time building data visualization. Just making the data available to building occupants and operators has been found to have a significant impact on their behavior, but building ‘dashboard’ developers are finding that it is much more effective if the current energy use data can be related to a reference. Some dashboards are using other buildings on a campus or other floors in the same building as references, which can also foster a collegial competitiveness that has been shown to be an effective motivator for energy-saving behavior (Petersen et al., 2007). Other dashboards are using other benchmarks for comparison. For example, the Pulse Energy software (Pulse Energy, 2011) uses a sophisticated regression-based baseline model to calculate how the building would be expected to perform at the current hour given its historical performance under similar conditions (outdoor temperature, time of day, etc) - this can be used to flag potential problems, such as equipment not turning off overnight when it should be. A data visualization project at UC Merced (Apte et al., 2010) developed a calibrated lumped-parameter model of one of the buildings to use in similar ways. More detailed models can do more than provide a reference point or raise a flag when something

seems wrong - they could also suggest to the occupants or operators what the problem might be and/or how to fix it, and thus overlap with the fault detection and diagnosis approach described below.

In addition to these real-time data visualization and comparison tools, there have been a number of recent developments aimed at comparing buildings' annual energy consumption to benchmarks. Such a strategy can be considered on two levels. At the building stock or portfolio level, the benchmarking process is useful in identifying problem buildings to look at in more detail. And at the level of the individual building, a more detailed benchmarking process might suggest particular subsystems or components that may have problems. These more detailed benchmarks generally involve some sort of energy simulation models. The Facility Energy Decision System (FEDS) was developed by the Pacific Northwest National Lab with this intent, and is used by various federal agencies (PNNL, 2011). The Energy Profile Tool (EnerSys Analytics, 2011) is a similar piece of software which uses a database of results from a wide variety of parametric simulations on template DOE-2.1 models. The LBNL-Target project (Haves et al., 2008) is an interesting example in that it was devised to operate at both the portfolio level and the individual building level, as well as to provide guidance on retrofit possibilities for the existing stores and on design changes for the next generation. The project took advantage of the relative standardization of the big box store design across the country, and developed a high-fidelity EnergyPlus model of a prototype store design that could be easily modified to match the small customizations in insulation levels and HVAC equipment that occurred in the various climate regions. With relative ease (although, it turns out, with perhaps not enough ease), the EnergyPlus model could then be matched to any store in the portfolio of a recent vintage (the older stores were of different designs). Annual energy use discrepancies between the simulated and actual would be used at the portfolio level to identify problem stores. And the details of such differences could be used at the store level as indicators of possible problems.

Fault detection and diagnosis

Fault detection and diagnosis may be done by expert rules systems, such as the Diagnostics Agent for Building Operators (DABO) (NRCan-Varennes, 2011), which uses a set of pre-programmed rules to analyze the data from an EMCS, identifies problems (e.g. when equipment is operating outside of its expected efficiency range) and suggests solutions. But model-based fault detection and diagnosis is perhaps more easily generalized. A model-based version of fault detection and diagnosis is being developed at LBNL (Xu et al., 2005). Measured outputs from actual compo-

nents are compared to the outputs of modeled components given the same inputs. Flags and suggestions result when the actual behavior deviates from the modeled behavior to some extent deemed significant. The model-based version has a number of other benefits: it can possibly link well with the other uses of building simulation in operations discussed herein, and could also link with HVAC visualization and training tools - when a problem occurs, the component model can be used within a visualization environment like the Flash-based interface for LearnHVAC (SuPerB and LBNL, 2011) to demonstrate to the operator what the problem is and how it can be fixed.

Retrofit analysis

Retrofit analysis is perhaps the most common current use of building simulation in the operations phase, and it is the most akin to simulation use in the design phase. Retrofit studies with simulation are usually carried out as part of a larger energy auditing process, where an auditor may use whole building models from the design phase (potentially re-calibrated with measured data) to inform the analysis.

1.2.6 Summary

Driven by demands for higher building efficiency while still meeting contemporary comfort standards, building designs are changing such that they more often integrate various previously independent subsystems (e.g. facades and HVAC), take advantage of efficiency opportunities afforded by the particularities of the site and program, and often incorporate thermal storage. The control of such systems is crucial to their success. Existing building control systems can handle new supervisory control strategies as long as they are simple to implement, ideally in the form of a decision tree or something similar, but they are difficult to extend much beyond their existing structures. Various existing building simulation tools may be used during the design phase, and this is when most of the energy analysis happens for a building. Some research projects and practical applications are exploring ways of using these models to improve operations through benchmarking, fault detection and diagnosis, and retrofit analysis.

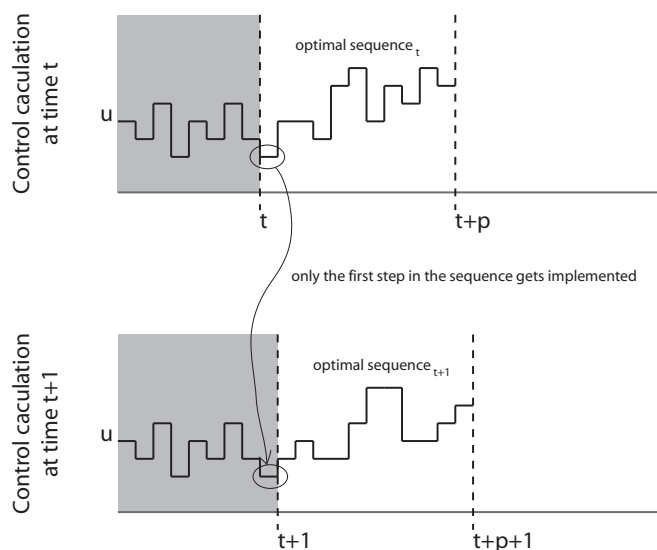
Methods to improve the controls of complex buildings by using building simulation models, while fitting in with current design practices and control systems, and integrating with other operations-phase uses of building simulation, could thus be valuable to the industry.

1.3 Control optimization with building simulation tools

1.3.1 Model Predictive Control (MPC)

Online MPC offers a way of approaching the complex control problems associated with many of the new high-performance building systems. Instead of trying to define the control logic explicitly, a building model and an optimization algorithm are used within the control system in real-time to calculate the best setpoint values given the current and predicted conditions. MPC is a repeated solution of a finite-time optimal control problem: at each controller time step, an optimal sequence of control values over a prediction horizon is calculated, only the first of which is implemented, and at the next controller time step the horizon shifts forward one step and the process is repeated, as shown in Figure 1, where u is the control output, p is the horizon length, the horizontal axis is time and the greyed out part is the past. (Note that a number of variants on this are also possible, such as implementing two or more time steps of inputs and performing the optimization less often, or having different control and prediction horizon lengths.)

Figure 1: Receding horizon strategy



MPC is well established in other fields. It was first used in the chemical process industry in the 1960s, and has seen increasing use since then. A survey by Qin and

Badgwell (2003) notes that MPC is used in more than 4,000 industrial applications. It is well suited to the control of non-linear systems with strong state constraints and relatively slow system dynamics, which is often found in chemical process control. It was a practically proven technique before it was studied theoretically, with mathematical investigations of stability and optimality criteria beginning essentially with the work of Mayne in the 1980s. Overviews of the field are available in Morari and Lee (1999) and Mayne et al. (2000).

There have also been a growing number of MPC studies for building systems, originating not in controls research but in building energy simulation research. The idea was referenced in the buildings engineering literature at least as early as 1988 (Kelly, 1988) as something worth considering for buildings, but it was not until the turn of the century that computation power and building simulation were at the levels required to test it. Mahdavi (2001) provided a succinct description of simulation-based control using standard simulation tools (but without prediction, since the systems he considered did not require it), and studied its application to solar shading (Mahdavi et al., 2005) and natural ventilation (Mahdavi and Proglhof, 2005). A more complete investigation of MPC for passive and active thermal storage systems has been carried out by Henze et al (Henze and Liu, 2005; Henze and Krarti, 2005). This work includes a test of a real-time implementation of model predictive control of a system with active and passive thermal storage, using a TRNSYS model of the system as the online model in the controller. This work also includes analyses of automated calibration to improve the models used in model predictive control (Henze et al., 2005), and a study of the effects of prediction inaccuracies (Henze et al., 2004). Henze is now leading a team of researchers on the investigation of model predictive control for buildings with natural ventilation and massive chilled slabs (see (Henze et al., 2007) for some of the preliminary work). Various other researchers have made contributions in this field, including the investigation of optimal start and stop times for slab charging (Kummert et al., 2005; Kummert and Andre, 2005; Clarke et al., 2002), and VAV control (Wang and Jin, 2000; Nassif et al., 2005a,b). Clarke et al. (2002) also worked with Honeywell to test the use of ESP-r for model predictive control. The paper by Clarke et al also compares MPC to fuzzy logic and neural network controllers, noting the various benefits of having a physics-based model in the controller, including the avoidance of an in-situ learning period. An open-source software framework for model predictive control using any text-based building simulation software, and using GenOpt to perform the online optimization, is presented in (Coffey, 2008) and (Coffey et al., 2010a). Some recent work has come from controls researchers from other fields turning their attention to buildings (e.g. Ma et al. (2010) and Oldewurtel et al. (2010)). Potential for energy savings,

demand reduction and performance improvement has been shown with a wide variety of systems, including chilled water storage, radiant slab pre-cooling and integrated HVAC and facade control. And as buildings become more complex the benefits of MPC are expected to become more pronounced.

However, MPC is currently far from common practice in building design and operation - it has yet to get beyond the stage of individual case study implementations by researchers. It is difficult to use standard building simulation tools for this because of their long run-times and the fact that many do not allow the user to explicitly specify initial state values, and the software used by most controls researchers is unfamiliar to most buildings researchers and practitioners. In addition, online optimization is difficult to implement with existing building control systems, and the fact that the control rules are implicit rather than explicit makes it difficult for system designers to integrate it into their design processes.

Because of the long run-times of common simulation-tools, researchers have been attempting either the use of long controller timesteps with standard building simulation models, or making custom lower-order models for particular buildings or systems. Some research groups are postulating that MPC with simplified models can be standardized enough to be used in many buildings. Others are suggesting that insights gained through online MPC implementations and annual simulation studies with particular buildings can lead to the development of simplified rules that can be applied to many buildings with configurations similar to the particular buildings studied. However, buildings designed to be highly responsive to weather and occupants and use low-exergy site-specific heat sources and sinks are inherently difficult to match with prototype buildings or to effectively model with standardized simple models. Indeed, one of the primary motivations for using MPC in buildings is that it provides a coherent method for near-optimal supervisory control of complex integrated systems for which good heuristic rules are difficult to define.

1.3.2 Offline approximation to MPC

For some types of MPC problems, multiparametric programming can be used to solve the problem explicitly, providing a set of control laws that fully cover the conditions space and that exactly replicate the control behaviour of online MPC (referred to as ‘explicit MPC’, see Bemporad et al. (2002)). However, multiparametric programming can only be used to solve certain types of MPC problems (e.g. linear or switched-linear models with linear or switched-linear objective functions), into which categories most of the challenging building control problems do not fit. And with the possible future exception of Modelica (Wetter, 2009b), this approach would not

be possible with any of the commonly used building simulation tools.

However, the idea of explicit MPC is very appealing for buildings applications, because it would be easier to implement in existing building control systems than online MPC, and being able to describe and visualize optimal control responses over the full conditions space would provide useful feedback during system design. Finding ways to approximate it using common building simulation tools could thus be worthwhile, even if it brings with it some performance penalties relative to online MPC. Current work by May-Ostendorp and Henze considers the approach of simulating online MPC over some or all of a representative weather year and then using statistical techniques to derive near-optimal control laws from the results. This could provide a useful way of getting these benefits.

A potential alternative, discussed herein, is to develop tools and techniques for using building simulation and optimization to compute a lookup table of optimal control setpoints for a grid of current and predicted disturbance values and initial state values. The lookup table is then used for online control of the real building system or in annual simulations, rather than running the optimization online at each timestep.

1.4 Statement of the Problem

There is a need for better methods for designing supervisory controls for innovative building systems, in order to capture more of the available energy savings and comfort improvements that these systems offer. The methods must fit well within current building design and operations practices in order to see market uptake. Ideally, the solution should be useable in design-phase annual simulations, for evaluation of energy savings and payback periods of different systems. Methods for offline approximation to MPC are promising in these regards, in particular the use of offline optimization over a grid of possible conditions to produce a lookup table that can be used with interpolation in real-time control. This latter approach, however, has not been studied in previous research in terms of its range of feasibility, techniques required to make it work, and its performance relative to MPC and heuristic rule-based controls.

2 Objectives and Overview

This dissertation investigates the feasibility and relative performance of an offline approximation to MPC using optimization over a grid of possible conditions (e.g. outdoor temperature, occupancy level) to produce a lookup table that is used with interpolation for real-time implementation. Methods for the approach are developed and demonstrated through a series of case studies, and these studies are extended to analyze various aspects of the approach.

2.1 Essential concepts and terms

Although methods are discussed in much greater detail in the next chapter, some essential concepts and terms are provided in overview here to act as a general guide to the reader, and in order to make comprehensible the research objectives delineated below.

The term ‘conditions’ is used throughout to refer to both (1) current and predicted disturbances (e.g. outdoor temperature, outdoor horizontal diffuse radiation, plug load heat gains) and (2) initial states (e.g. current indoor air temperature, current storage tank charge level). At any control timestep, the controller is responding to conditions and outputting control signals (e.g. shading position setpoint, tank charging start time).

As described in the previous chapter, an online MPC controller calculates the control signals given the conditions by running an optimization algorithm that iteratively calls, or otherwise references, a system model. It does this optimization at each control timestep in the actual implementation, so the model and optimization algorithm are a part of the controller itself. The offline optimization approach, described herein, produces a ‘lookup controller’, which calculates the control signals given the conditions by interpolation between pre-calculated control signals in a ‘conditions grid’. The term ‘conditions grid’ refers to an n -dimensional grid of conditions values (i.e. n conditions variables, each with a set of values c_i), that is intended to cover the range of possible conditions that the controller might face in its application. The size of the conditions grid is the product of the number of values in each dimension ($\prod_{i=1}^n c_i$).

As discussed in greater depth in the next chapter, for the offline optimization approach to be computationally feasible, the conditions grid size must be limited, which means that the number of conditions variables that can be considered must be kept low (to roughly 5 or 6, given current computing costs). Two methods of limiting the size of the grid are investigated in detail herein: (1) ‘conditions parametrization’,

and (2) ‘problem decomposition’. ‘Conditions parametrization’ refers to a process of correlating conditions variables so that estimated values of a larger set of conditions variables can be derived from a given values of a smaller set of conditions variables - the smaller set is then used in defining the conditions grid, while the larger set is used as inputs to the model in the offline optimization process. ‘Problem decomposition’ refers to the splitting of a problem into subproblems - in some cases the subproblems might be tractable with offline optimization while the original problem is not.

Visualization of the lookup table is an important aspect of the approach described herein. It helps the system designer to understand how the controller is responding to conditions, provides a way of verifying the reasonability of the control responses, affords opportunities to debug the model and optimization configuration, and can lead to insights that might result in improved system designs. Graphs of lookup table slices are used extensively herein. Upon first viewing, they can be difficult to comprehend, but with repeated viewing of these types of graphs they tend to become easier to understand. The next chapter includes an explanation of how to read them, which may help to speed the familiarizing process.

2.2 Research objectives

The objectives of this research are as follows:

- To develop and test methods for offline approximation of MPC with standard building simulation tools, including:
 - techniques for reducing problem dimensionality (conditions parametrization and problem decomposition, as described in the next chapter)
 - conditions grid definition and solution methods
 - visualization methods for the lookup table controller
 - methods for testing the controller through simulated implementations
 - methods for facilitating physical implementations, including human-in-the-loop implementations
- To test the approach’s range of feasibility and applicability through case studies
- To evaluate the lookup controller’s performance relative to online MPC and heuristic rule-based controls
- To identify areas for further research and development

2.3 Dissertation overview

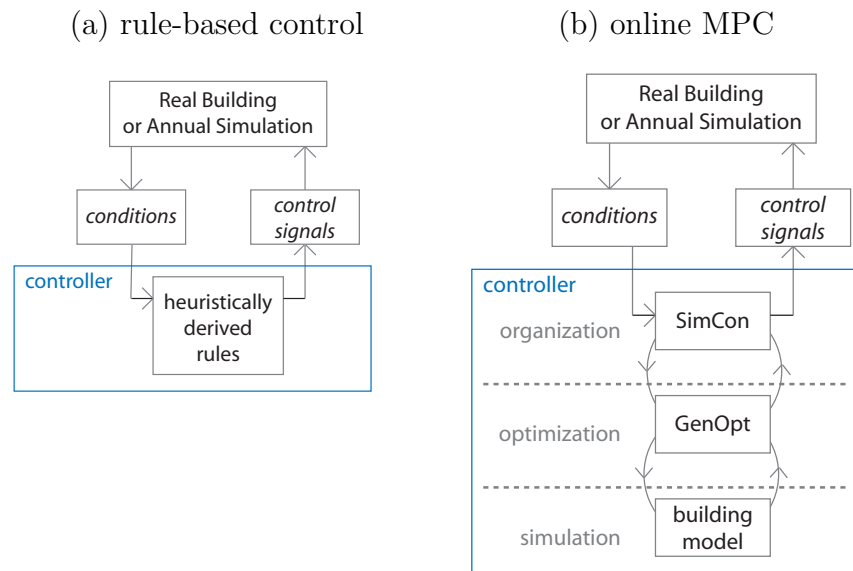
Methods are described in the next chapter, followed by three abstract case studies and two detailed case studies. These case studies are used to demonstrate the approach and its benefits, and are analyzed to address the research questions. In particular, the first abstract case study includes an analysis of the performance impacts of grid spacing, the second abstract case study includes an analysis of the effects of conditions parametrizations, and the third abstract case study considers problem decomposition. The two detailed case studies demonstrate the application of the approach to more complex problems.

3 Methods

3.1 Approaches: Rule-based control, online model predictive control, and lookup control with offline optimization

A heuristic rule-based controller is illustrated in Figure 2a. Controllers like this will be used as base cases for comparison in the case studies that follow. Figure 2b illustrates an online MPC controller, using the open-source software SimCon developed in previous research (Coffey et al., 2010a). SimCon acts as an organization layer, reading the conditions and setting up the optimization problem at each controller timestep, which is then solved by GenOpt and a building model in any simulation tool that can be used with GenOpt (ie. any tool that uses text files as inputs and outputs and can be called from the command line). The optimal control signals are then collected from GenOpt by SimCon and sent to the actual building (or annual simulation) for use.

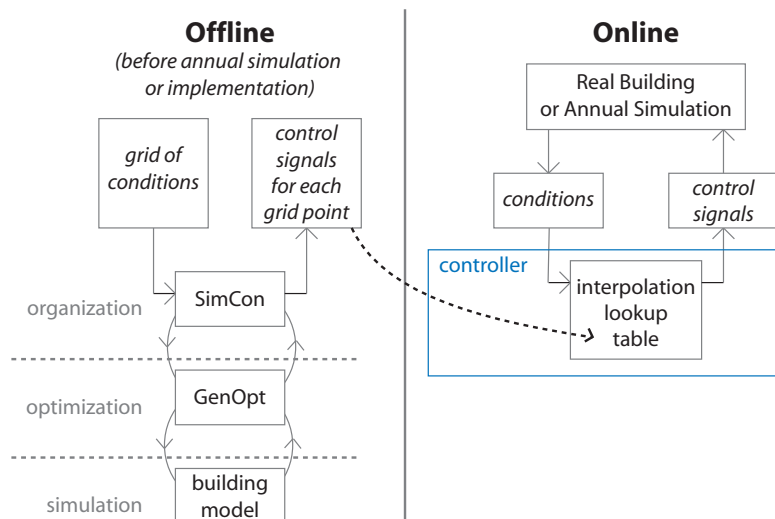
Figure 2: Controller configurations: rule-based control versus online MPC



As noted earlier, there are various challenges associated with implementing online MPC controllers in real buildings, or even with performing annual simulations of the

control, in large part because of the long run-times of standard building simulation tools. But the same basic structure can be used offline to produce lookup tables for near-optimal control, as shown in Figure 3. The optimizations over the conditions grid can be performed in parallel and are carried out prior to implementation, so that the optimizations do not need to be run in real-time within the controller.

Figure 3: Controller configuration for lookup control using offline-optimization



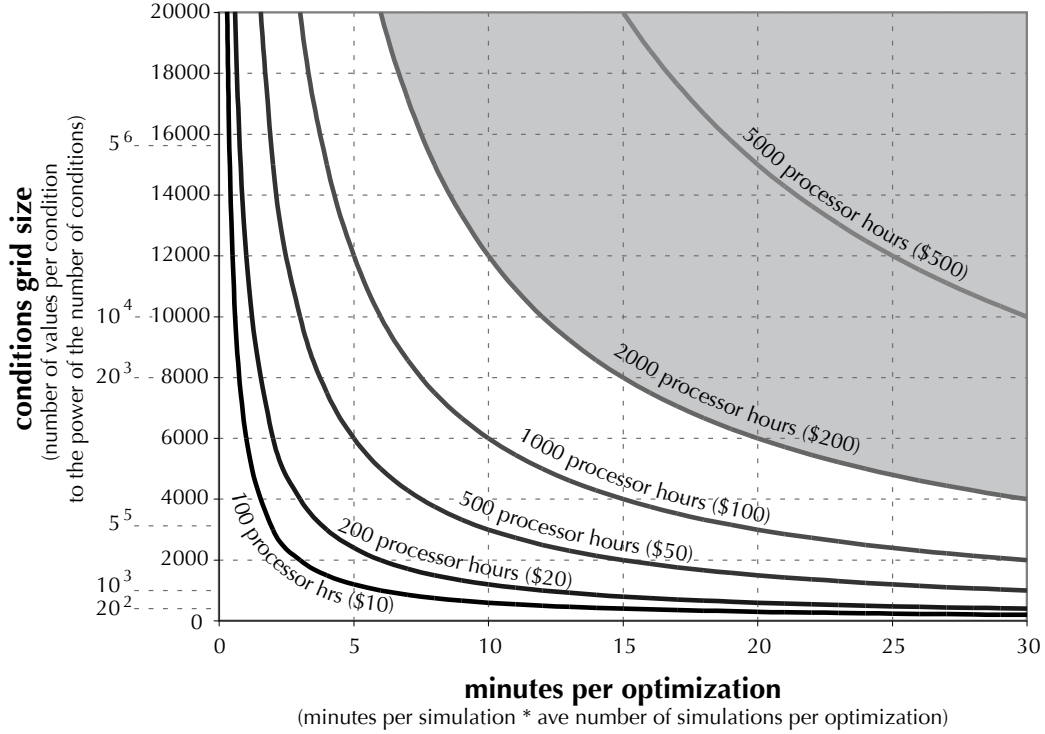
The resulting lookup table can be used for supervisory control in physical implementation and annual simulations much more readily than online MPC. In addition, the lookup table provides an excellent way of visualizing the controller’s behavior, which facilitates debugging and can also lead to a better understanding of the building system under consideration. These aspects are described later in this chapter and demonstrated through the case studies in subsequent chapters.

3.2 Range of computational feasibility

This approach of using lookup tables has numerous benefits. The big question, however, is how feasible it is for what types of problems. Figure 4 considers the question in terms of how much computing time would be needed for particular problem configurations. The figure is based on the fact that the number of processor hours required is equal to the product of the conditions grid size and the average number of processor hours per optimization. The conditions grid size is the product of the number of

elements per dimension. The processor hours per optimization is the product of the simulation time and the number of simulations per optimization. The dollar costs in the figure are based on \$0.10 per processor hour, which is roughly the current commercial cloud computing cost for small-scale users.

Figure 4: Range of feasibility of lookup table approach



The shaded area is a somewhat conservative cut-off point for feasibility for a consulting or design firm working on a single building, although it might easily be extended upwards, perhaps to the 5000 processor hours line. It shows the trade-offs in terms of simulation time (ie. model complexity), optimization precision and grid spacing, and it shows the scale of problems that are feasible - simulation time for the control model must be within seconds (note that this is over a simulation length of a few hours to a few days, rather than a full year), and the number of conditions variables must be kept to less than 5 or 6.

As such, this approach usually requires that approximations be made to limit the dimensionality of the lookup table, and it is not appropriate for use in all cases.

A significant part of the research described herein is aimed at better understanding the performance costs associated with various types of approximations that might be used to limit lookup table size. Two methods of approximation to limit dimensionality are described in the next two sections, followed by an explanation of methods to solve the optimizations over the conditions grid.

3.3 Conditions parametrization

The optimization problem at each timestep in MPC may be generalized to Equation 1, where f is the objective function (output from the simulation tool or from a post-processor), u is the vector of control variables, U is the set of allowable values for u , x_0 is the vector of initial state variables, and w is the vector of disturbances and disturbance predictions. (Note that this form does not explicitly consider state constraints or terminal constraints, but they may be considered as penalty functions within f .)

$$\min_{u \in U} f(x_0, u, w) \tag{1}$$

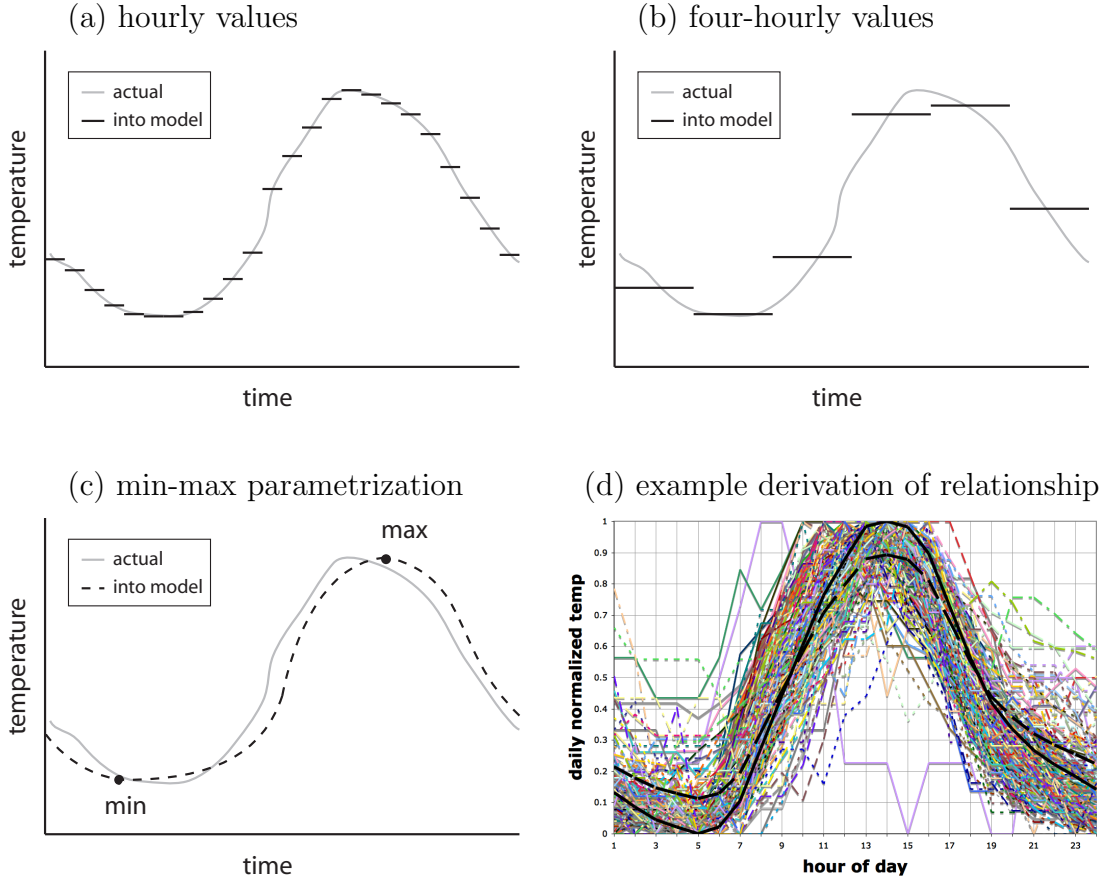
The overall approach, as noted above, is to solve this problem offline over a grid of conditions, i.e. over a grid of values for x_0 and w . The challenge, however, lies in the number of elements in the vectors x_0 and w , which defines the dimensionality of the conditions grid and of the resulting lookup table of solutions to be used for control.

Conditions parametrization is used to decrease the dimensionality of the conditions grid. The basic idea is to create a mapping from the values of a small number of conditions parameters (used in the conditions grid) to a large number of conditions values that are used in the building simulation model. This process is dependent on the structure of the problem under consideration. However, some types of disturbance predictions are common across many buildings-related applications, such as ambient temperature or solar gains predictions, and some of these disturbances have strong patterns across samples, such as the sine-wave shape of ambient temperatures over the course of a day. So useful examples can be given.

Consider a control problem that requires day-ahead predictions of ambient temperatures, such as the overnight cooling of a massive chilled floor or ceiling. If hourly predictions were used (as illustrated in Figure 5a), this would require 24 dimensions in the lookup table just for the ambient temperature. If 5-minute predictions were used, 288 dimensions would result. One way of decreasing the dimensionality would

be to use a coarser prediction, for example using average temperatures for 4-hour blocks instead of 1-hour blocks, as shown in Figure 5b.

Figure 5: Example parametrizations



A conditions parametrization approach to this case is to make use of the expected shape of the diurnal temperature profile, and use a smaller number of parameters to define the prediction. In this example, the parameters could be the maximum and minimum temperatures, as shown in Figure 5c. Variants on this include a 1-parameter approximation using just the maximum or minimum temperature and assuming a constant range, and a 4-parameter approximation that uses the maximum and minimum temperatures and their times of occurrence. In all of these variations, a normalized curve is required to produce the values of the temperature at each timestep of the simulation, based on the parameter values. In the case of ambient

temperature, there are standard curves that can be used for this (ASHRAE, 2005; Thevenard, 2009). A more generally-applicable method is to derive a curve based on typical or historical data for the site by normalizing each day in the data set by its maximum and minimum values, finding this set of normalized days' average hourly values, and using these values as the basis for the normalized curve - Figure 5d illustrates this, and the process is demonstrated in greater detail in the abstract case study #2 in Chapter 5.

Other disturbance predictions that might be similarly parametrized include diffuse and direct solar, occupancy and plug loads. Solar predictions could be parametrized in various ways, including by length of day and cloudiness factor, making use of solar position equations. Other predictions, such as occupancy and plug loads, might be harder to generalize by making use of underlying physics or geometry, but could also be approached by deriving parametric curves based on historical data.

It is also possible to correlate disturbance variables to one another in order to decrease the size of the conditions grid. The case study in Chapter 8 provides an example of this, approximating all the variables in a weather files as a function of ambient temperature, direct normal radiation and diffuse normal radiation.

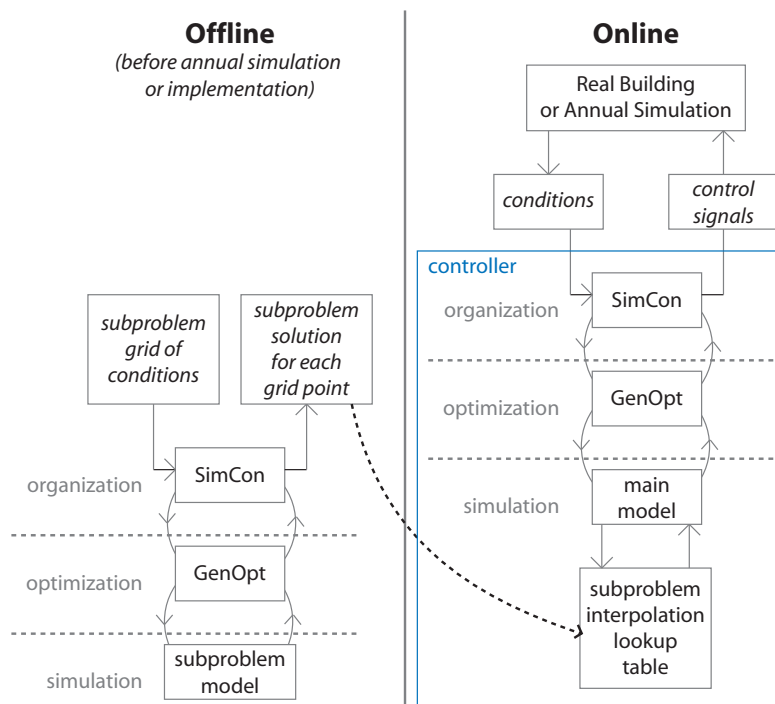
Approximated state initialization

Similar approaches may also be used for initial states, if there are a lot of initial states of interest (e.g. in cases with thermal storage in many different tanks or massive building components). But the bigger concern pertaining to state initialization in buildings-related control is that many common building simulation tools do not provide easy access to the state vector. Instead of being able to explicitly initialize states, one often has to use an 'initialization horizon' (termed a 'pre-conditioning horizon' in the papers of Henze et al), simulating over days or weeks of previous disturbances and control actions to get the model state values to be roughly equal to the observed initial state values. This means that instead of having a small vector of initial states as part of the conditions grid, there would have to be a much longer vector of previous disturbances and previous control actions in the conditions grid, ballooning its dimensionality. One approach to this problem, and perhaps the simplest, is to define an initialization horizon for the simulation where all of the values are constant except some small number of parameters that can be modified to produce the desired initial states. As such, a desired initial state value can be translated into parameter values in the simulation. This technique is demonstrated in the abstract case study #2 in Chapter 5.

3.4 Problem decomposition

Even with the approximations of input parametrization, many control problems in buildings still have a dimensionality that is too high to be tractable as lookup tables. However, the lookup table approach may still offer some help in such a case, if the structure of the problem allows it to be decomposed into a hierarchical set of problems, where some of the subproblems are small enough to be solved offline. Figure 6 illustrates a case with one subproblem that is solved offline.

Figure 6: Controller configuration for lookup control using offline-optimization



Generic mathematical description

For any MPC optimization problem,

$$\min_{u \in U} f(x_0, u, w) \quad (2)$$

given a parsing of a control vector u into components as follows,

$$u = [u_\alpha, u_{\beta_1}, u_{\beta_2}, \dots, u_{\beta_N}] \quad (3)$$

if the corresponding subsets of U , i.e. $U_\alpha, U_{\beta_1} \dots U_{\beta_N}$, are independent (e.g. in problems with only box constraints on the control variables), and if the effects on the objective function f of the components u_{β_i} , $i \in 1 \dots N$, are independent, then the following optimization problem is equivalent to the problem in Equation 2 above, with $u_{\beta_j}^k$ representing arbitrary but constant values within U_{β_j} :

$$\min_{u_\alpha \in U_\alpha} f(x_0, u_\alpha, u_{\beta_1}^{*|u_\alpha}, \dots, u_{\beta_j}^{*|u_\alpha}, \dots, u_{\beta_N}^{*|u_\alpha}, w) \quad (4)$$

$$\text{where } u_{\beta_i}^{*|u_\alpha} = \min_{u_{\beta_i} \in U_{\beta_i}} f(x_0, u_\alpha, u_{\beta_i}, u_{\beta_1}^k \dots u_{\beta_{j \neq i}}^k \dots u_{\beta_N}^k, w) \quad \forall i \in 1 \dots N$$

Parsing the problem in this manner may be beneficial in cases where all or some of the subproblems in Equation 4 are of low enough dimensionality to be pre-solved offline, and particularly beneficial if the subproblems are repeated. Note, however, that simply decomposing a problem does not necessarily make it easier to solve as a whole. (In fact, for fully linear or quadratic problems with all continuous variables, such decomposition can in some cases actually lead to slower solutions.) For mixed-integer non-linear problems, using heuristic optimization algorithms with opaque models (as often describes the situation when performing optimization with building simulation tools), injecting expert knowledge into the optimization process through problem decomposition will often result in a faster overall solution, but it is not guaranteed. The main reason for considering problem decomposition is if the subproblems can have few enough conditions dimensions to be solved offline as a lookup table, thus allowing some of the optimization computation requirement to be handled offline before implementation. In some cases, this can result in a faster online solution. Chapters 6 and 7 show particular examples of this.

Similar to conditions parametrization, the application of problem decomposition depends on the details of the problem under consideration. Below are two conceptual examples to illustrate some types of problems for which the approach might be useful.

Example with repeating subproblems

Consider the optimization of overnight charging lengths and start times for a chilled water storage system with a 7-day prediction horizon (this is akin to the case

study in Chapter 7). Even if the approximation of using just the minimum and maximum daily temperatures is acceptable, that still leaves 14 dimensions for the lookup table, which is likely too many to be practical for offline solution. However, the optimization can be broken into a higher level problem that determines the charge length for each night (referencing the lower level problem solution as it does so) and a lower level problem that determines the optimal start time and energy consumption for a particular night given the charge length and the overnight temperature predictions. With this decomposition, the lower level problem could be solved as a lookup table, leaving the higher level problem to be solved as an online MPC, but now potentially faster since the online MPC problem has just 7 control variables to consider (only the charge lengths), instead of 14 (the charge lengths plus the start times).

Example with conditionally-applicable optimization variables

Consider a mixed mode HVAC control problem in Equation 5, with u_1 being a vector of window opening percentages and u_2 being a vector of mechanical cooling zone setpoint values, over some prediction horizon of length N .

$$\min_{u_1, u_2} f(x_0, u_1, u_2, w) \tag{5}$$

To the decision variables u_1 and u_2 add a third decision variable u_α to signify a binary decision between natural and mechanical ventilation, and reconsider the problem as follows, where, for example, $k_1 = 0\%$ (ie. the windows are closed) and $k_2 = 30C$ (ie. the mechanical cooling is deactivated).

$$\begin{aligned} \min_{u_\alpha, u_1, u_2} f(x_0, u_1, u_2, w) & \tag{6} \\ \text{s.t. } u_\alpha & \in \{0, 1\} \\ & \text{if } u_\alpha = 0 \text{ then } u_1 = k_1, u_2 \in U_2 \\ & \text{if } u_\alpha = 1 \text{ then } u_1 \in U_1, u_2 = k_2 \end{aligned}$$

This can be implemented with two optimizers as shown in Equation 7. Note that the U sets in Equation 7 could be subsets of those in Equation 6.

$$\begin{aligned}
\min_{u_\alpha \in \{0,1\}} \quad & \min_{u_1, u_2} f(x_0, u_1, u_2, w) & (7) \\
\text{s.t.} \quad & u_\alpha \in \{0, 1\} \\
& \text{if } u_\alpha = 0 \text{ then } u_1 = k_1, u_2 \in U_2 \\
& \text{if } u_\alpha = 1 \text{ then } u_1 \in U_1, u_2 = k_2
\end{aligned}$$

Instead of the original optimization problem with $2N$ continuous decision variables, we now have a hierarchical problem, with N binary decision variables at the top level and N continuous variables in the subproblem (ie. only N continuous variables are active during each call to the subproblem optimizer, since the other N continuous variables are constrained by the equality constraints). Even without offline solution of the subproblem, this could result in a faster solution (although there is no guarantee of that, as discussed above), and it should avoid some cases of equivalent solutions by keeping the optimization from searching in inactive parts of the search space. More significantly, if the subproblem (in either or both of its modes) can be squeezed into few enough conditions dimensions through conditions parametrization, then it can be solved offline over a conditions grid, which could allow for a faster online solution to the main problem.

Approximations in problem decomposition

In the generic mathematical description above, the decomposed problem is mathematically identical to the original problem. As such, problem decomposition does not necessarily imply approximation. However, the same approach can be used to split a problem into components when those components are nearly but not entirely independent, ie. if the effects on the objective function f of the components u_{β_i} , $i \in 1 \dots N$, are *not* independent, but one nonetheless simplifies the problem from Equation 2 to Equation 4. The detailed case study of the UC Merced chilled water system in Chapter 7 involves an approximation like this in its decomposition. Various other control problems in buildings could have similar approximated decompositions.

3.5 Solution methods for the conditions grid

Once a conditions grid has been defined, optimizations must be carried out for all of the points in that grid. In all of the case studies herein, GenOpt has been used for the optimizations, primarily with the GPS-Hookes-Jeeves algorithm. But different optimization algorithms or platforms could easily be used instead, while still fitting within the same conceptual framework.

The optimizations for each grid point can be solved independently, and thus the computation can be massively parallel. The Amazon Elastic Compute Cloud (EC2) has been used extensively in the case studies, with numerous virtual machines solving different parts of the conditions grid.

The other noteworthy aspect of solving over the conditions grid is that it can be done iteratively - one can begin by solving over a relatively coarse grid and potentially with fewer dimensions (by holding some of the dimensions constant), view the results, debug the model and optimizer if necessary, refine the grid if necessary (potentially focusing grid precision in particular areas rather than always using a uniform spacing, although this has not been studied in the case studies and analyses herein), and re-run the optimizations if changes were made or just continue solving over the remaining grid points if no changes were made, repeating this process as desired until the final grid precision is reached. The overall approach thus provides many opportunities for debugging the model and the optimization configuration as the grid solution process unfolds.

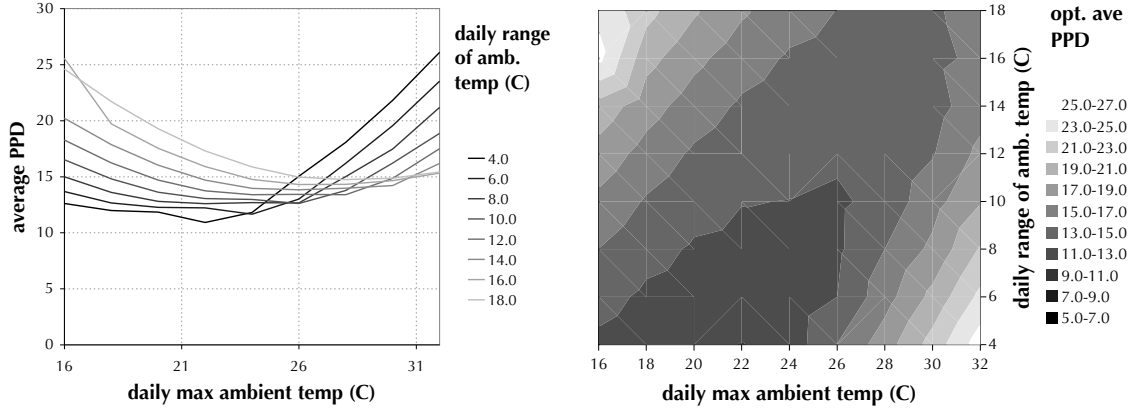
3.6 Lookup table visualization

Although the main impetus for exploring this approach was the potential benefits of faster annual simulations and easier physical implementations, the ability to explore and better understand the controller's behavior through lookup table visualization is perhaps even more beneficial. It facilitates debugging, and it can lead to insights into the nature of the problem under consideration, which could lead to better system design and/or to the development of better simplified controls for the system.

The challenge in visualizing the solutions grid lies in its being multi-dimensional, and the fact that it is difficult to show more than 3 dimensions in a flat graphic. However, by holding all but two of the conditions dimensions constant and selecting just one output variable at a time (since each point in the conditions grid has a solution consisting of a vector of control variable values and the objective function value), one can view a particular slice through the lookup table as a 2-dimensional graphic.

As an example, Figure 7 shows some conditions grid results from abstract case study #2 (Chapter 5). The lookup table in this case is 6-dimensional. For the graphs shown in Figure 7, 4 of the 6 conditions dimensions are held constant, while the other two are shown varying over their range. Figures 7a and 7b show the same slice of the lookup table in two different ways. Sometimes one of these views is more intuitive than the other.

Figure 7: An example lookup table slice shown in two different ways



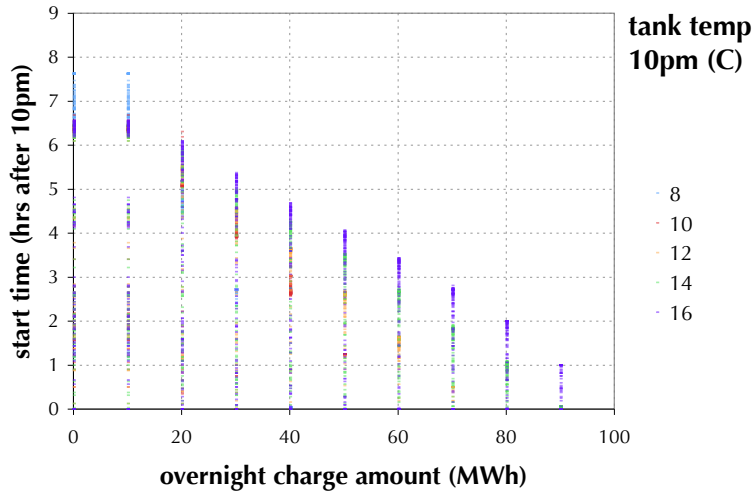
In the lines plot of Figure 7a, one of the conditions dimensions (daily max ambient temp) is on the horizontal axis, and the output variable (average PPD, an objective function output in this example) is on the vertical axis. The different lines correspond to different values of the second conditions variable (daily range of amb. temp), with the darker lines showing smaller values of that second conditions variable.

In the surface plot of Figure 7b, the two conditions dimensions are along the vertical and horizontal axes, and the output variable value is represented by the tonal gradient, again with darker tones representing lower values.

More dimensions can be shown by repeating these slice graphs for different values of the dimensions that are being held constant for each particular slice graph. This approach is used extensively in displaying the grid results in the case studies that follow.

Two-dimensional scatter plots of all the points in the grid can also provide useful information, particularly in showing if there are strong patterns in how two variables relate. For example, the scatter plot in Figure 8 shows that for the chilled water system under consideration (this graph is taken from the case study in Chapter 7), the optimal start time is limited by the charge length such that longer charge lengths mean earlier optimal start times, but the optimum start time varies within the constraints imposed by the charge length, in response to the other conditions values (which includes both the initial tank temperature shown as different colors in the graph, and the minimum and maximum overnight wet bulb temperatures).

Figure 8: Example lookup table scatter plot



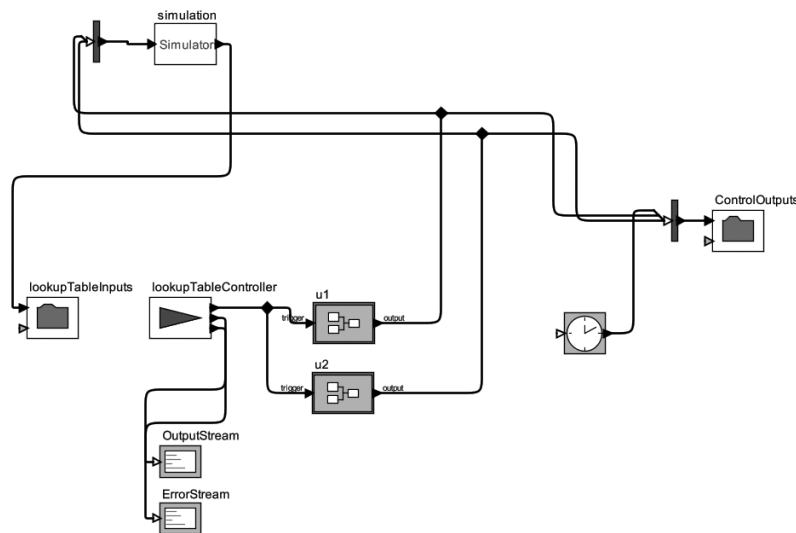
3.7 Simulated implementation

To test the lookup table controller and to analyze its performance over a typical year or over some historical period of disturbance data, one may simulate it in much the same way that one would simulate any other controller. The simulation is done with a model that may either be exactly the same as the model used in the offline optimizations (simulating the theoretical case of zero mismatch between the model within the controller and the behavior of the real system), or it may be a modified version of that model or a more detailed model. Within the simulation model, there must be a call at each controller timestep to a method that interpolates over the lookup table to calculate the control variable values given the current conditions in the simulation (including the current predictions, which may either be perfectly matched to the upcoming simulation conditions, for the theoretical case of perfect predictions, or a modification thereof).

In some cases, with some building simulation tools, the call to the lookup table interpolation can happen directly within that simulation tool: for example, if the lookup table is just two dimensions and the tool being used is Modelica, then one can simply use the `CombiTable2D` object in the standard Modelica library to perform the interpolation. However, a more generic option is to use the Building Control Virtual Test Bed (BCVTB, (Wetter and Haves, 2008)) to couple the simulation model to an external process that performs the interpolation and sends the control variable

values back to the simulation. A diagram of the basic BCVTB configuration used to perform this function in the case studies herein is shown in Figure 9.

Figure 9: Simulated implementation with BCVTB



In each of the case studies that follow, the annual performance of the lookup controller will be compared with a heuristic rules-based controller (as illustrated in Figure 2a), and in some of the cases it will also be compared with an online MPC controller (Figure 2b). These are also simulated through the BCVTB, swapping out the lookup controller for a rules-based controller (generally also programmed in java in these case studies) or for a direct call to the online MPC using similar SimCon-GenOpt configurations as used for the lookup table creation.

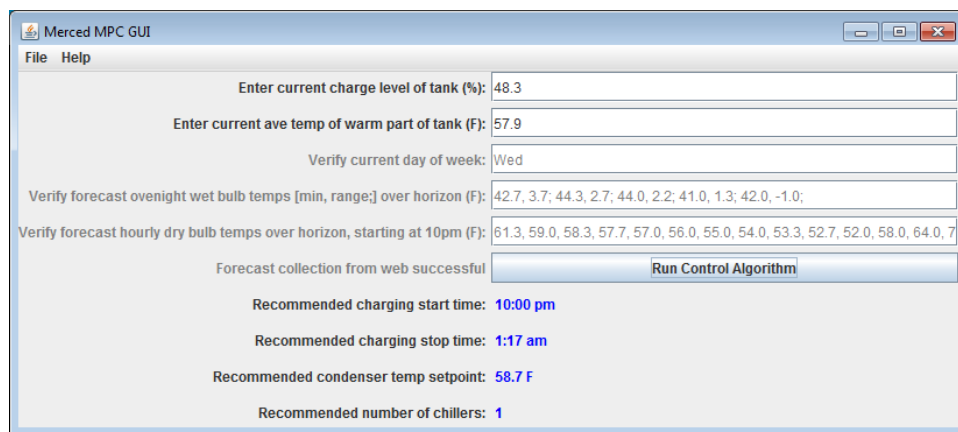
3.8 Physical implementation

Implementation in a real system is just a matter of feeding initial state values and current and predicted disturbance values to a component that interpolates over the lookup table and returns the control values to be used. Aside from technical challenges associated with linking to closed-source proprietary control systems in buildings, setting up an automated procedure for this is generally simple, and can use the same interpolation component used in the simulated implementation. Weather predictions are usually available on the internet, for example through the National Digital Forecast Database (NDFD), which can be accessed through a Simple Object

Access Protocol (SOAP) programming interface. One notable complexity, however, in moving from simulated implementation to physical implementation is the treatment of state variables - in simulation these can usually be treated as known values, but in physical implementation they must often be derived from measured values.

As an alternative to a direct automated connection between the building control system and the lookup table controller, for use in testing or to simplify the implementation in some cases, a human interface for the lookup table controller can be used, like the one shown in Figure 10, which has been developed for the chilled water system at UC Merced, as discussed in Chapter 7. The user inputs the initial state values. Weather predictions are automatically downloaded from the NDFD and can be checked by the user. The computed control setpoints are then provided to the user, who can check them for reasonableness before manually implementing them in the real system.

Figure 10: Example human-in-the-loop interface for lookup table controller



3.9 Open-source software

Open-source software for online MPC with standard building simulation tools was developed in previous research (Coffey et al., 2010a), using GenOpt (Wetter, 2009a) as the optimizer, which allows for the use of any text-based building simulation tool (e.g. DOE-2, EnergyPlus, TRNSYS, Modelica). This software has been extended to be used for computing control lookup tables. The extended software has been developed and tested through the case studies described in the chapters that follow.

The software is written in java, and the source code is freely available for download (Coffey, 2011). The software's current functionality includes:

- The option of running in either online MPC mode or in lookup table calculation mode, so it can still be used for online MPC, including cases with decomposed problems that involve a higher-level problem that must be solved online and lower-level problems that can be solved offline as lookup tables
- Core methods to set up a sequence of optimization problems for a user-defined conditions grid, solve them using GenOpt, and record the results to a lookup table file
- Multi-variable interpolation for lookup tables stored in a text file format
- An extensible library of algorithms to convert conditions inputs to and from parametrized forms

Peripheral components are also included in the open source code, including sample BCVTB files for running annual simulations with the controller, java methods for collecting weather predictions from the National Digital Forecast Database, sample java-based interfaces for the controller, a visualization tool for the lookup tables (currently in Excel with VBA), and two conditions parametrization tools (also currently in Excel with VBA).

3.10 Case studies

Five case studies are presented in the following chapters. The first three are abstract studies designed to demonstrate and test particular aspects of the approach. The last two are more detailed studies of more realistic systems.

Abstract case study #1 (Chapter 4) demonstrates the general approach and is used to analyze the relationship between grid spacing and controller performance. Abstract case study #2 (Chapter 5) demonstrates the use of conditions parametrization and analyzes its impacts on controller performance. Abstract case study #3 (Chapter 6) demonstrates the use of problem decomposition. All of the case studies show the methods of lookup table visualization and simulated implementation, and show the range of the applicability (and degrees of suitability) of the approach both in terms of problem types and simulation tools. They also each produce various lessons for potential users and for further developments of the approach.

4 Abstract Case Study #1: Active Facade, Grid Spacing

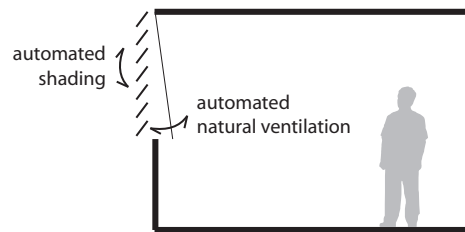
4.1 Case study: Solar shading and natural ventilation

This simplified case study illustrates the basic methods for making a lookup table controller. Because the problem is small, no problem decomposition or conditions parametrization is required. The model is simple enough to have all of the equations written out herein, so that it could be easily replicated in a different modeling tool if desired, and so that the calculation results can be more easily traced back to the model equations to verify that the control responses are reasonable.

4.1.1 Case description

Consider a single south-facing zone with automated external shading and automated natural ventilation, as illustrated in Figure 11. A supervisory controller must determine hourly setpoints for the shading percentage (defined as the percentage decrease in solar gains, not necessarily a linear function of the shade angle) and the natural ventilation percentage (defined as the number of air changes per hour (ACH) relative to its minimum and maximum possible values) to minimize the combined heating, cooling and lighting energy consumption.

Figure 11: Solar shading and natural ventilation case study illustration



The control is simulated over a typical meteorological year in San Francisco, the annual savings are compared against a simple heuristic control strategy, and selected weeks are also compared against a full online MPC implementation. To keep the problem simple, the zone is considered as massless, glare is not considered in the shading control, a highly simplified approach to the daylighting calculation is used, the dimmable lights are controlled perfectly to maintain a pre-specified lighting level at a point in the space, the daylighting and solar gains vary in proportion when the

shading percentage varies, and the variable capacity for natural ventilation under different wind and thermal conditions is ignored.

4.1.2 Model description

A one-zone thermal model is used for this example. As shown in Figure 12, the model inputs are of two types: disturbances ($T_{amb}(C)$, $\dot{Q}_{solarDirect}(W/m^2_{window})$, $\dot{Q}_{solarDiffuse}(W/m^2_{window})$, $\dot{Q}_{peopleAndPlugLoads}(W/m^2_{floor})$); and control inputs ($u_{shading}$, $u_{natVent}$). The model consists of Equations 8 - 19, with the parameter values in Table 1.

Figure 12: Massless zone model

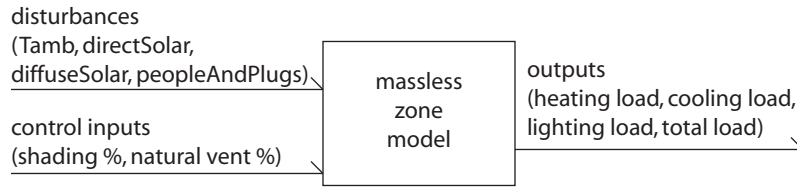


Table 1: Model parameters

T_{zone}	22	C	zone setpoint temperature
U_{wall}	0.25	W/m^2K	wall U-value
U_{window}	3	W/m^2K	window U-value
A_{floor}	10	m^2	floor area
$h_{ceiling}$	3	m	ceiling height
A_{wall}	6	m^2	wall area (excluding window)
A_{window}	3	m^2	window area
$infil_{min}$	0.25	ACH	minimum infiltration + ventilation rate
$infil_{max}$	5.0	ACH	maximum infiltration + ventilation rate
$SHGC$	0.5	—	window solar heat gain coefficient
β_{direct}	1	lux/W	desktop illuminance per unit of direct solar gain
$\beta_{diffuse}$	5	lux/W	desktop illuminance per unit of diffuse solar gain
$\alpha_{lightLevelSetpoint}$	500	lux	desktop illuminance setpoint
$\dot{Q}_{lightGainMax}$	7	W/m^2	heat gain from lights at full power
$\mu_{heating}$	0.75	—	heating system efficiency
$\mu_{cooling}$	3.0	—	cooling system COP
$\mu_{lighting}$	5	lux/W	desktop lighting efficiency
$c_{p,air}$	1012	J/kgK	specific heat of air
ρ_{air}	1.2	kg/m^3	density of air

$$\dot{Q}_{conduction} = (T_{amb} - T_{zone}) \cdot (U_{wall}A_{wall} + U_{window}A_{window}) \quad (8)$$

$$\begin{aligned} \dot{Q}_{infil} = & (T_{amb} - T_{zone}) \cdot (infil_{min} + u_{natVent} \cdot (infil_{max} - infil_{min})) \\ & \cdot (1/3600) \cdot A_{floor} \cdot h_{ceiling} \cdot \rho_{air} \cdot c_{p,air} \end{aligned} \quad (9)$$

$$\dot{Q}_{solarGain} = (\dot{Q}_{direct} + \dot{Q}_{diffuse}) \cdot A_{window} \cdot SHGC \cdot (1 - u_{shading}) \quad (10)$$

$$\alpha_{daylightLevel} = (\beta_{direct} \cdot \dot{Q}_{direct} + \beta_{diffuse} \cdot \dot{Q}_{diffuse}) \cdot (1 - u_{shading}) \quad (11)$$

$$\alpha_{artificialLightLevel} = \max(\alpha_{lightLevelSetpoint} - \alpha_{daylightLevel}, 0) \quad (12)$$

$$\begin{aligned} \dot{Q}_{internal} = & ((\alpha_{artificialLightLevel} / \alpha_{lightLevelSetpoint}) \cdot \dot{Q}_{lightGainMax} \\ & + \dot{Q}_{peopleAndPlugLoads}) \cdot A_{floor} \end{aligned} \quad (13)$$

$$\dot{Q}_{thermalLoad} = \dot{Q}_{conduction} + \dot{Q}_{infil} + \dot{Q}_{solarGain} + \dot{Q}_{internal} \quad (14)$$

$$\dot{Q}_{heatingEnergyCons} = (1/\mu_{heating}) \cdot \min(\dot{Q}_{thermalLoad}, 0) \cdot (-1) \quad (15)$$

$$\dot{Q}_{coolingEnergyCons} = (1/\mu_{cooling}) \cdot \max(\dot{Q}_{thermalLoad}, 0) \quad (16)$$

$$\dot{Q}_{lightingEnergyCons} = (1/\mu_{lighting}) \cdot \alpha_{artificialLightLevel} \quad (17)$$

$$P_{multSolPenalty} = M \cdot \max(\alpha_{daylightLevel} - \alpha_{lightLevelSetpoint}, 0) \cdot u_{natVent} \quad (18)$$

$$\begin{aligned} \dot{Q}_{totalEnergyCons} = & \dot{Q}_{heatingEnergyCons} + \dot{Q}_{coolingEnergyCons} + \dot{Q}_{lightingEnergyCons} \\ & + P_{multSolPenalty} \end{aligned} \quad (19)$$

The penalty function $P_{multSolPenalty}$ in Equation 18 was added to the model after learning from initial test iterations, as discussed below. It is used to avoid noise caused by the existence of multiple optimal control solutions in some parts of the conditions space. When computing the lookup table and within the online MPC model, M is set to an arbitrarily large number. When the model is being used as the simulation test model in annual simulations, M is set to zero.

A Modelica model with these equations was created in Dymola. The model reads a disturbance input text file and a controls input text file, and writes $\dot{Q}_{totalEnergyCons}$ to an objective function output text file.

4.1.3 Optimization configuration

The multi-dimensional grid used for the disturbances was as shown in Table 2. This grid configuration results in 2880 sets of disturbances for which optimal control was to be determined. The Hookes-Jeeves algorithm in GenOpt was used, with 5 step size reductions, and non-parallelized so running on just one processor. The sequential optimizations required approximately 84.5 hours (3.5 days) on a single Windows virtual machine.

Table 2: Disturbances grid

	min	max	step size
ambient temp (C)	0	30	2
direct solar ($\text{W}/\text{m}^2_{window}$)	0	1000	200
diffuse solar ($\text{W}/\text{m}^2_{window}$)	0	1000	200
gains from people and plugs ($\text{W}/\text{m}^2_{floor}$)	2	18	4

4.1.4 Lookup tables results

Figure 13 shows the optimal shading and natural ventilation levels as a function of the ambient temperature, direct and diffuse solar irradiation, and internal heat gains. In each of the graphs, the control signal is on the vertical axis and the ambient temperature along the horizontal axis, and the different lines correspond to different internal load levels (the higher the internal load level, the darker the line). For each individual graph, the other two conditions dimensions (direct solar and diffuse solar) are fixed at constant values. The graphs are repeated with different values of these two conditions to show their impacts on the control responses.

Figure 14 highlights the response of the optimal shading control to different values for direct and diffuse solar. In these graphs, both the vertical and horizontal axes are used for conditions (direct solar and diffuse solar), with the control signal (shading level) shown by the gradient from black (no shading) to white (fully shaded).

Figure 13: Optimal shading and natural ventilation levels

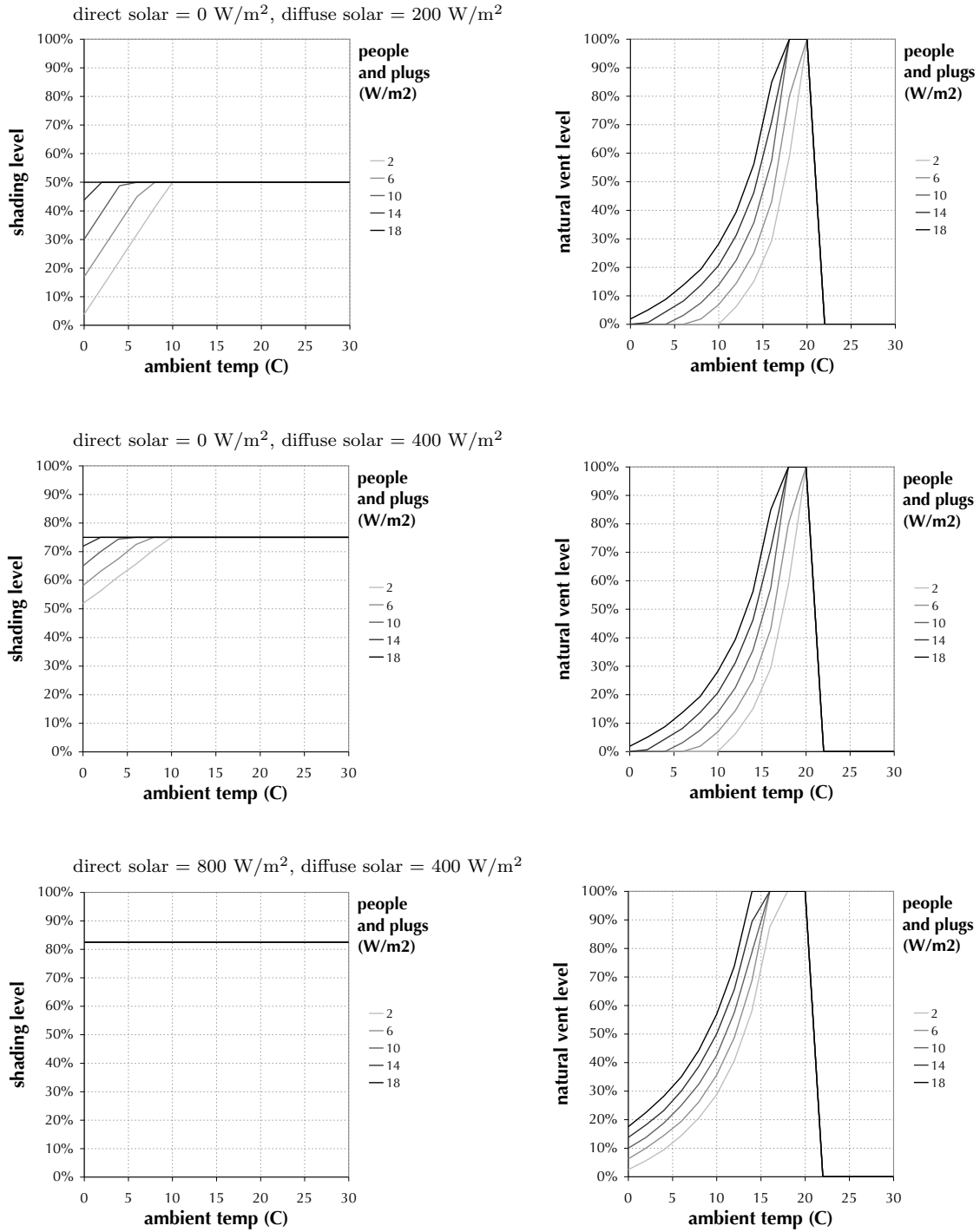
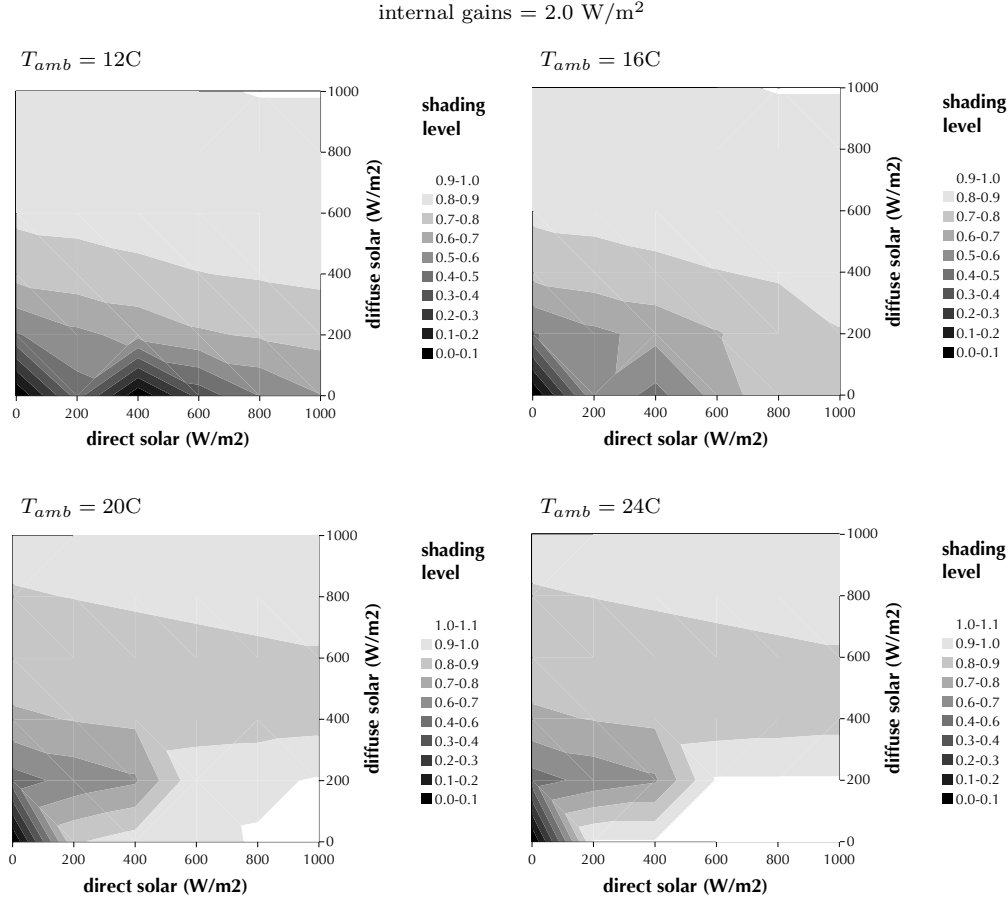


Figure 14: Optimal shading levels versus direct and diffuse solar gain (white = fully shaded, black = no shading)



The results make intuitive sense. The natural ventilation level generally increases from 0% to 100% as ambient temperature goes from very cold up to the zone temperature setpoint ($22^{\circ}C$), and then drops back down to 0%. The solar shading level increases with greater solar gains, except when the ambient temperature and internal loads are low enough to warrant using the solar gains to heat the zone. When the ambient temperature is slightly lower than the zone setpoint, the solar shading only approaches zero as the solar gains approach zero and the light can be used for daylighting. However, when the ambient temperature is greater than the zone setpoint, the shading control blocks the sun completely when the direct solar is high and the diffuse solar is low (cooling these solar gains is more costly than using the artificial

lights), but with a more favorable direct-diffuse split the shading control allows just enough daylighting in to eliminate the artificial lighting requirement.

4.1.5 Annual simulations

To test the lookup table control, an annual simulation was run using TMY data for San Francisco. The TRNSYS weather reader was used to calculate the solar gains on the window (assuming south-facing glazing) given the time of day, day of year and direct and diffuse data in the TMY file. Figure 15 shows the hourly disturbances values over the year. A weekly schedule for internal loads was created for the annual simulation, as shown in Figure 16. The annual simulations were carried out using a copy of the Modelica model as the ‘real’ system, connected to a controller through the BCVTB. A full annual simulation with hourly calls to the lookup table controller requires about an hour of computation time.

Figure 15: Weather disturbances used in the annual simulation

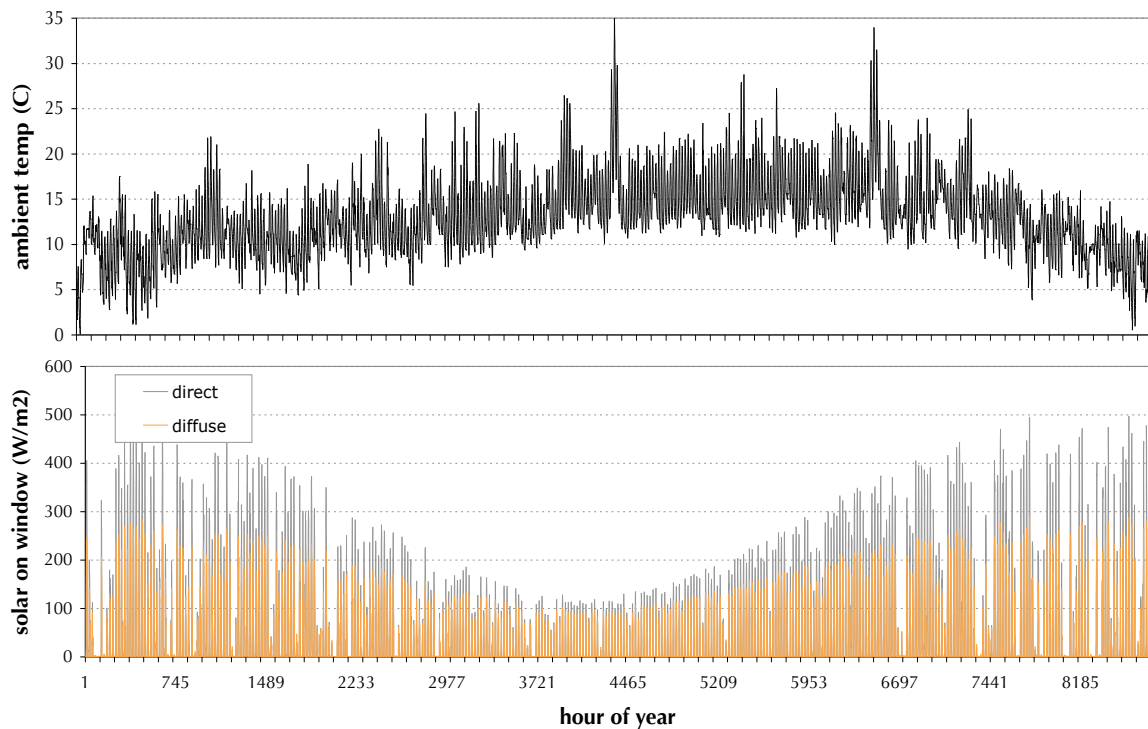
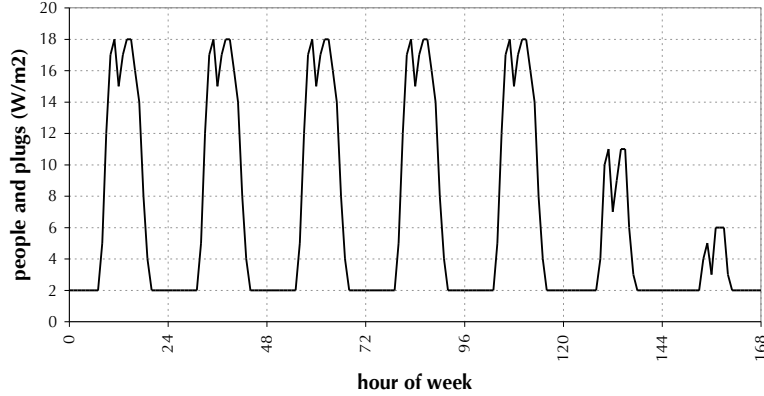


Figure 16: Weekly schedule for waste heat from people and plugs



For comparison with the lookup table controller, a base case controller was also simulated, with the following control rules (derived as heuristic simplification of the trends in the optimal control shown in Figure 13): if $T_{amb} \leq 12C$ then $u_{shading} = 0\%$ and $u_{natVent} = 0\%$; if $12C < T_{amb} \leq 22C$ then $u_{shading} = 75\% \cdot (T_{amb} - 12C)/10C$ and $u_{natVent} = 100\% \cdot (T_{amb} - 12C)/10C$; otherwise $u_{shading} = 75\%$ and $u_{natVent} = 0\%$. This base case controller is not intended to be the best controller that could be devised without a model-based approach (for example, a much better heuristic controller could make use of the idea that when the zone is in heating mode the blinds should always be open and when the zone is in cooling mode the blinds should only be open enough to meet the daylighting setpoint), but rather it is chosen for its simplicity and for its ease of comparison with the lookup controller.

Also for comparison, three two-week periods were simulated with full online MPC (using the same optimization configuration used in the lookup table creation, but with the actual disturbance values for each hour of the simulation, connected again through BCVTB): March 1-14, June 9-23 and December 17-31. A full-year online MPC simulation was not attempted because of the computation time requirements - each of the two week periods required 6-8 hours of run time, so a full year simulation would require roughly 150 to 200 hours (6-8 days). The three simulation periods were chosen to be representative, as they capture summer solstice, winter solstice and a shoulder season.

Figure 17 shows the control decisions over the year. As expected, the shading percentage and natural ventilation percentage are both higher in the later summer and early autumn, when the ambient temperatures and solar gains are both high.

But at this resolution it is difficult to see the differences between the base case controller and the lookup table controller. Figure 18 shows just the two-week period of March 1-14, for which a full online MPC comparison was also run. The lookup table control responses are very similar to the online MPC responses in all but a few times during this period of the simulation. For this period, the lookup table control (and the online MPC control) are generally calling for both more shading and more natural ventilation than used by the base case control. This is likely because there are relatively high solar gains during this period, and the base control considers only ambient temperature in its decision while the optimization-based lookup table also considers the solar gains.

Figure 17: Control decisions, lookup table control versus base case, annual

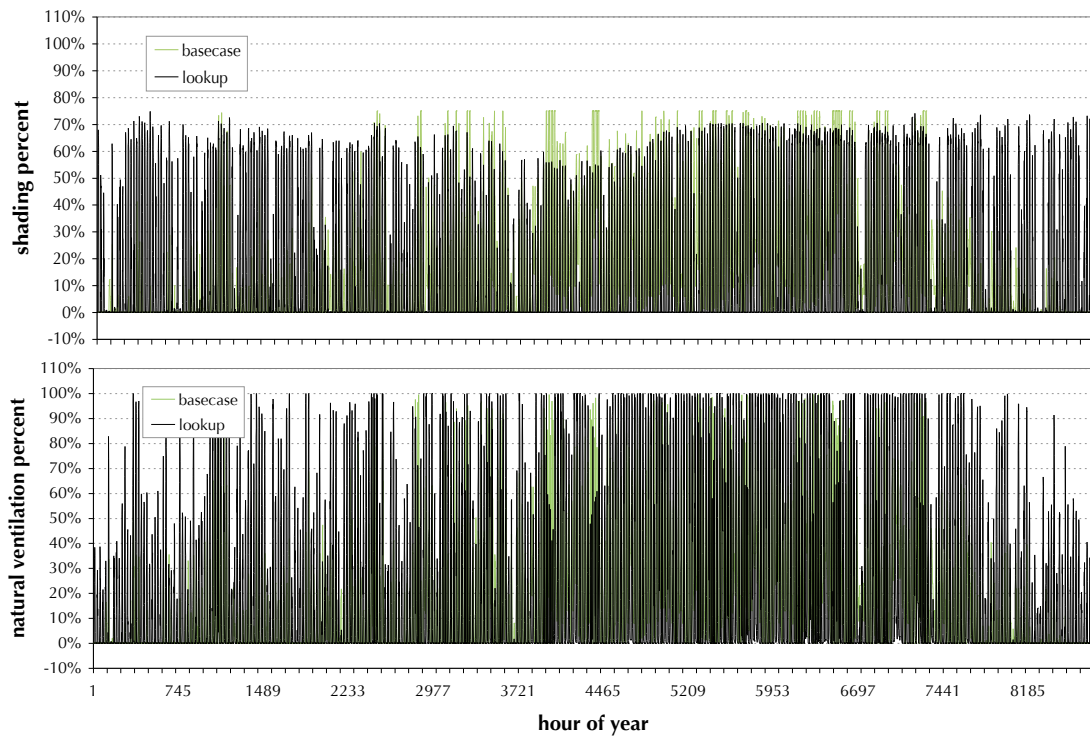
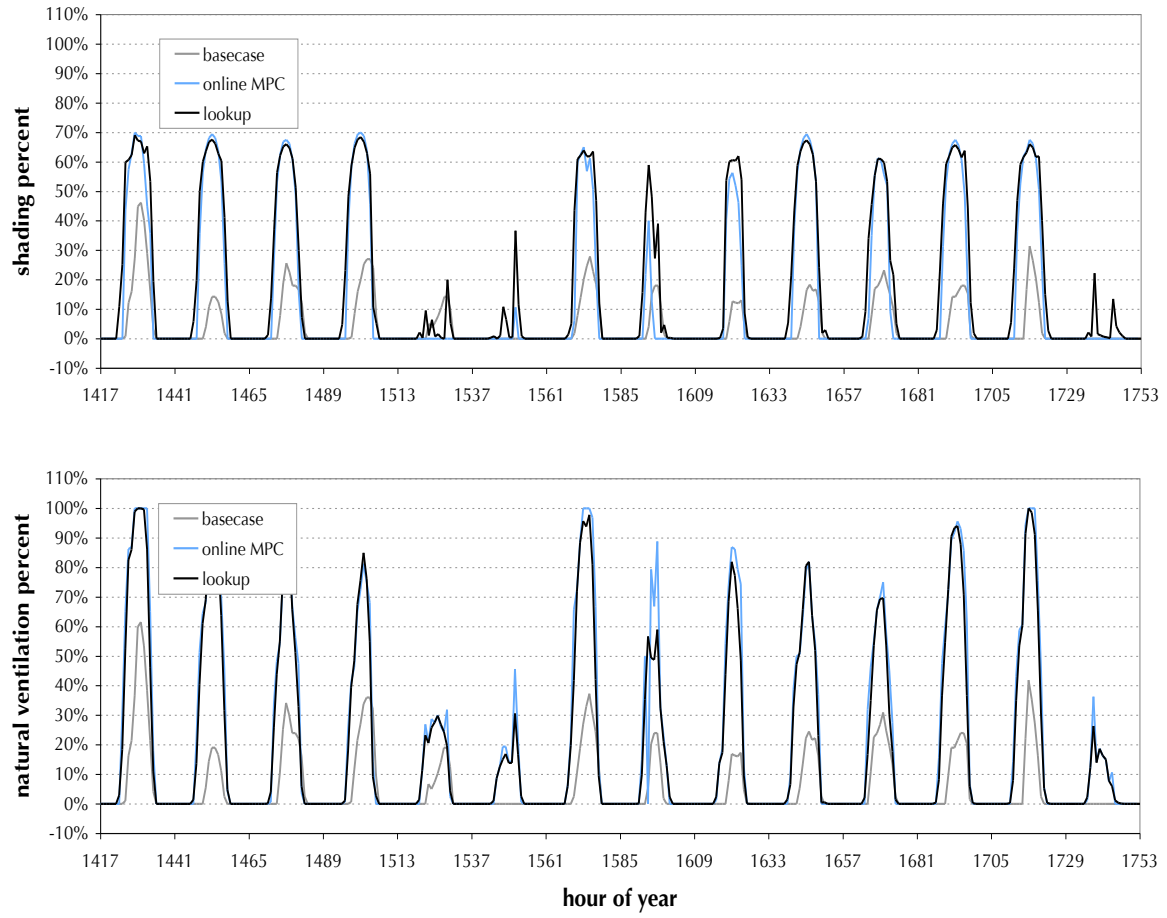


Figure 18: Control decisions, March 1-14



The scatter plots in Figure 19 compare the controllers' behaviours perhaps more clearly, with the spread in the lookup controller points being caused by its response to different solar conditions and different internal loads. Also note in Figure 19 the drop in natural ventilation percentage in the lookup case between 20°C and 22°C . However, given the model equations, we know that the optimal control should be at 100% natural ventilation until it drops to 0% at exactly the zone setpoint temperature (22°C). The reason that it does not behave this way is because it is interpolating linearly between the grid points at 20°C and 22°C . This is an illustration of one of the performance penalties in the lookup table approach.

Figure 19: Control decisions versus ambient temperature

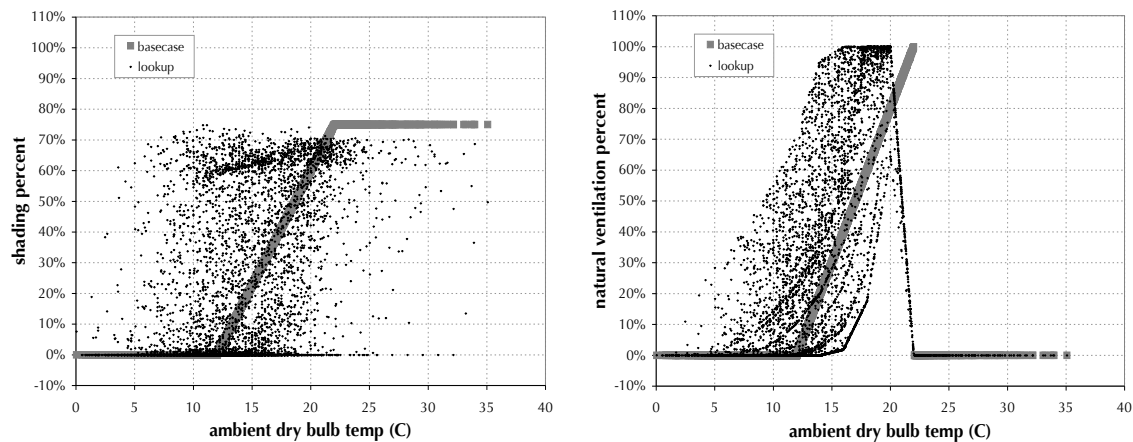


Figure 20 shows when and how the lookup table is outperforming the base case: primarily by using less cooling in the winter (afternoons) and less heating in the summer (start and end of work day). In the two weeks shown in Figure 21, for example, the lookup case is using much less cooling energy than the base case. The comparison with the online MPC suggests that the lookup table control could be saving more lighting energy on some days.

Figure 20: Hourly energy consumption, lookup table control vs base case, annual

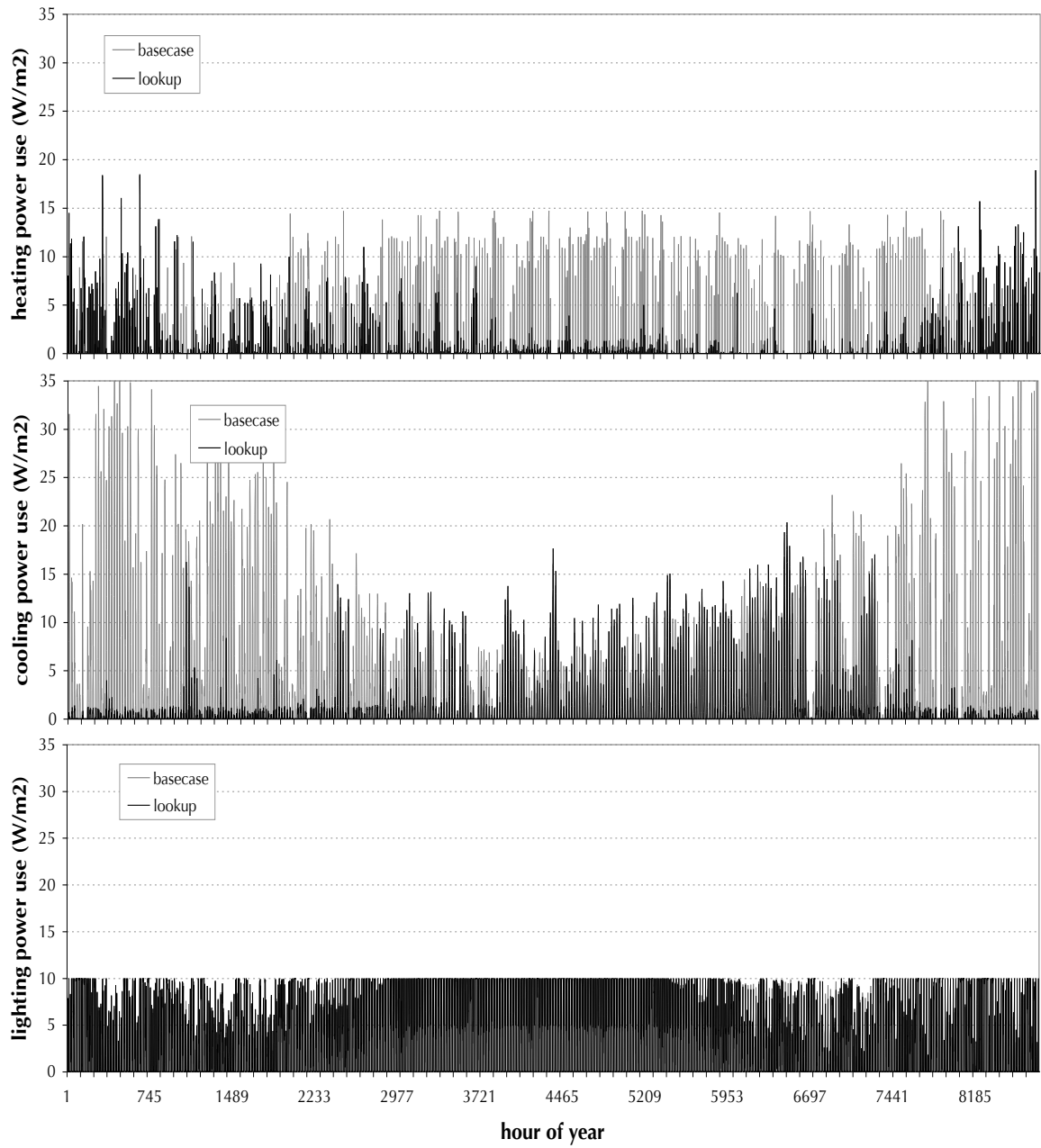


Figure 21: Energy consumption outputs, March 1-14

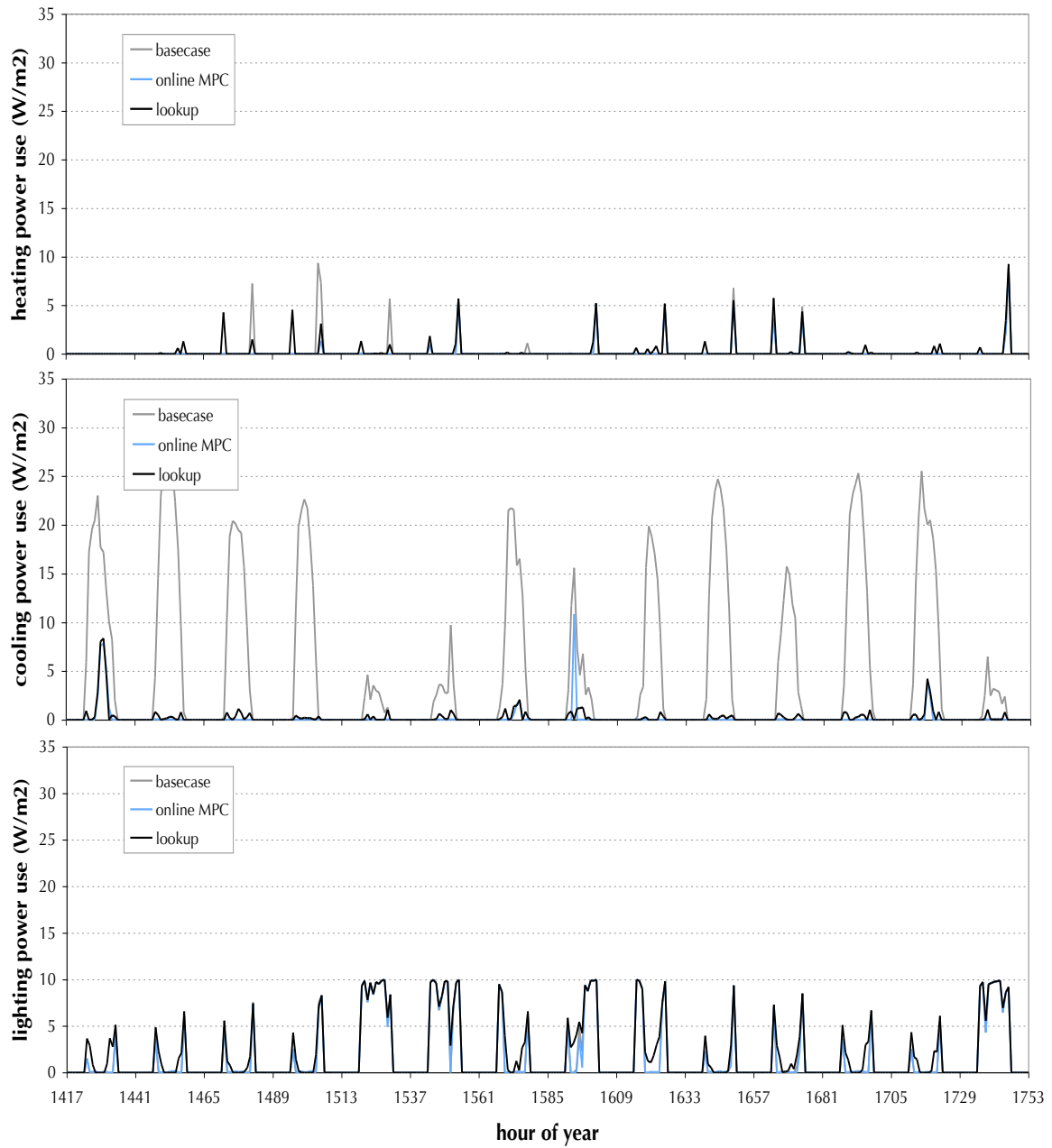


Figure 22 compares energy use of the base case to the lookup control over the year. The annual total energy consumption (heating + cooling + lighting) during occupied hours (8am-6pm) is 54.1 kWh/m² in the base case and 33.4 kWh/m² in the lookup table control case, an annual savings of 38.2%. Figure 23 shows the hourly energy savings versus the ambient temperature and direct solar gains. The lookup controller is often significantly outperforming the base case controller during periods when the ambient temperature is within economizer range and also when the solar loads are high. It does less well at low solar loads and when the temperature is just barely below the zone setpoint.

Figure 22: Total energy consumption outputs, lookup table control versus base case

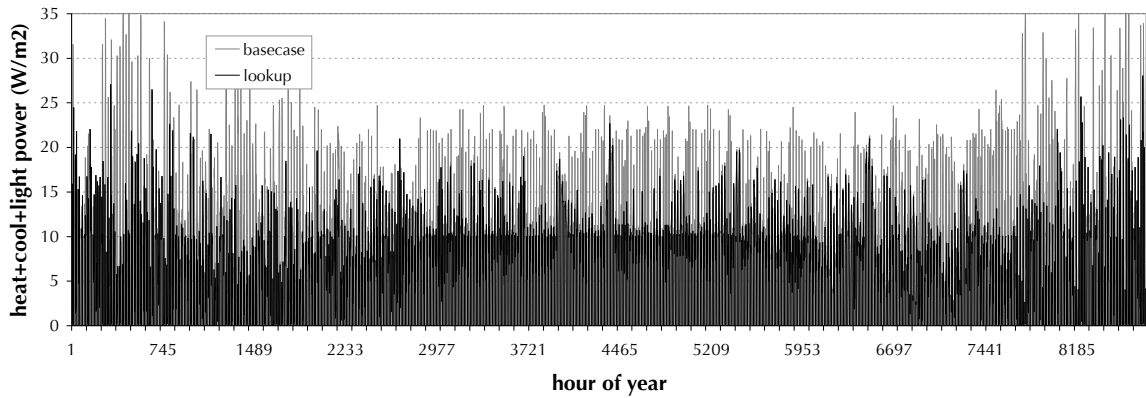
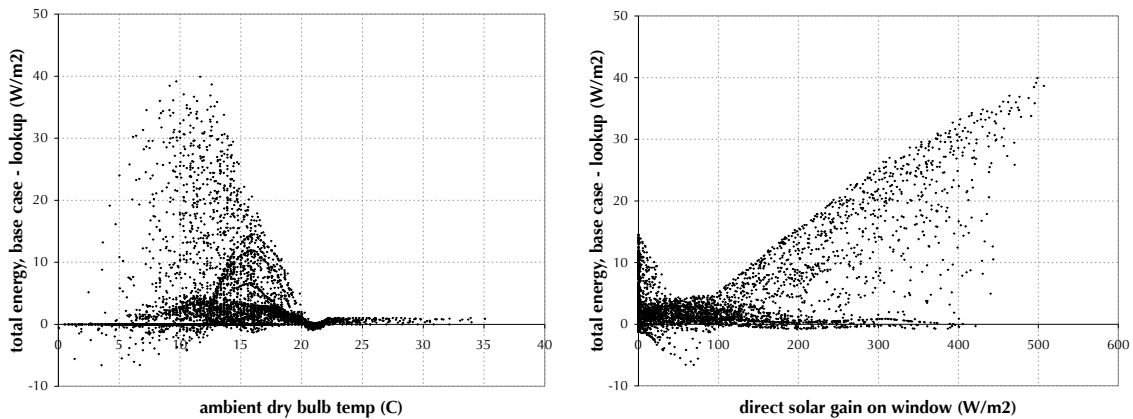


Figure 23: Lookup control energy savings over base case



The three 2-week periods of simulation with the online MPC provides a comparison from another perspective, showing where the lookup controller is not performing as well as an ideal controller. (In this case, the performance losses are entirely because of interpolation inaccuracies between grid points; in the remainder of the case studies the approximations from conditions parametrization or problem decomposition will add to the loss in performance.) Figure 25 shows the total energy consumption of the base case, lookup controller and online MPC for the three periods. Table 3 summarizes the results. With the online MPC demarcating the upper bound of savings potential, over the six test weeks the lookup controller with the configuration outlined above was able to capture 87.1% of the available savings over the base case control. Figure 24 shows how the lookup table controller is performing relative to the online MPC at different ambient temperature and direct solar gains conditions - most of the inaccuracies occur when the direct solar gain is between the grid points of 0 and 200 W/m^2 .

Table 3: Energy savings over base case for full MPC test periods

	lookup controller	MPC controller
March 1-14	64.1%	72.2%
June 9-23	18.1%	26.6%
December 17-31	62.4%	68.2%
Total for 6 weeks	49.6%	57.0%

Figure 24: Lookup control energy minus online MPC energy

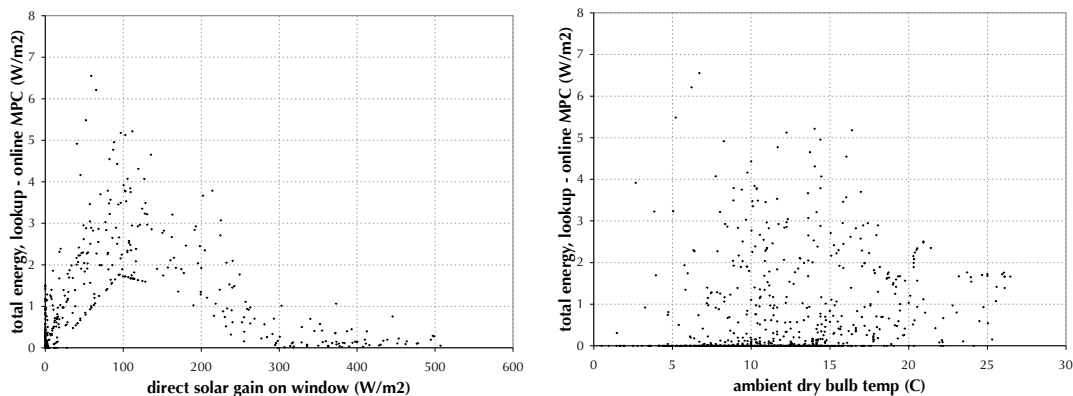
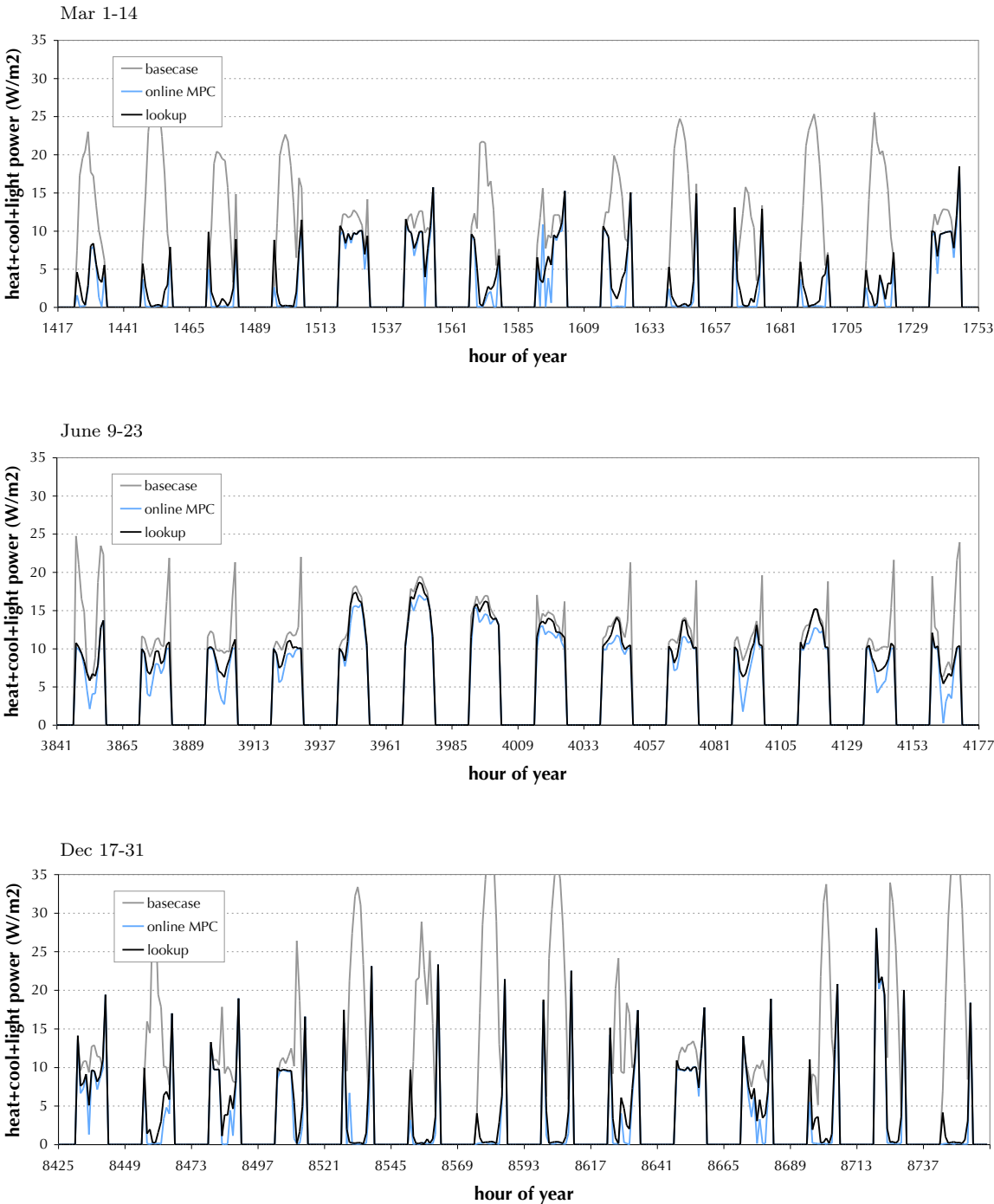


Figure 25: Total energy consumption outputs



4.2 Analysis: Effects of grid spacing on control performance

To test the effects of grid spacing on control performance, this simple case study was repeated with different granularities of grid for the direct and diffuse solar gains conditions. This cannot provide a definitive relationship between grid spacing and performance in general, since the relationship is case dependent. However, the analysis does give a sense of the effects, and points to possible areas for future research and development.

4.2.1 Controller performance relative to grid spacing

The three different conditions grids considered are shown in Table 4. Note that the diffuse and direct solar were decreased to a maximum of 600 W/m^2 in these grids (except for the case with solar spacing of 400, which required a max of 800 W/m^2 because of its coarser spacing), because the annual values for this site never exceeded these values. Lookup tables were calculated and annual simulations performed for each of them in the same way as shown above.

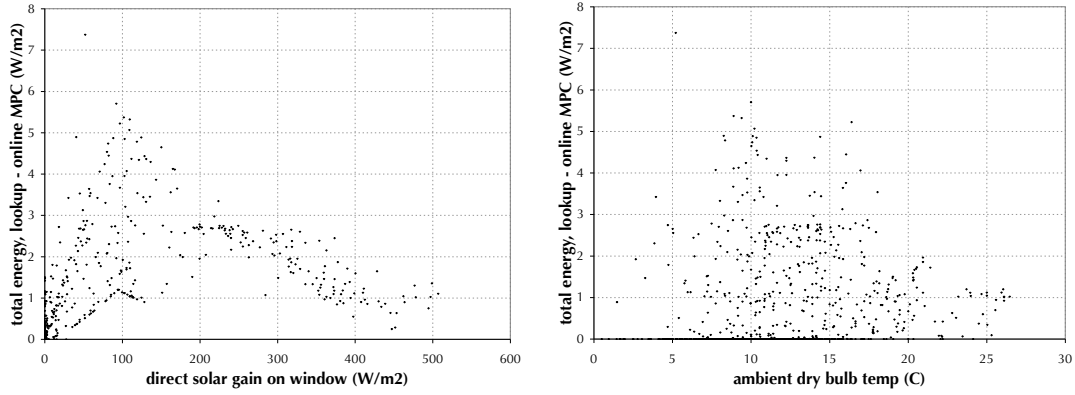
Table 4: Disturbances grids considered

	min	max	step size		
			test 1	test 2	test 3
ambient temp (C)	0	30	2	2	2
direct solar (W/m^2_{window})	0	600 or 800	400	200	100
diffuse solar (W/m^2_{window})	0	600 or 800	400	200	100
gains from people and plugs (W/m^2_{floor})	2	18	4	4	4
grid size (# points)			1152	2048	6272
processor-hours required			35	62	189

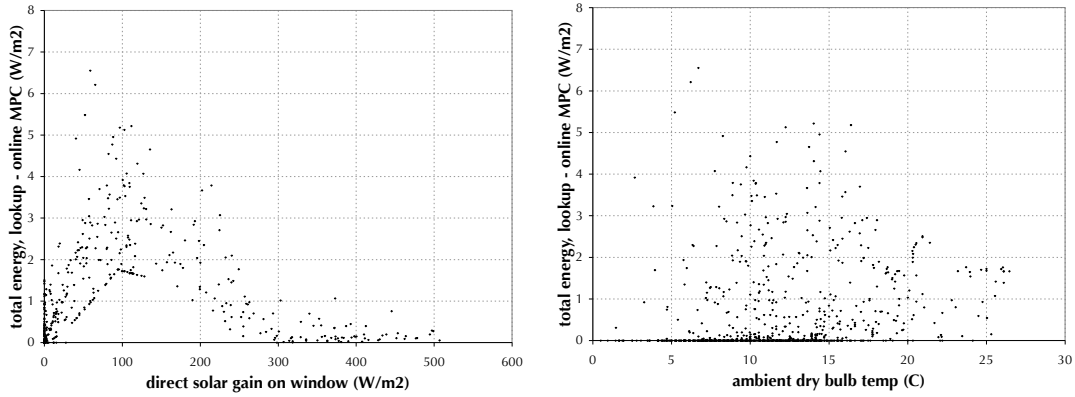
Figure 26 shows the hourly performance of each of the resulting lookup tables versus the full online MPC for the 6 weeks of available online MPC results. Note that the difference between the lookup control performance and the full MPC performance decreases with smaller grid spacing, and that the most notable decrease is when the solar gains are low. A graph of the energy savings versus grid spacing for these 6 weeks is shown in Figure 27, where the blue line shows the energy savings from online MPC for comparison. The percentage savings over the baseline are: test 1 (400 spacing) - 47.3% (capturing 83.0% of available savings); test 2 (200 spacing) - 49.6% (capturing 87.1% of available savings); and test 3 (100 spacing) - 53.5% (capturing 93.8% of available savings).

Figure 26: Lookup control energy minus online MPC energy

Direct and diffuse grid spacing = 400 W/m^2



Direct and diffuse grid spacing = 200 W/m^2



Direct and diffuse grid spacing = 100 W/m^2

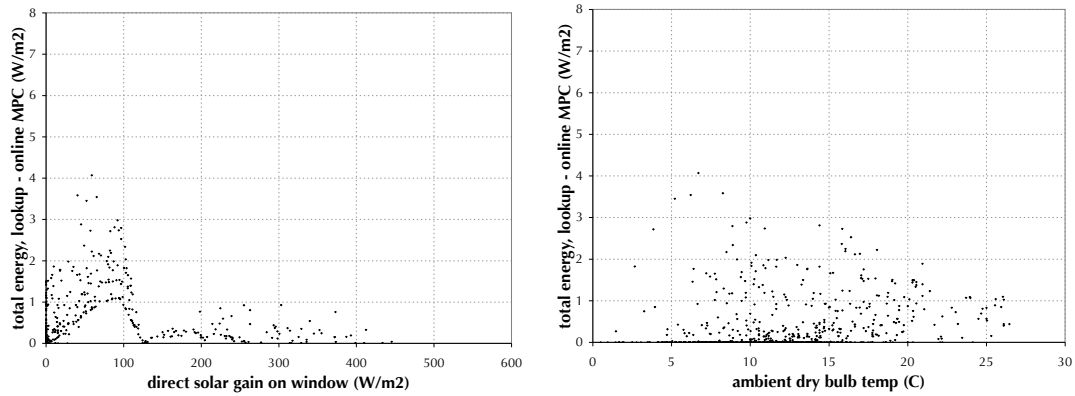
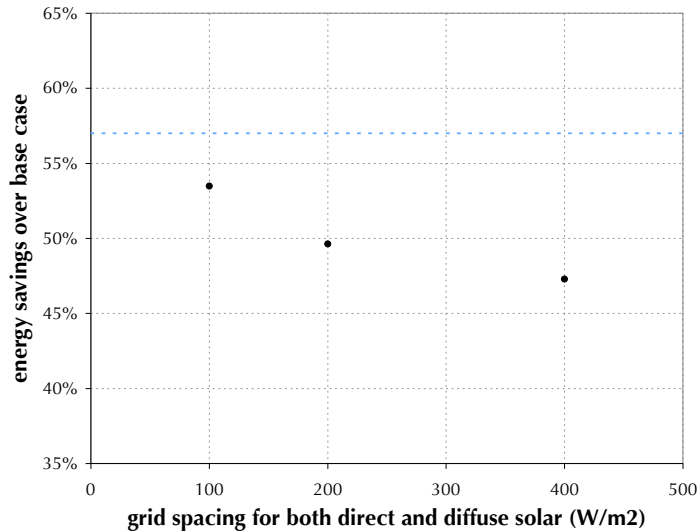


Figure 27: Energy savings vs direct and diffuse grid spacing



Note that as the grid spacing for direct and diffuse approaches zero, the energy savings should approach a savings level that is less than the 57% upper limit defined by the online MPC, because the grid spacing for ambient temperature and internal loads is being held at a non-zero constant. The three points shown do not suggest as simple of a relationship between grid spacing and performance as may have been hoped. In general, the relationship must certainly be monotonic and asymptotic to online MPC, but beyond that it is case dependent and difficult to characterize.

4.3 Discussion

This case study illustrates the essential methods of lookup table creation and use. A lookup table of optimal control solutions for a grid of conditions was created. This was then used with linear interpolation between grid points to act as a controller in an annual simulation. It was found to perform much better than a simple heuristic base case controller but not quite as well as an online MPC controller (for the six weeks of comparison, the lookup table control captured 87% of the savings that were captured by the MPC controller).

This case study also provides an example of the benefits of being able to visualize the control response over the conditions grid, in that it provides the user with an opportunity to test, debug and understand the controller's behavior.

Because this example is intended to illustrate the basic process of lookup table creation and use, and to illustrate ways of viewing and analyzing the results, it has been kept very simple. There was no prediction horizon in this example, only four disturbance variables, and no initial states or terminal constraints, so none of the techniques for conditions parametrization are required. Some practical problems are like this (e.g. central plant cogeneration dispatch without thermal storage or monthly demand charges but with fuel switching or similar complexities), but most problems of interest require approximations like those used in the next case study.

4.3.1 Process notes

As noted in the model description, a penalty function was added to avoid noise in the solution grid caused by the existence of multiple optimal solutions in parts of the conditions space (i.e. multiple control responses produce an objective function value of zero). The noise did not have a significant effect on controller performance, but it made the visualizations of the lookup table hard to interpret. The noise was encountered in an earlier iteration of this case study - Figure 28 shows some slices through this earlier lookup table. Further exploration of the grid led to the insight that parts of the conditions space had multiple solutions. Figure 29 shows that the objective function output is roughly zero when there is noise in Figure 28. The physical explanation of this is as follows: if the outdoor air temperature is less than the zone temperature and the thermal loads in the zone are balanced such that the the heating and cooling requirements are at zero, then a decrease in the shading percentage (increasing solar gains) alongside an increase in the natural ventilation percentage (increasing convective loss) of the appropriate ratio can keep the heating and cooling loads at zero. So to avoid the noise, a penalty function was added that only has an impact on the optimal value when multiple solutions exist, and which forces the optimizer to select the equivalent solution with the highest shading value.

4.3.2 For future consideration

As shown in Figure 26, most of the performance improvements with the smaller grid size were because of the addition of a point at 100 W/m^2 . The other added points had lesser impacts on the performance. This raises the question whether the performance to computation-time ratio might be improved through the use of an adaptive grid or through some other variety of sampling method. This is discussed further in other case studies and in the general discussion in Chapter 9.

Figure 28: An earlier iteration: optimal shading and natural ventilation levels

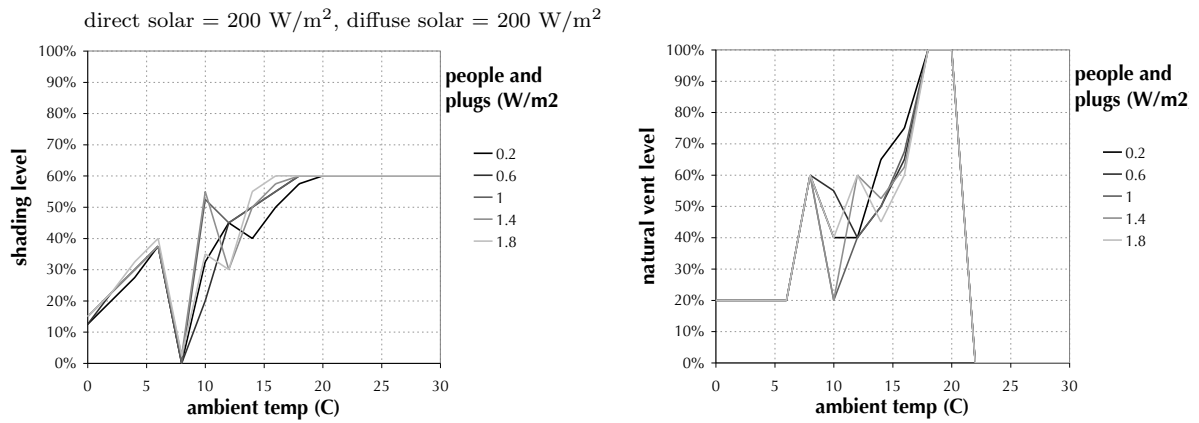
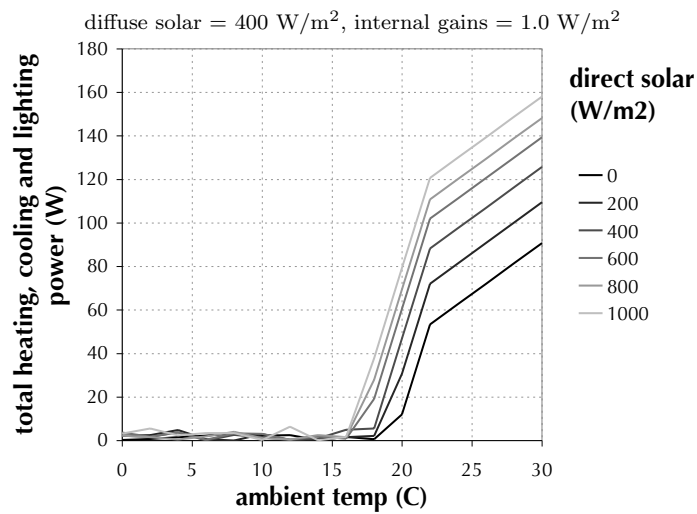


Figure 29: An earlier iteration: optimal total energy



5 Abstract Case Study #2: Slab Pre-Cooling, Conditions Parametrization

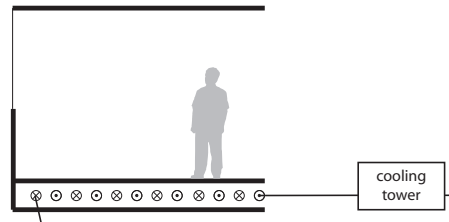
5.1 Case study

This case study illustrates the use of EnergyPlus within this procedure, as well as the use of conditions parametrization, both for disturbance predictions and for state initialization in the simulation tool.

5.1.1 Case description

The controller must determine the start time and charging length for overnight charging of a massive chilled slab with a cooling tower only, to minimize the average percentage people dissatisfied (PPD) during the following day with a floating slab and zone temperature. A 48-hour prediction horizon is used for the ambient temperature, and the control time-step is 24 hours long. The EnergyPlus example file RadLoTempHydrCoolTower.idf (from the example files folder in the EnergyPlus standard release) is used for this case study, along with the EnergyPlus TMY weather file for San Francisco.

Figure 30: Chilled slab case study illustration

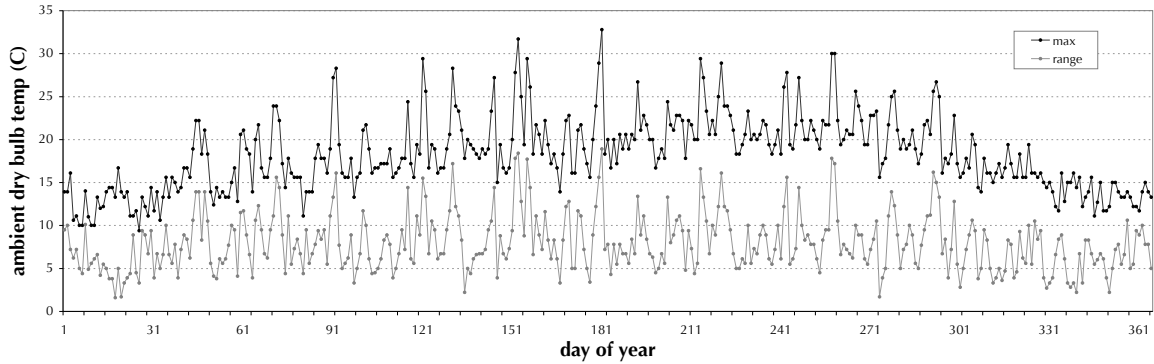


5.1.2 Model description

As per the description at the top of the RadLoTempHydrCoolTower.idf example file:

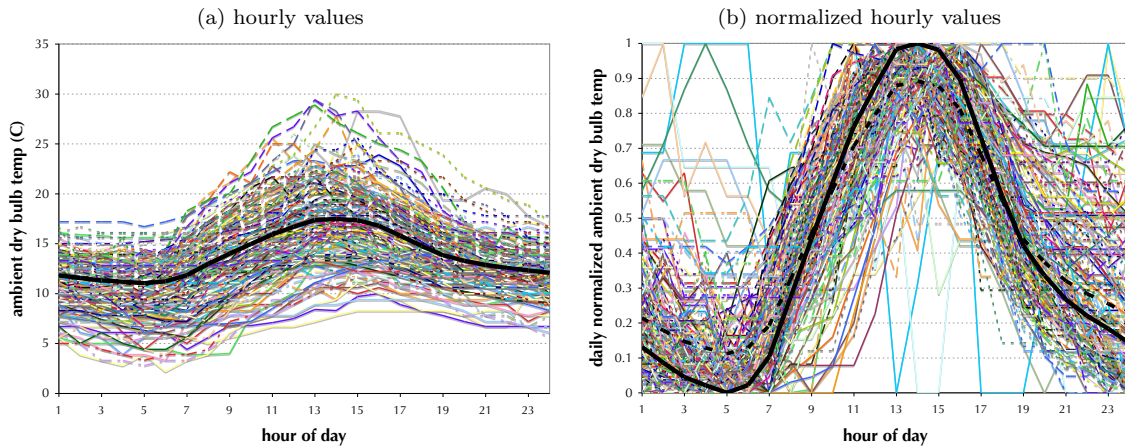
The building is one story tall and has three zones, the total floor area is 130.1 m² (1403 ft²), and the layout as shown in Figure 31. “The walls are 1 inch stucco over 4 inch common brick and gypboard. The roof is a built up roof with 1/2 inch stone over 3/8 inch felt over 1 inch dense insulation supported by 2 inch heavy weight concrete. The window is single pane 3mm clear. The window to wall ratio is approximately 0.07. The building is oriented due north.” The internal gains are as follows: West zone - 0W installed lighting, 2929W installed equipment, 3 occupants; North zone - 879W installed lighting, 2929W installed equipment, 4 occupants; East zone - 1464W installed lighting,

Figure 32: Ambient temperatures - daily maximum and range, SF TMY data



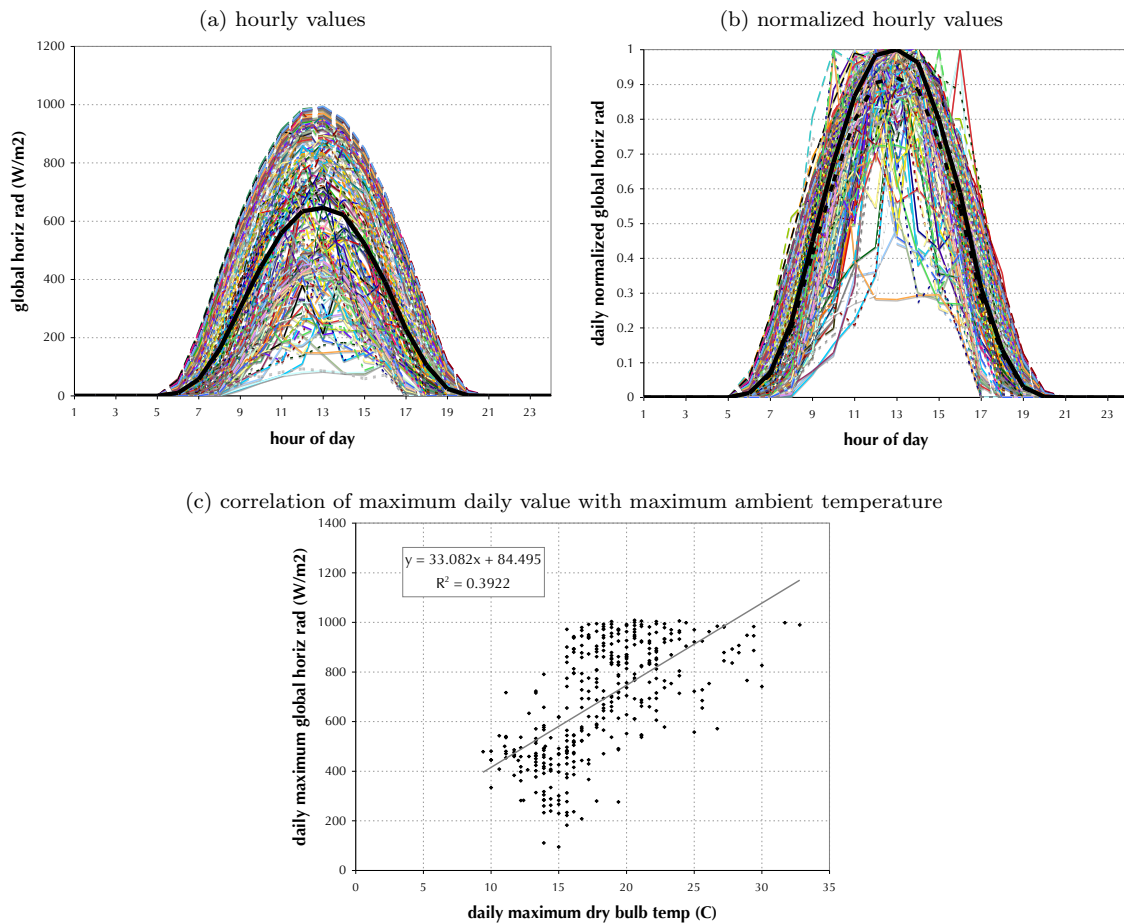
The EnergyPlus model, however, requires hourly inputs for a lot of weather variables, so a relationship must be derived to estimate these hourly values over the prediction horizon given the maximum temperature and daily temperature range. Figure 33(a) shows the daily temperature profiles from the TMY data, with each colored line representing one day. Figure 33(b) shows each of the the daily temperature profiles normalized to $[0,1]$ by their maximum and minimum. The dotted black line is the hourly average of these daily normalized curves. The solid black line is the normalization to $[0,1]$ of that dotted black line. The hourly values for this solid black line are used in the model to estimate the hourly temperatures over the day-ahead prediction given the predicted maximum and minimum temperatures.

Figure 33: Ambient temperatures parametrization, based on SF TMY data



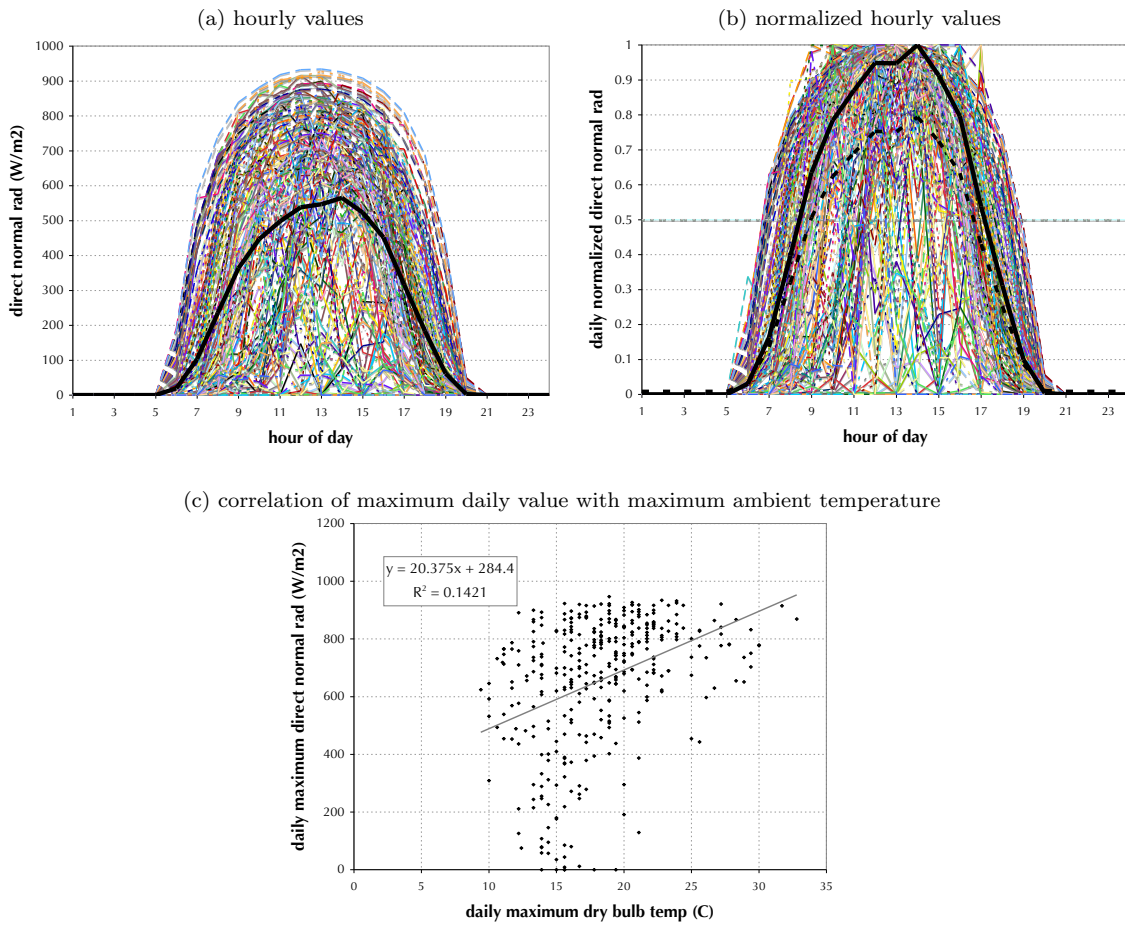
The other weather variables are normalized by their own daily minimum and maximum values in the same way. Their maximum and minimum values are then checked for correlations with the maximum and minimum dry bulb temperature. Some of these normalizations and the correlations of the maximums are shown in Figures 34 through 37. If the variable's daily maximum or minimum values show correspondence with the maximum or minimum temperature values (subfigure (c), with 'correspondence' meaning above some nominal R^2 value, which could be quite small and still be useful), then the normalized curves (subfigure (b)) and the correlations (subfigure (c)) are used in the model as described below. If no correspondence exists, then the average hourly values (subfigure (a)) are used in the model.

Figure 34: Global horizontal solar parametrization, based on SF TMY data



When the correlation is used, the process of mapping from the daily maximum temperature and daily temperature range conditions variables to the hourly values of the other variable (e.g. direct normal radiation) needed in the simulation is as follows: (1) given the maximum and minimum temperatures, the line of best fit equations (shown on the graphs in subfigure (c)) are used to estimate the maximum and minimum daily values of the variable under consideration; then (2) given these maximum and minimum values, the normalized curve in subfigure (b) is used to estimate the hourly values.

Figure 35: Direct normal solar parametrization, based on SF TMY data



Of the four variables shown in the graphs (the process was carried out for 10 weather variables in total, covering the most significant epw weather file columns), the first two shown (global horizontal radiation and direct normal radiation) used the correlation with ambient temperature, while the last two (diffuse horizontal radiation and relative humidity) just used the average hourly values (subfigure (a)).

Figure 36: Diffuse horizontal solar parametrization, based on SF TMY data

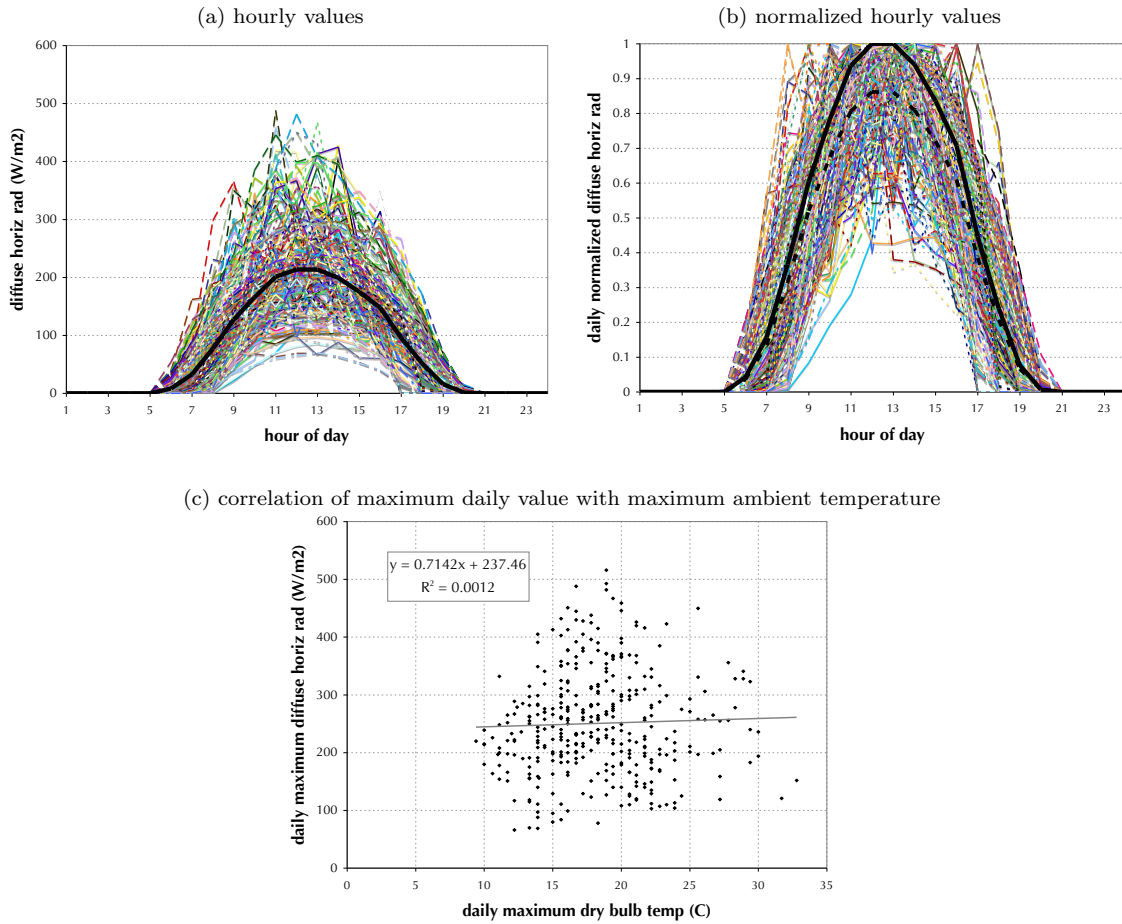
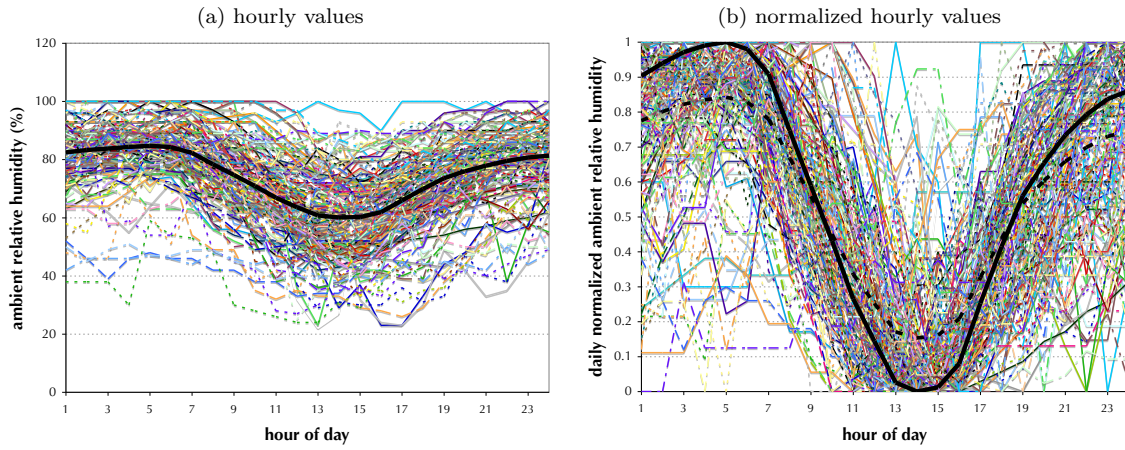
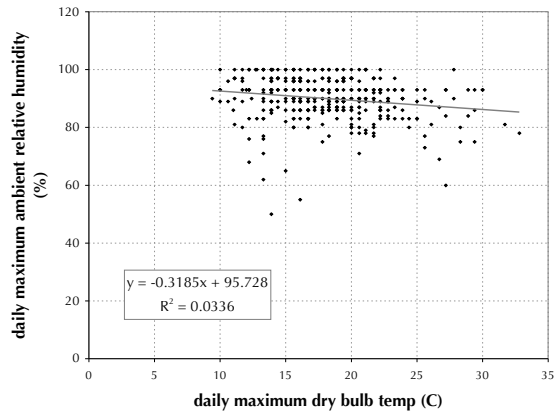


Figure 37: Relative humidity parametrization, based on SF TMY data



(c) correlation of maximum daily value with maximum ambient temperature



5.1.4 Conditions parametrization: initial states

As noted in the Methods chapter, many commonly-used building simulation tools do not allow the user to explicitly initialize state variables. EnergyPlus is one such tool. It was conceived for annual simulations, where state initialization is not particularly important. Instead of asking users to specify initial state values, it runs a warmup period wherein it repeats the first day of the simulation until a periodic steady state is approximately reached, and then carries on with the actual simulation from there. For MPC, however, this is problematic, and it is particularly problematic for offline optimization, as noted earlier.

One possible workaround is illustrated in this case study. The process is to set up an ‘initialization horizon’ with a few well-selected model or weather parameters and then determine the mapping between these parameters and the state values. This mapping is then used to set those parameter values in the simulation to produce the desired initial state. The approach is difficult to use, and while it is shown to work reasonably well here for two state variables, it seems that it is very hard to generalize or scale up beyond a handful of state variables.

In this case, an ‘initialization horizon’ of one day was configured with constant weather inputs and two simulation parameters that could be modified to affect the initial values of the two state variables of interest. The simulation parameters being modified are a pre-charge length (using the same slab cooling mechanism used for the overnight charging in the rest of the simulation) and a fifteen-minute-long internal gain scheduled for 8pm (using the EnergyPlus object “OtherEquipment”). The two state variables that must be initialized are the floor surface temperature and the zone temperature, both at 9pm on the night before the first charging period to be optimized. The pre-charge length parameter primarily effects the floor temperature, but it also has some effect on the zone temperature, as shown in Figure 38. Likewise, the fifteen minute internal heat gain (called the ‘heat burst’ in the graphs) primarily effects the zone temperature but also has some effect on the floor temperature (Figure 39). So the mapping between the modifiable parameters and the states of interest is a complex 2D-2D mapping. In the java implementation of this mapping, the relationships shown in Figures 38 and 39 where used with an initial guess of one of the state values and the output of one graph being used as the input to the other, repeated until the state values converged within a tolerance.

Figure 38: Floor temperature as a function of pre-charge length and heat burst

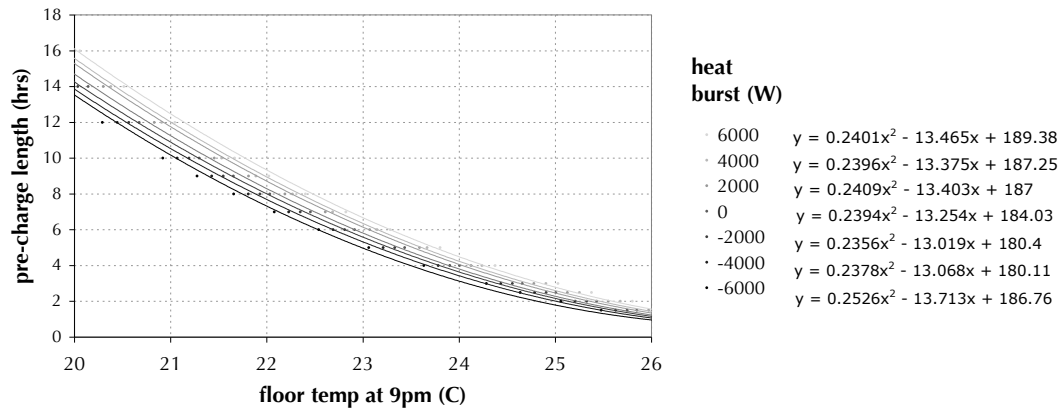
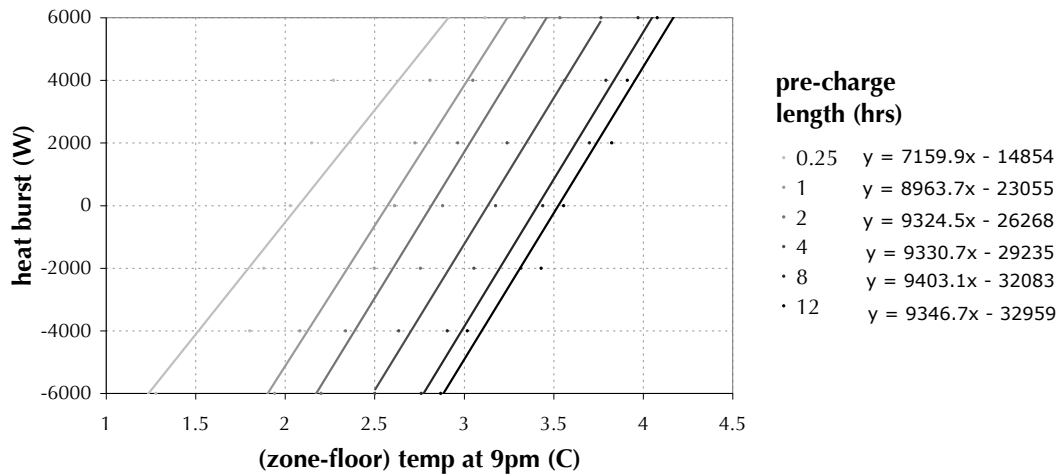


Figure 39: Zone temp minus floor temp, as a function of heat burst and pre-charge length



5.1.5 Optimization configuration

The charge length was constrained between 0 and 10 hours, and the start time between 0 and 10 hours after 9pm. The conditions grid is shown in Table 5. The Hookes-Jeeves algorithm in GenOpt was used, with 2 step size reductions. The 2880 optimizations required approximately 275 processor-hours (11.5 processor-days). The computations were carried out on 5 virtual machines on the Amazon Elastic Compute Cloud.

Table 5: Conditions grid for the massive slab cooling case study

	min	max	step size
floor temp at 9pm (C)	20	26	2
zone temp at 9pm (C)	2.0	3.5	0.5
maximum ambient temperature, day 1 (C)	16	32	4
range in ambient temperature over day, day 1 (C)	4	16	4
maximum ambient temperature, day 2 (C)	16	32	8
range in ambient temperature over day, day 2 (C)	4	16	6

5.1.6 Lookup table results

Figure 40 shows the charge length (vertical axis) as a function of the next day’s maximum ambient temperature (horizontal axis) for different values of the next day’s temperature range (different greyscale lines, darker is larger) and for different values of the initial zone and slab temperatures. In general, when the conditions are such that more pre-cooling is required, then the charge length is longer. The optimal charge length is found to be zero hours when both the day-ahead predictions of ambient temperature and the initial zone temperatures are low, ten hours when predicted ambient temperatures and initial zone temperatures are high, and it varies roughly linearly between zero and ten over a band of conditions that lies in between these two groups. The same results are shown on the right in Figure 41, with the tonal gradient showing the control signal (black when charge length is zero, white when it is ten hours). On the left of Figure 41 is the optimal start time (black is the very start of the night, white is the last available charging time), which shows less coherent of a pattern. The model’s PPD outputs are much less sensitive to the start time than to the charge length, which could explain some of the variation in those results. Figure 42 shows the optimal PPD versus the next day temperatures.

Figure 40: Optimal charge length

$$T_{maxDay2} = 16, T_{rangeDay2} = 4$$

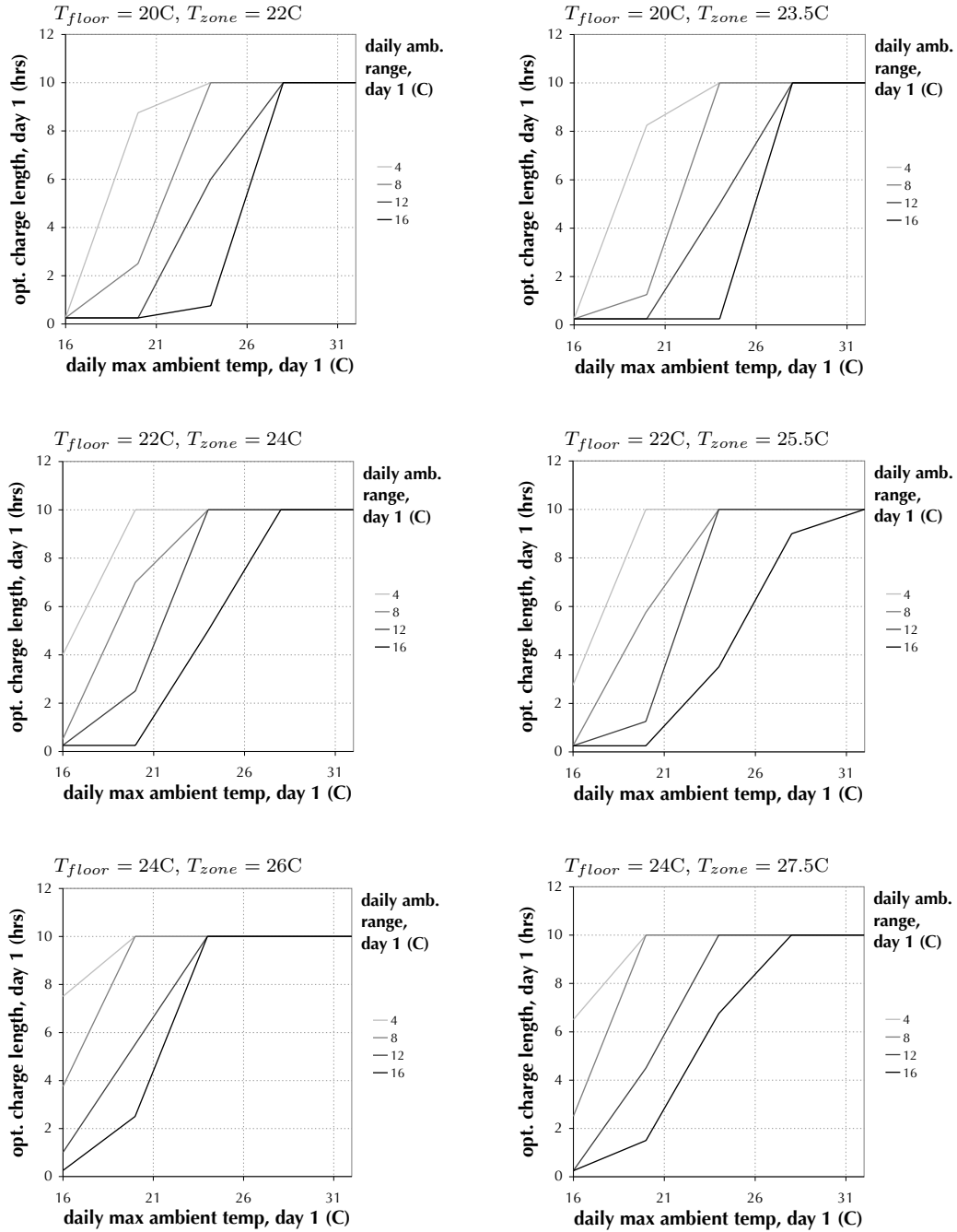


Figure 41: Surface plots of optimal start time and charging length

$$T_{maxDay2} = 16, T_{rangeDay2} = 4$$

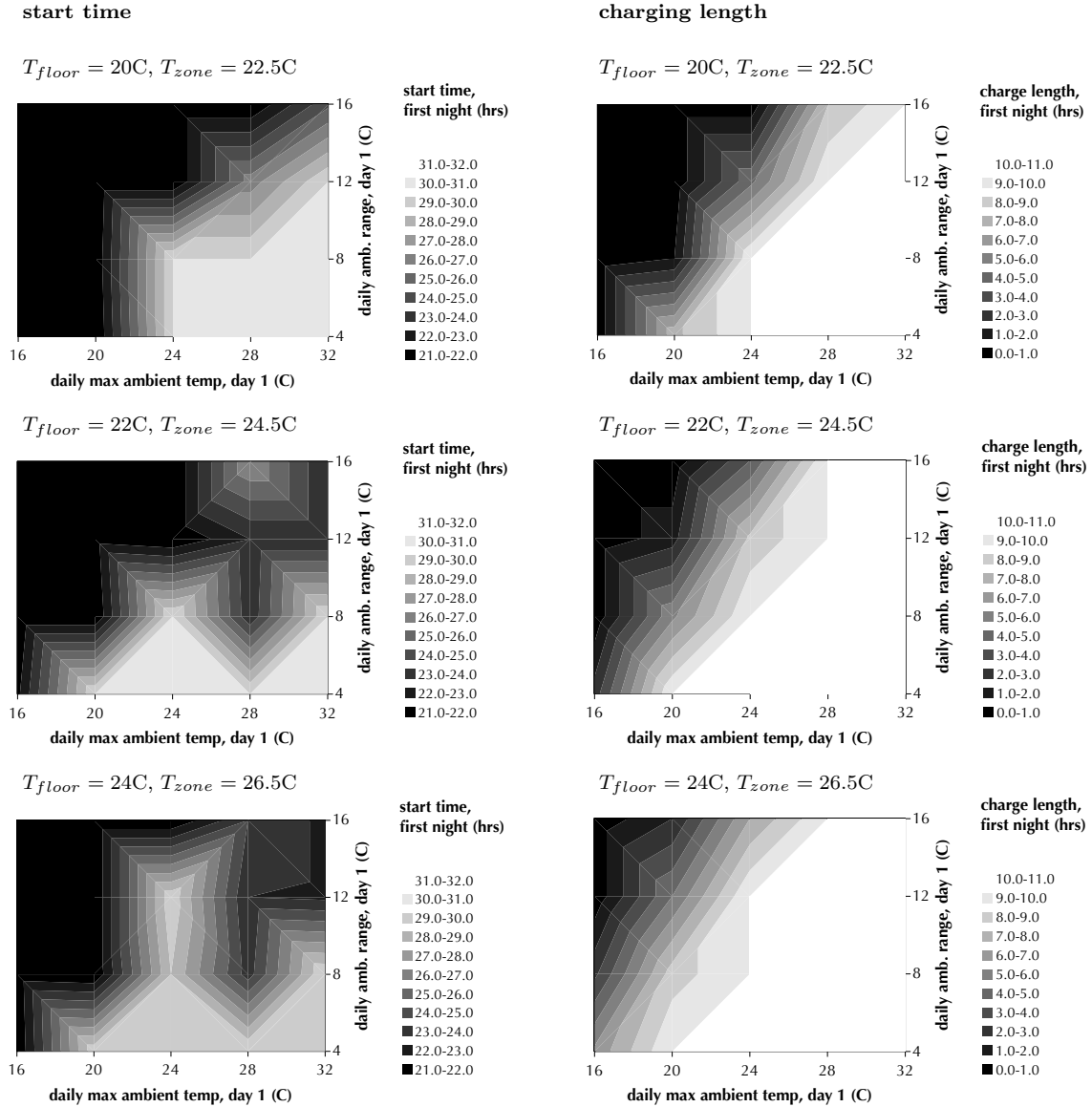
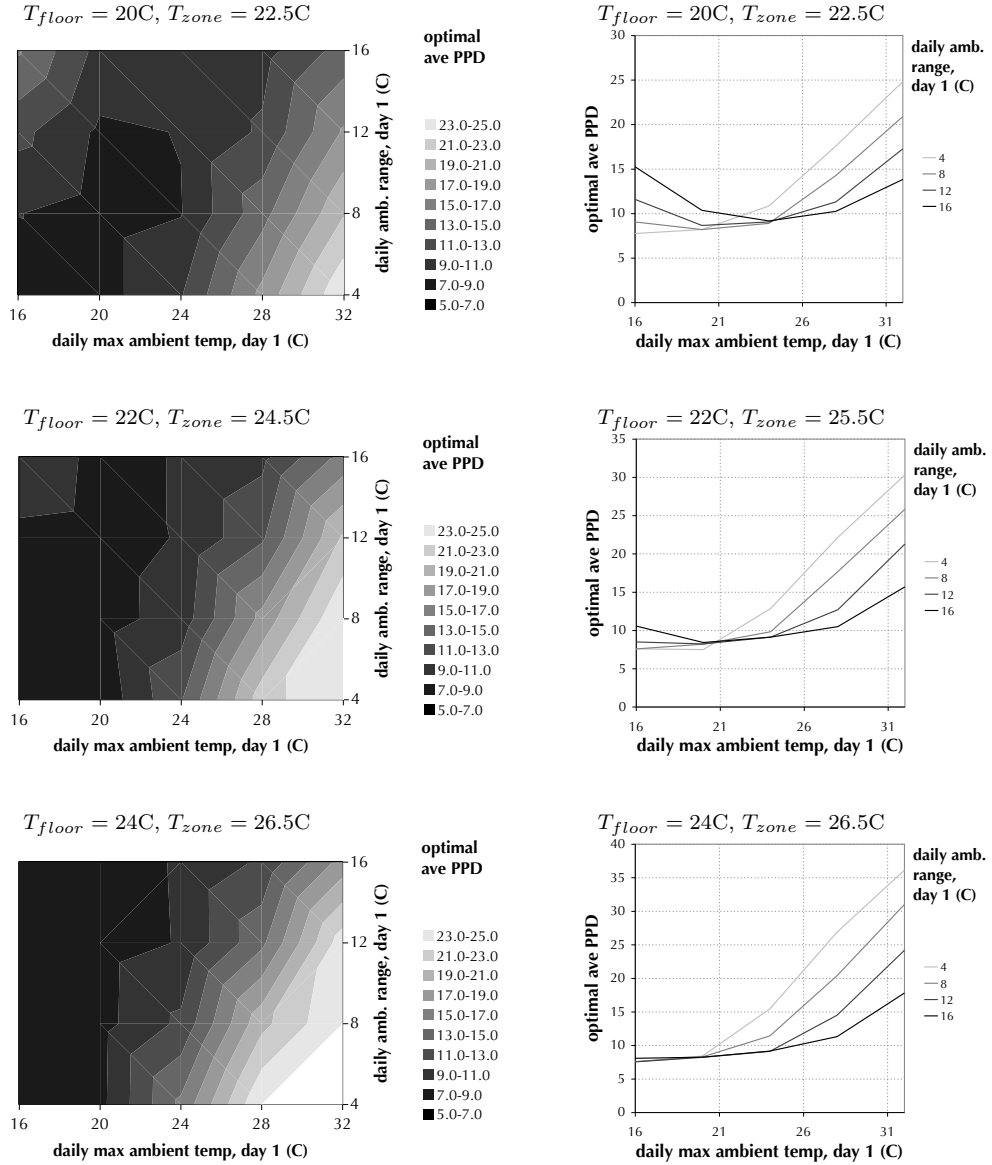


Figure 42: Optimal average PPD

$$T_{maxDay2} = 16, T_{rangeDay2} = 4$$



5.1.7 Annual simulations

Annual simulations were run to test the performance of the lookup table control versus two different base cases. The annual disturbances, derived from the TMY file for San Francisco, are shown in Figure 43. Figures 44 and 45 show the control decisions for the lookup controller and the base cases, and Figure 46 shows the resulting state values over the simulation length. “Basecase1” does not use any day-ahead prediction and its charge length varies monthly. “Basecase2” was created after learning from the lookup table control outputs, and it represents a very good base case. The comfort results are shown in Figures 47 and 48 and summarized in Table 6.

Figure 43: Disturbances, annual

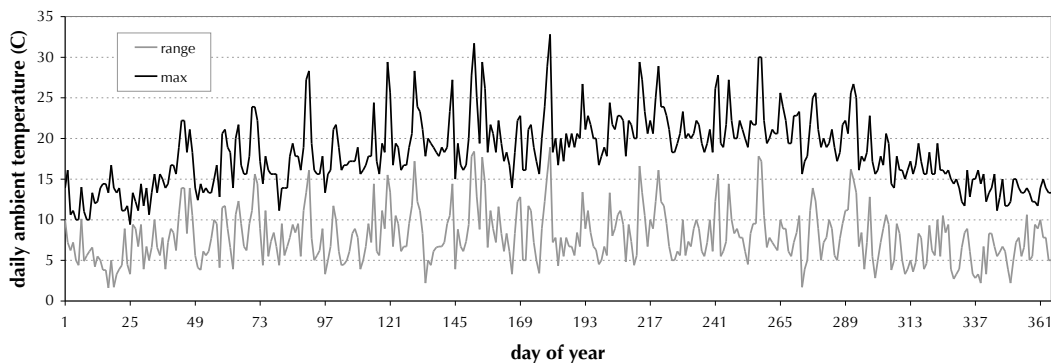


Figure 44: Control decisions versus next day max ambient temperature

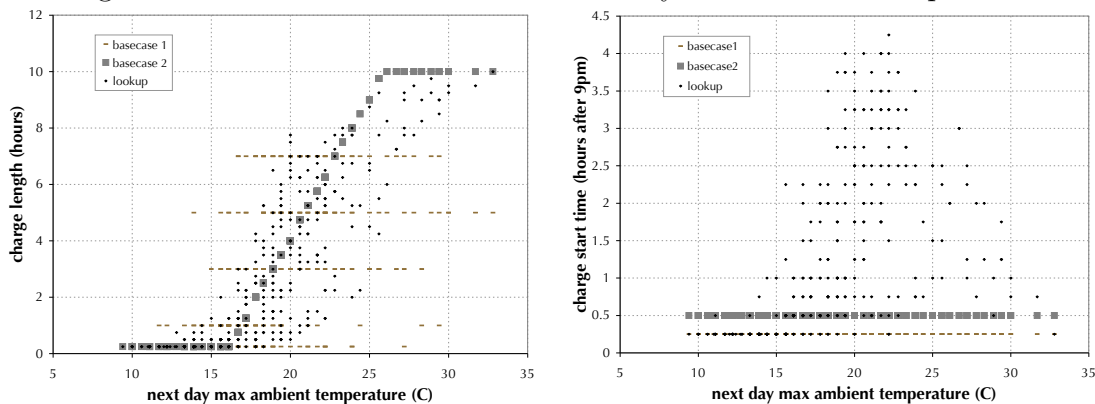


Figure 45: Control values, summer - top, June only - bottom

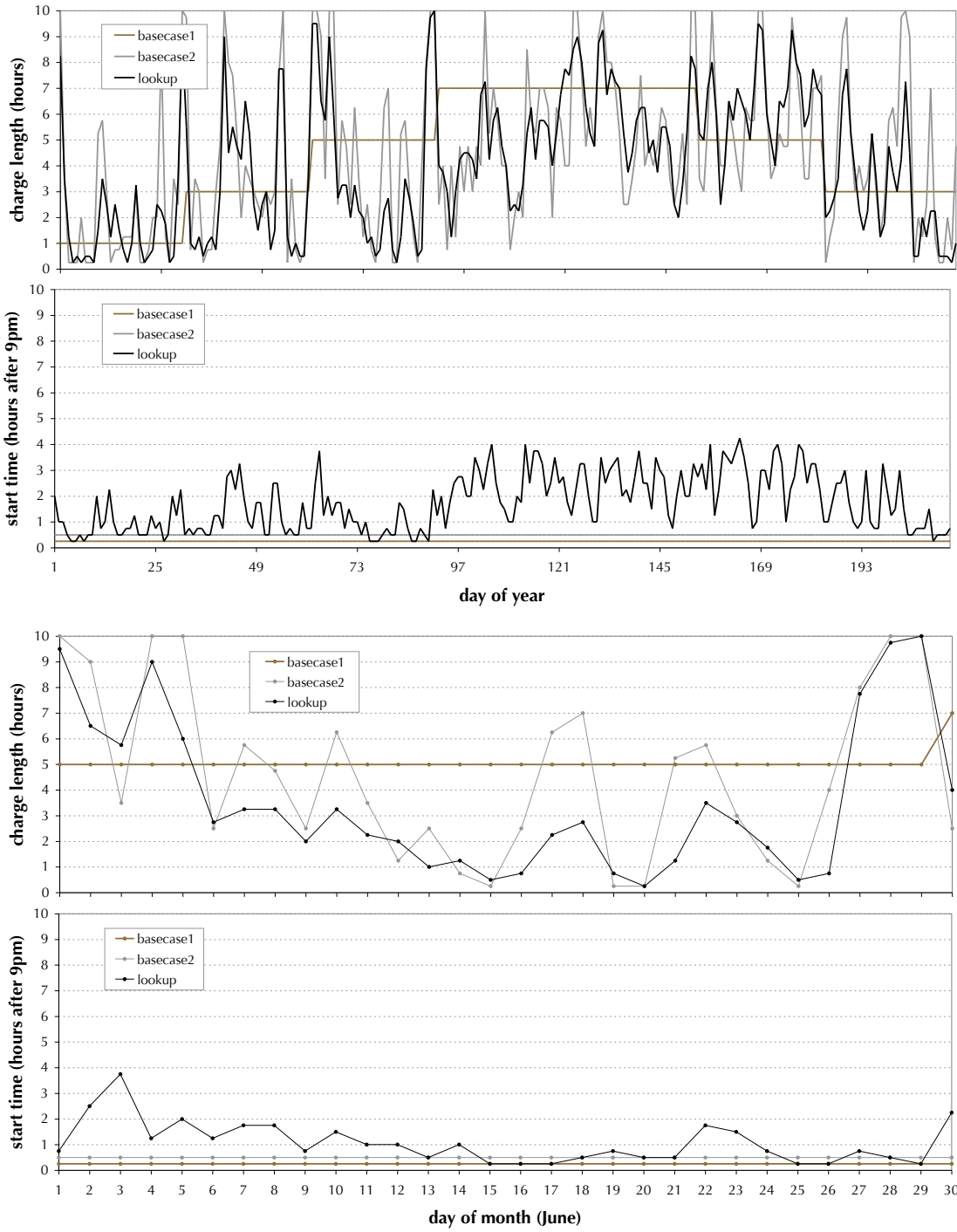


Figure 46: State values, summer - top, June only - bottom

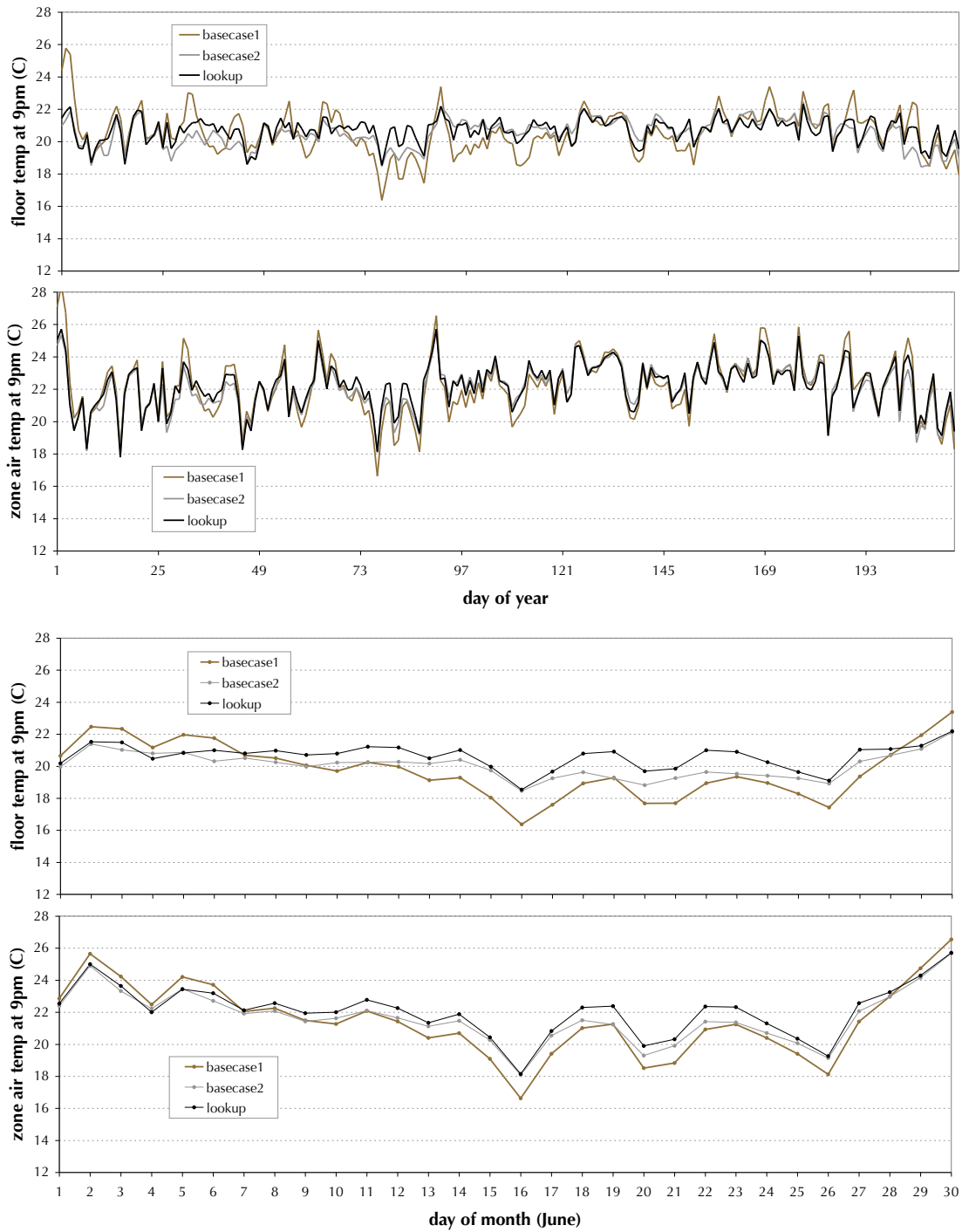


Figure 47: PPD values, summer

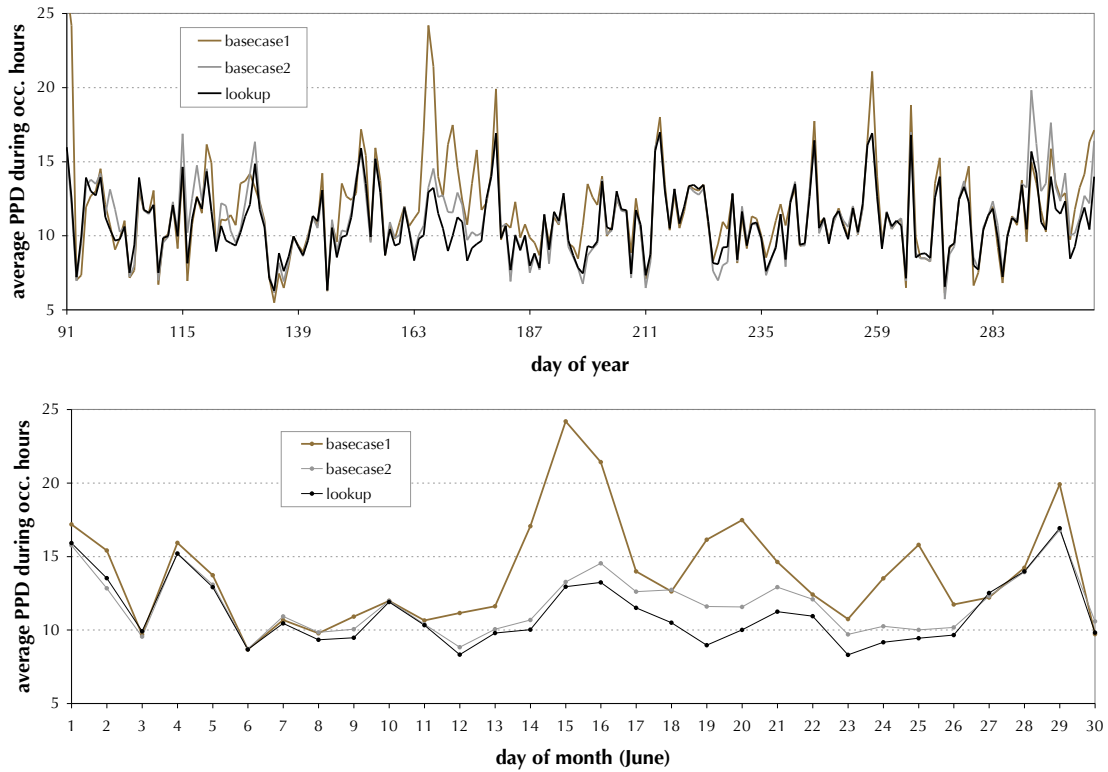


Figure 48: Control performance versus next day max ambient temperature

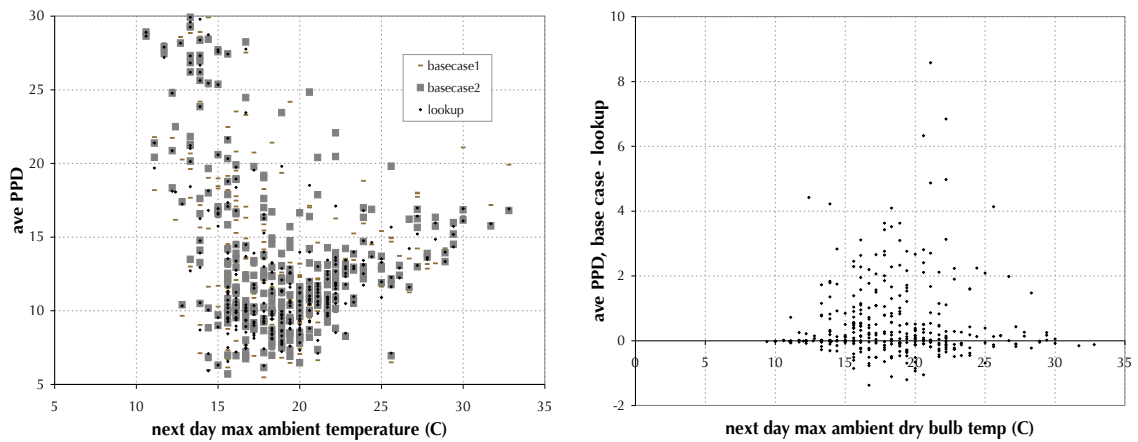


Table 6: Results for the massive slab cooling case study: basecase vs lookup

	ave PPD summer
basecase1	11.71
basecase2	11.12
lookup	10.84

Note that the minimum possible PPD is 5.0 (from the equations that define PPD, based on empirical data). “Basecase2”, which was derived after learning from the lookup table visualizations, performs much better than the original “Basecase1”, but the lookup table control performs still better. The Analysis section below compares this performance with online MPC.

5.2 Analysis: Effects of conditions parametrization on performance

5.2.1 Full online MPC configurations for comparison

Annual simulations were run to test the performance of the lookup table control versus three different online MPC configurations designed to tease apart performance losses due to grid spacing, disturbance parametrization, and initial state approximation. The three online MPC cases are defined as follows: “MPC1” uses the same 6 conditions variables as the lookup controller, thus keeping the disturbance and initial state parametrizations, but runs online instead of interpolating and so defines how good the lookup table control could do if the grid spacing was zero; “MPC2” uses the actual hourly weather predictions rather than the parametrized approximations, and thus shows the performance gains that could not be captured because of the use of disturbance parametrizations; and “MPC3” uses a 7-day initialization horizon with historical weather and control values, and thus allows for an estimate of the performance loss due to initial state approximation.

5.2.2 Results

The control decisions of the three MPC cases, the lookup controller and the base case are shown in Figure 49. The resulting state values are shown in Figure 50, and the objective function outputs are shown in Figure 51. The performance results are summarized in Table 7.

Figure 49: Control values with MPC, summer

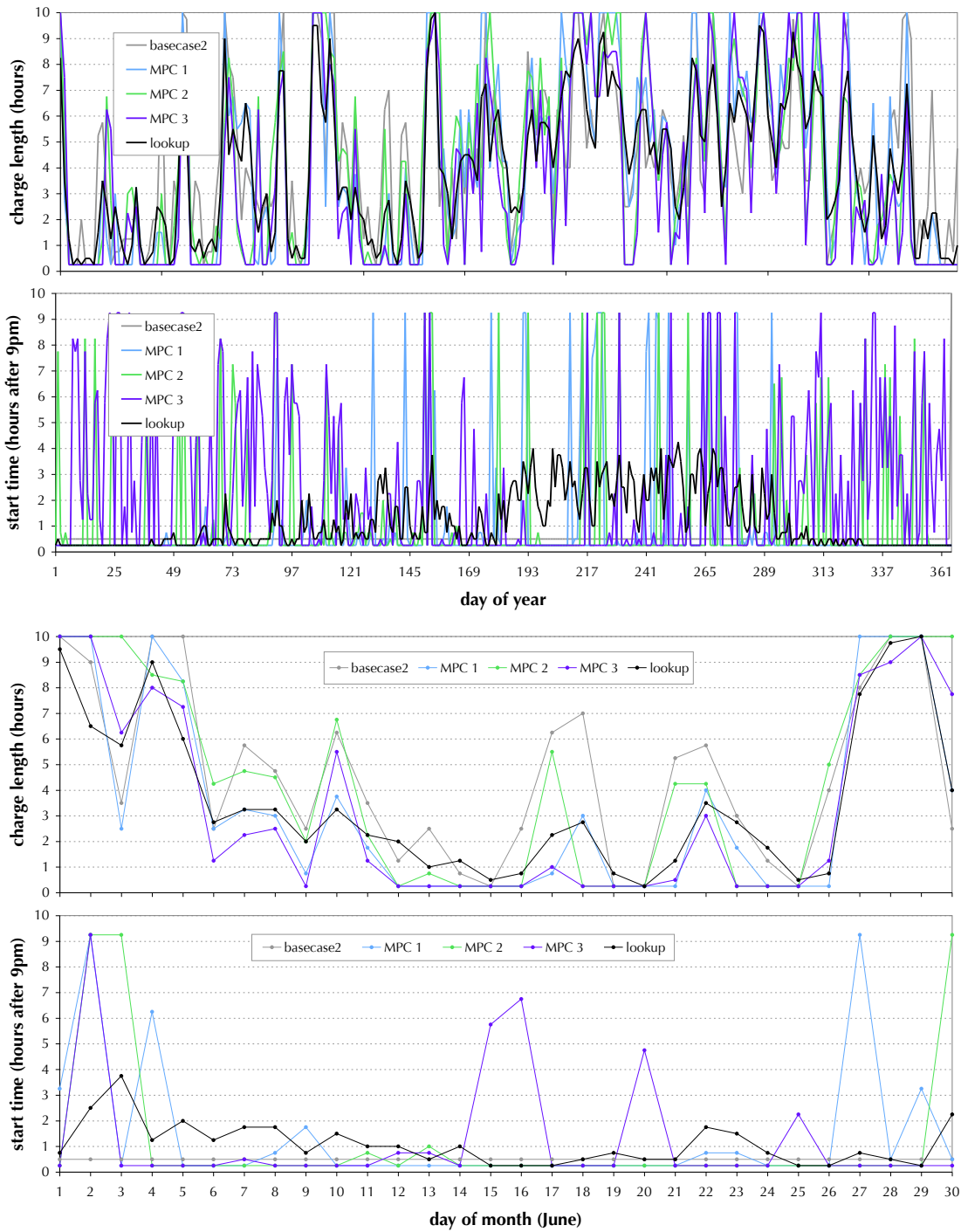


Figure 50: State values with MPC, summer

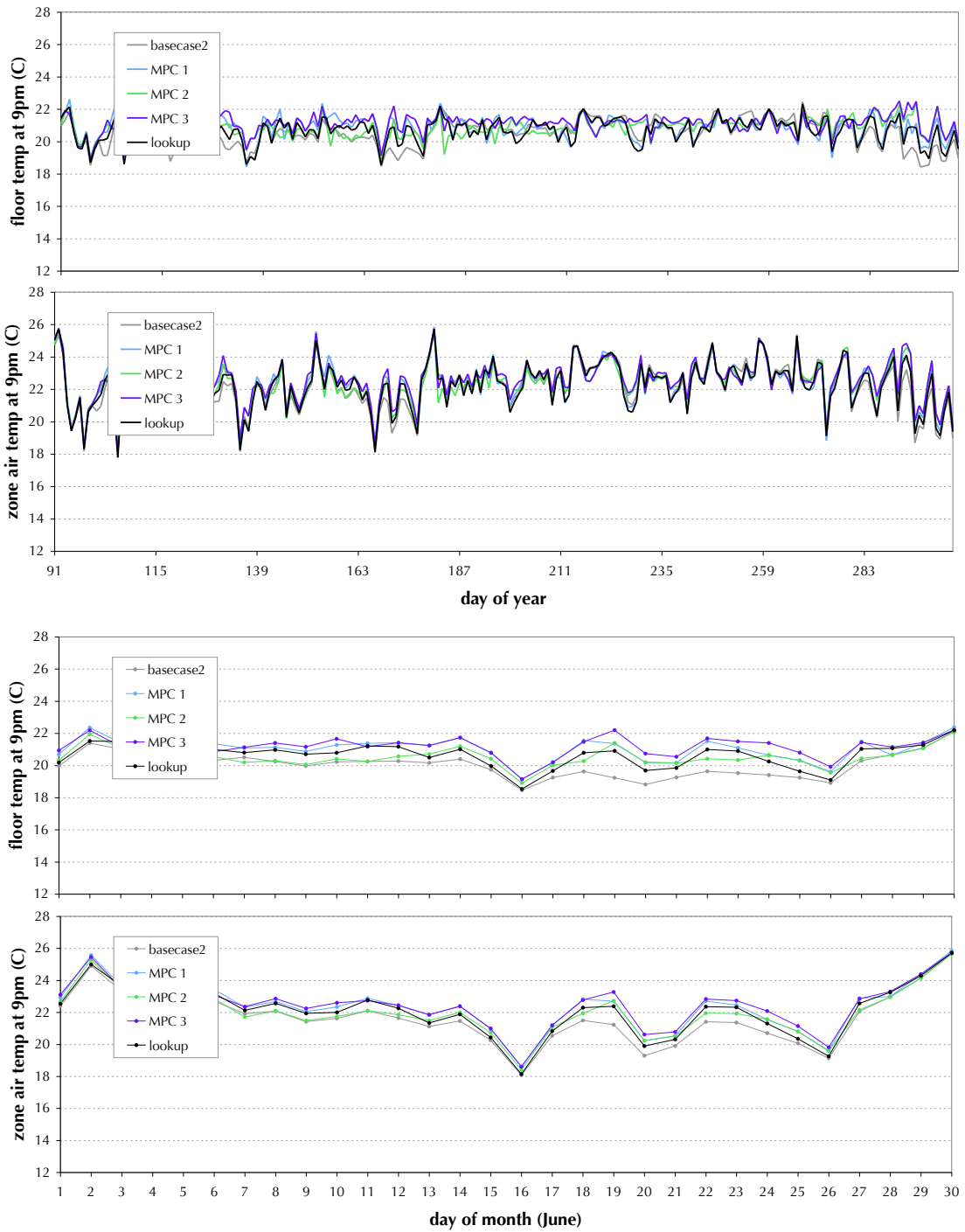


Figure 51: PPD values with MPC, summer

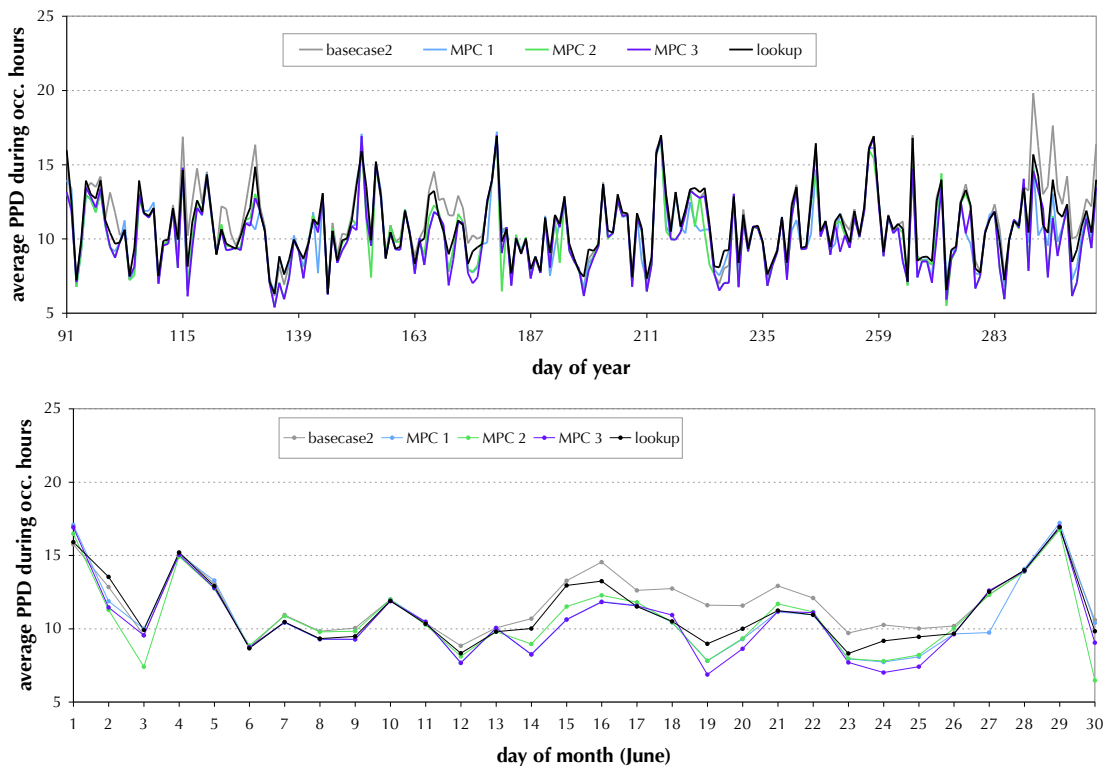


Table 7: Results for the massive slab cooling case study: lookup vs MPC

	ave PPD summer
basecase1	11.71
basecase2	11.12
lookup	10.84
mpc1	10.33
mpc2	10.26
mpc3	10.24

Defining the performance improvement potential as the difference between “MPC3” and “basecase1”, the heuristic control rule (“basecase2”) derived from looking at the lookup table slices was able to capture 40%, and the lookup table control (with its

relatively coarse grid spacing) was able to capture 59% of the available performance improvement. With a finer grid, the lookup table control could capture no more than 94% of the savings (as determined by “MPC1”), with the remaining 6% being lost through the parametrizations, most of this because of the disturbance approximations rather than the initial states approximations.

5.3 Discussion

In hindsight, average PPD is perhaps not the most appropriate objective function metric for use in this case study. The maximum hourly PPD over the day or the number of hours above a certain threshold would likely be better to use for future work in this application area. They would be both more representative of what the desired outcome, and they would likely also show greater differences between the various cases described above.

This case study demonstrated the use of EnergyPlus with this approach. The most difficult part of doing so is in dealing with the initial states. In this case a work-around was devised that had little impact on the controller performance, and which may be used for similar problems. However, it is unlikely that such work-arounds can be developed for cases with many more initial states under consideration.

The case study also tested the use of conditions parametrization over a prediction horizon, and presented a generic method for doing so. In this case, the performance losses due to the parametrizations were small. The performance impacts are very much case dependent, so general conclusions are hard to draw from this single case, but it does suggest that the losses in performance due to conditions parametrization are not so high to consider the approach as infeasible, and that it should be considered in further case studies and applications.

6 Abstract Case Study #3: Cogeneration Dispatch, Problem Decomposition

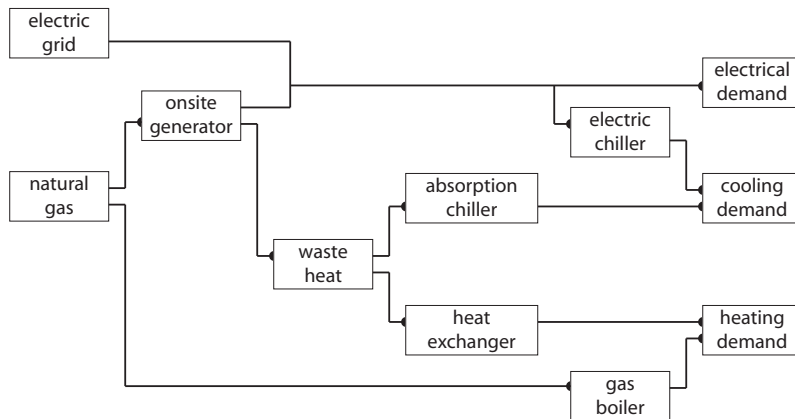
6.1 Case study

This case study demonstrates the decomposition of a problem into a main problem and a subproblem, where the subproblem is tractable for offline optimization over a conditions grid, and whose solution grid is referenced by an online MPC in solving the main problem.

6.1.1 Case description

Figure 52 shows the schematic of a generic campus heating, cooling and electricity plant which provides hot and cold water to meet the heating and cooling demand of a set of buildings, as well as providing onsite generation or purchasing grid electricity to meet the electrical demand of the buildings. Real-time decisions of what level at which to run the onsite generation must be made, as well as the output levels of the the absorption chiller, electric chiller, heat exchanger (generation waste heat to the hot water loop) and gas boiler, given the varying demands for electricity, cooling and heating, the varying price of grid electricity and also considering a charge on the monthly peak draw from the electric grid. (A monthly demand charge like this is found in various electric utility regions in North America.)

Figure 52: Fuels to loads diagram of the cogeneration plant considered in this study



6.1.2 Problem decomposition

The main challenge in this problem is in dealing with the monthly peak demand. The controller must look ahead over the remainder of the month to determine if it is worth paying more for generator use than the current grid price in order to save on the monthly peak demand charge. This could thus require a lot of optimization variables over the prediction horizon, too many for the problem to be computationally feasible. However, we know that the optimal solution over the remainder of the month would see the peaks in grid consumption trimmed off at a flat threshold (Coffey and Kutrowski, 2006). This structure can be used to reformulate the problem into a high-level problem and a repeating sub-problem. The high-level problem determines the grid threshold for the month. The subproblem determines real-time setpoints for the cogeneration plant based on current conditions and the given threshold. The subproblem is of low enough dimensionality to be solved offline, and the meta problem is an online optimization with just one optimization variable, and references the subproblem solution in its objective function.

6.1.3 Subproblem model description

The subproblem model inputs and outputs are as follows,

- disturbance inputs: $\dot{Q}_{electricity}$, $\dot{Q}_{cooling}$, $\dot{Q}_{heating}$, $c_{unitGrid}$, $\dot{Q}_{gridThreshold}$
- control signals: $u_{genSignal}$, $u_{absChillerSignal}$
- objective function output: $c_{gridConsumption} + c_{gasConsumption} + p_{threshold}$

where $\dot{Q}_{electricity}$ is the campus demand for electricity, $\dot{Q}_{cooling}$ is the campus demand for cooling, $\dot{Q}_{heating}$ is the campus demand for heating, $c_{unitGrid}$ is the price of grid electricity per kWh, $\dot{Q}_{gridThreshold}$ is the threshold for grid consumption determined by the meta problem, $u_{genSignal}$ is the signal to the generator (a value between 0 and 1, where 0 is off and 1 is full output), $u_{absChillerSignal}$ is the signal to the absorption chiller (a value between 0 and 1, corresponding to the fraction of waste heat available that will get used by the absorption chiller), $c_{gridConsumption}$ is the cost of grid electricity consumption (not including the demand charge), $c_{gasConsumption}$ is the cost of gas consumption, and $p_{threshold}$ is a penalty function that is used if the grid threshold is exceeded.

A screenshot of the TRNSYS model is shown in Figure 53, and some of the key model parameters are shown in Table 8.

Figure 53: TRNSYS model of cogeneration system

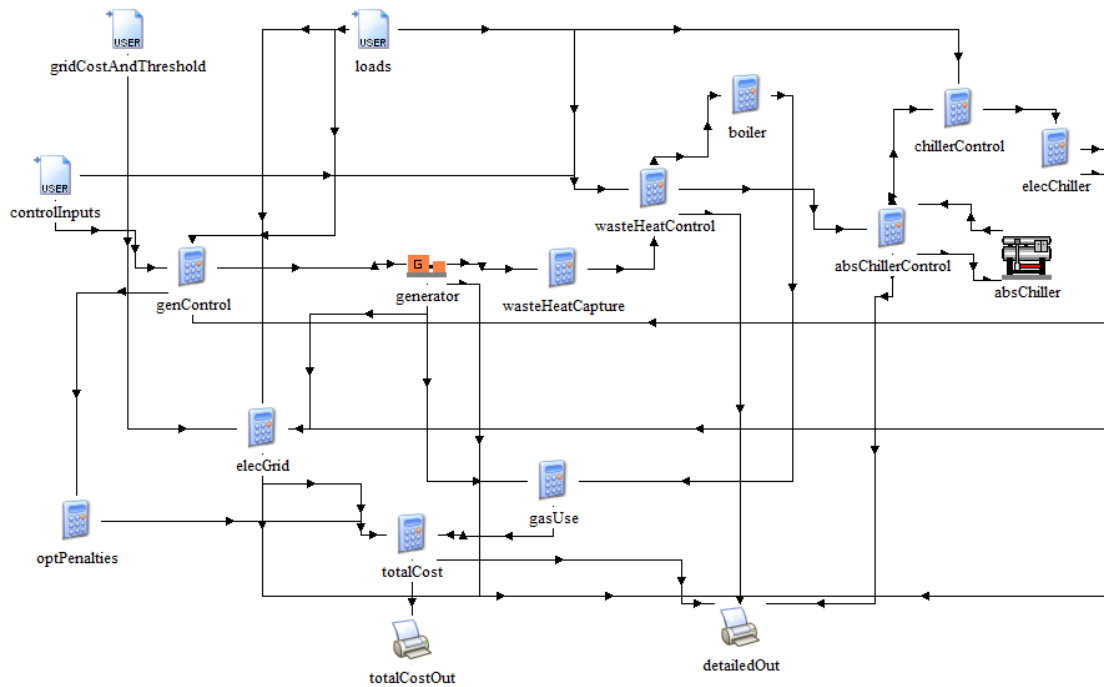


Table 8: Key model parameters

generator electrical capacity	350 kW
generator minimum electrical loading	125 kW
generator rated electrical efficiency (default in standard TRNSYS generator model) (modified by standard TRNSYS curve for other conditions)	0.42
waste heat capture effectiveness	50%
absorption chiller rated COP (default in standard TRNSYS abs. chiller model) (modified by standard TRNSYS curve for other conditions)	0.53
electric chiller COP (modeled as constant)	3.0
natural gas price	\$1.50 / therm
controller timestep	1 hr

6.1.4 Optimization configuration

The conditions grid is shown in Table 9. This grid configuration results in 15125 sets of disturbances for which optimal control was to be determined. The GenOpt Hookes-Jeeves Multi-Start algorithm was used (to avoid local minima), with 4 step size reductions and 20 initial points, and non-parallelized so running on just one processor. The sequential optimizations required approximately 210 hours (8.8 days) on a single Windows virtual machine.

Table 9: Disturbances grid

	min	max	step size
electricity demand (kW)	100	500	100
heating demand (kW)	0	200	50
cooling demand (kW)	0	100	25
electricity price (\$/kW)	0.10	0.20	0.01
electrical grid threshold (kW)	250	500	25

6.1.5 Subproblem lookup table results

Figures 54 through 56 show various slices through the resulting lookup table. In Figure 54, the optimal generation level is shown in the vertical axis versus the electric grid cost on the horizontal axis. The different lines correspond to different electric demand levels for the campus, with the darker lines representing higher demand. In this graph, both the heating and cooling demands are fixed at zero, so any waste heat will be unused. The shapes of the lines are instructive: for the most part, when the electricity price is low, the generator is turned off (since it is cheaper to purchase electricity than it is to run the generator), and when the electricity price is high the generator is turned on to a level determined by the campus electric load, the electric demand threshold, and the maximum and minimum allowable generator operation level; the electricity price at which this switch from off to on occurs depends on the price of gas and the efficiency of the generator at the level at which it will be run. Figure 55 shows an example of how these switch prices are modified by the campus heating load - when there is a heat sink available, the generator will be run at a lower electricity price than it otherwise would. Figure 56 shows some of the behaviour of the cooling components (when the heating load is zero). Note, in particular, the bottom line of graphs, for a high grid threshold: when the threshold is high and the electricity price is low, the generator will be turned off and the electric chiller will

be turned on; under most other conditions, if there is a cooling load, the generator will run and the absorption chiller will be used to meet the cooling load.

Note that the generator, heat exchanger and absorption chiller are usually operating at one of their constraint boundaries (either off, or at capacity, or at the electric load level, or to just maintain the grid purchase threshold, etc). The optimal control values are usually at these constraint boundaries because the efficiency curves in the system model have a less significant impact on the optimal values than do the constraints. This is discussed further in the Discussion section below.

Figure 54: Optimal generator level vs elec price and elec demand

heating demand = 0 kW, cooling demand = 0 kW

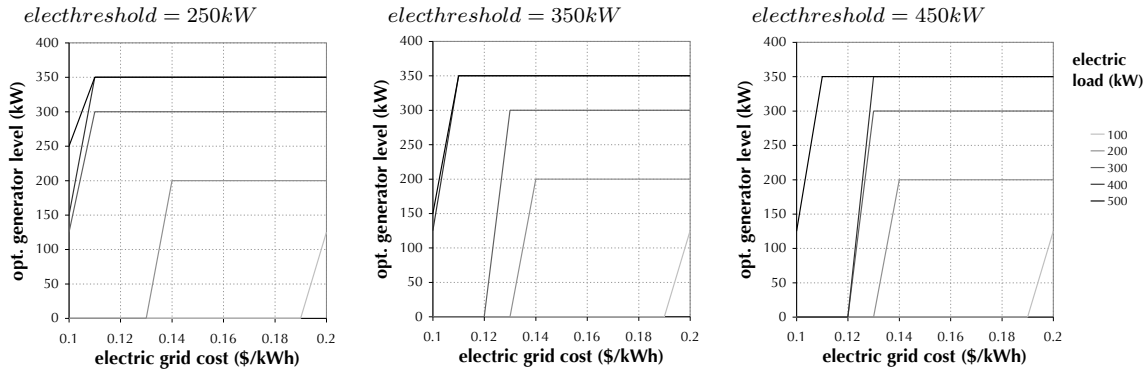


Figure 55: Optimal generator heating level vs elec price and heating demand

elec demand = 300 kW, cooling demand = 0 kW

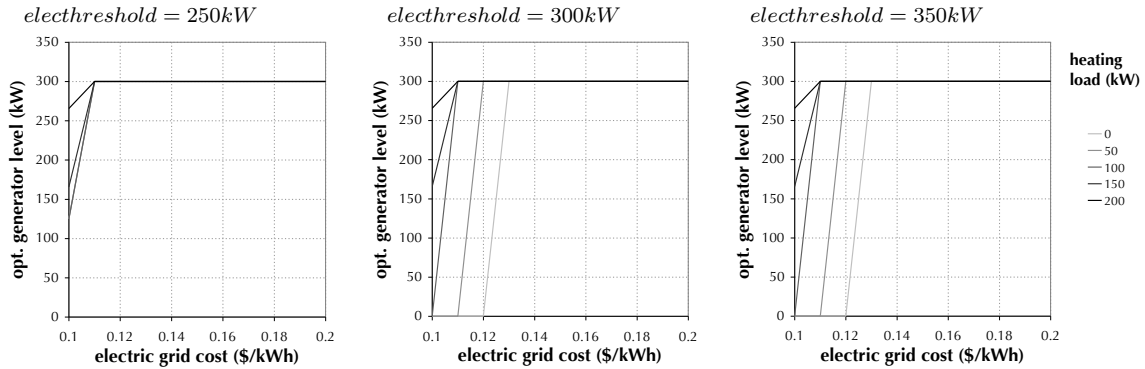
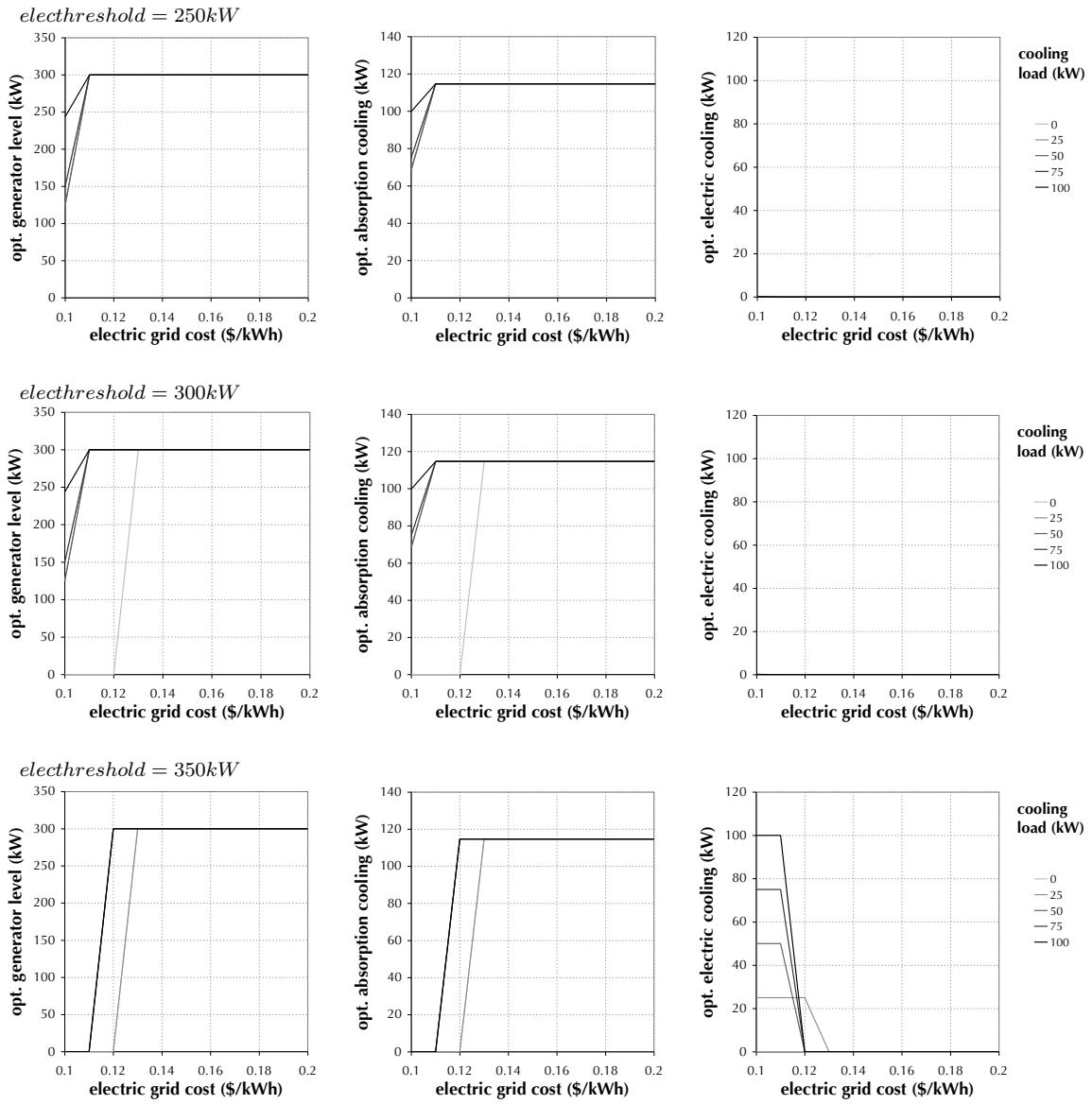


Figure 56: Optimal generator and chiller levels vs elec price and cooling demand

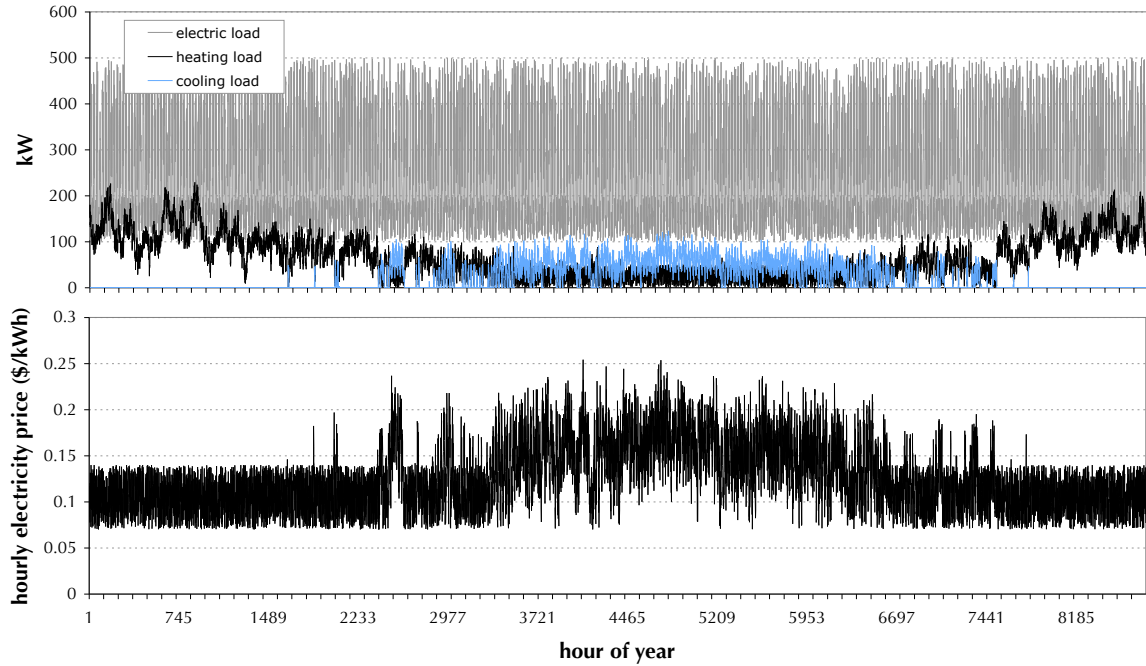
elec demand = 300 kW, heating demand = 0 kW



6.1.6 Annual simulations

To test the controller, annual simulations were run using randomized electrical, heating and cooling loads for a hypothetical campus, as shown in Figure 57, using the Chicago TMY weather file to provide variation. The electrical loads are uniformly random between (100-200kW) overnight and between (300-500kW) during the day. The heating and cooling loads are probability functions of the outdoor temperature. The electricity price is uniformly random between \$0.07-0.14/kWh when the outdoor temperature is low, and varies as a probabilistic function of the cooling load in the summer.

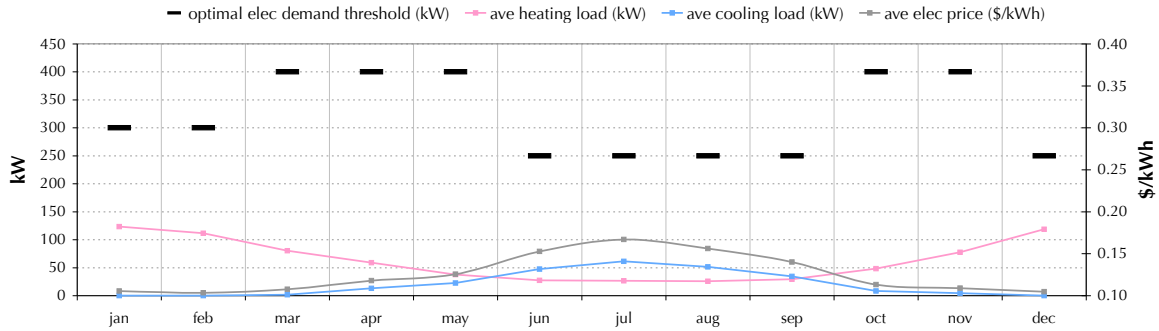
Figure 57: Loads and prices for annual simulation



Optimal monthly thresholds were found by running a GenOpt optimization for each month, minimizing the total monthly cost (gas consumption cost + electricity consumption cost + peak demand charge cost) as a function of the threshold, using the subproblem lookup tables to determine the control configuration for each hour given the threshold. The optimal monthly thresholds shown in Figure 58 are for a monthly peak demand charge of \$1.20/kW. Winter and summer have lower optimal

thresholds: in winter this is because there is usually enough demand for waste heat, which effectively decreases cost of using the generator to shave peaks; and in summer the electricity price is higher, so the generator with absorption cooling can be used cheaply to offset peaks. The spring and autumn have higher thresholds because they lack these benefits.

Figure 58: Optimal monthly thresholds



These monthly thresholds were then used with the lookup table controller for a full-year simulation. For comparison, a case without any cogeneration system was run, as well as a cogeneration system with a base case control strategy. The base case control simply runs the generator at capacity (or at the electric load level if this is less than the capacity, or turned off if the electric load is less than the minimum operating level of the generator), with the waste heat being used first for the heating load and then if any left it is used for the absorption chiller. Figure 59 shows the hourly grid purchase levels for the three cases over the year. The lookup control is using the generator extensively during the summer months, slightly less during the winter, and not very much in the spring and fall. Note the flat line demand trimming that varies monthly (it is particularly apparent in the graph during the winter months).

Figure 59: Annual hourly electric grid purchases

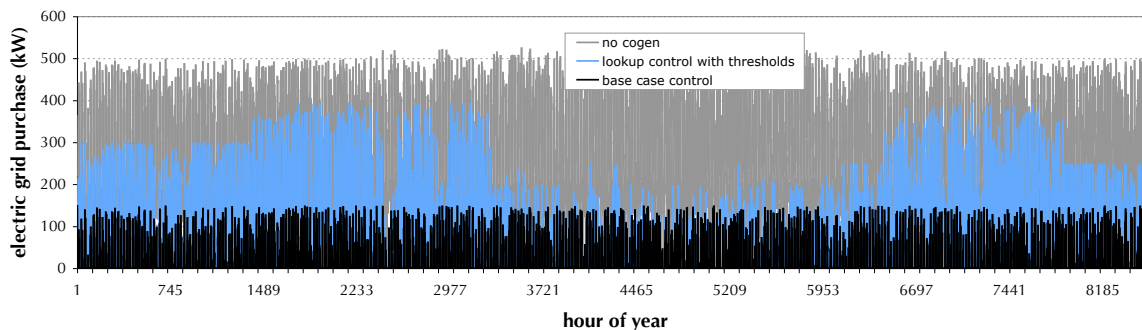


Table 10 compares the energy costs for the three cases. The cogeneration installation would pay back about 5% faster with the lookup table control than it would with the base case control strategy.

Table 10: Annual energy costs comparisons

		no cogen	base case	lookup
total electricity and gas consumption cost		\$309,527	\$279,201	\$275,422
monthly demand costs	jan	\$600	\$181	\$360
	feb	\$599	\$179	\$360
	mar	\$600	\$180	\$472
	apr	\$624	\$180	\$473
	may	\$632	\$180	\$474
	jun	\$628	\$179	\$300
	jul	\$635	\$179	\$300
	aug	\$629	\$180	\$300
	sep	\$625	\$179	\$300
	oct	\$621	\$179	\$474
	nov	\$602	\$176	\$471
	dec	\$600	\$180	\$300
total annual demand costs		\$6,794	\$2,151	\$4,225
annual total cost		\$316,322	\$281,352	\$279,647
annual savings			\$34,970	\$36,675
annual savings %			11.06%	11.59%
% faster payback on cogen investment				4.88%

6.2 Discussion

This case study has demonstrated the use of problem decomposition and the use of TRNSYS within this approach. The problem and the results are relatively straightforward. The biggest outstanding question is how best to deal with the fact that the optimal control responses tend to be on constraint boundaries and show many discontinuities.

6.2.1 Irregular grid and/or heuristics to deal with control switch points

Sampling over a regular grid is not particularly well suited to cases like this one with discontinuities in the optimal control values. Sampling techniques that provide higher density of points near changes of values are likely to provide better results.

Another potential approach to this problem is to combine the conditions grid optimization with analytical (or heuristic) determination of some particular switch points. For example, in this case when there is zero demand for cooling or heating and the electrical load is greater than the generator capacity and the demand threshold is high, one may calculate the switch point price at which the generator should switch from off to on as follows, where $c_{generation}$ is the cost of generation per kWh, c_{natGas} is the price of natural gas, and $\mu_{generator}$ is the generator rated electrical efficiency.

$$\begin{aligned}c_{generation} &= c_{natGas} / \mu_{generator} & (20) \\ &= (\$1.50/\text{therm}) / 0.41 \\ &= \$0.1248/\text{kWh}\end{aligned}$$

Similar calculations that incorporate the money saved by offsetting heating or cooling can also be used to calculate other switch point prices for other loading conditions. The model-based optimization over the conditions grid could thus be used to map out the optimal control across the conditions space, and analytical solutions could be used to refine the control at certain points. (Similarly, in abstract case study #1, an analytical solution could be used to improve the lookup table controller by keeping the natural ventilation level at 100% right until the ambient temperature equals the zone temperature, then dropping to 0% for any warmer temperature, as discussed in Chapter 4). Allowing for such heuristic interventions into the control configuration is a practicality worth considering in any further development of this approach.

6.2.2 Computational infeasibility of a full online solution

As noted in the problem description, running a full online MPC for any given hour, without the problem decomposition, would be computationally infeasible. The optimization problem faced at the first hour of the month would be for 1440 optimization variables ($30 \text{ days} \cdot 24 \text{ hrs/day} \cdot 2 \text{ optimization variables / hr}$). As such, a comparison with full online MPC could not be carried out for this case.

7 Detailed Case Study #1: Extension of the UC Merced chilled water system MPC study

7.1 Case description

7.1.1 Background

The campus chilled water system at UC Merced includes a two million gallon storage tank that is charged overnight when electricity is less expensive and the wet bulb temperature is lower (and thus when the charging COP is higher). A research project by a group of researchers from Lawrence Berkeley National Laboratory, UC Berkeley, UC Merced and United Technologies Research Center was carried out in 2009 to develop and test an online MPC implementation for this system. Two weeks of experiments were performed, the details and results of which are provided in Ma et al. (2010), Coffey et al. (2010b), and the final project report (Haves et al., 2011). The first experiment uncovered a number of bugs that were fixed before the second experiment, which showed an increased cooling COP of approximately 3% +/- 2% over their baseline operation.

This MPC implementation was written in Matlab (based on models created and tested in Modelica) and used the proprietary optimizer Tomlab. A 3-day prediction horizon was used. The online optimization solved each evening required about 20 minutes of computing time on a laptop. The experiments were carried out by collecting weather predictions and system state data every evening and running the optimization on the laptop before providing instructions to the operators in person or by phone or email.

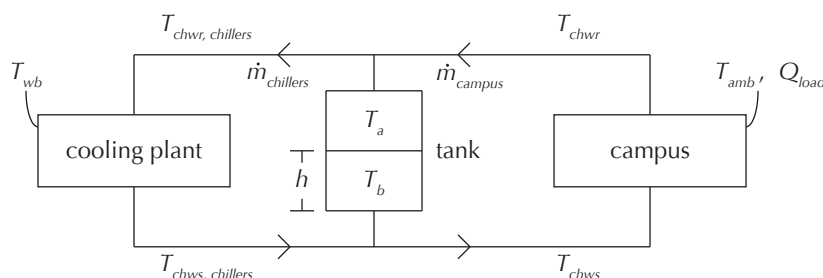
The work presented herein is a follow-up project with the goal of producing a fast-running approximation to the rigorous MPC implementation developed previously. This could be used in longer-term physical implementation on campus and in various annual simulation studies, and could be modified for use on other campuses with similar systems.

7.1.2 Description of campus chilled water system

The main components of the UC Merced chilled water system are shown in Figure 60. Water cooling is performed by the chillers and cooling towers in the cooling plant. The chilled water is stored in a stratified thermal energy storage (TES) tank and distributed to the campus buildings via a secondary loop (on the right in Figure

60). Tertiary loops (not shown in Figure 60) within each building distribute the chilled water to fan coils and air handling units. During tank charging, cool water from the chillers enters the bottom of the tank and warm water from the top of the tank is returned to the chillers. During tank discharging, cool water from the bottom of the tank is supplied to the campus, and warm from the campus returns to the top of the tank. The central plant energy management and control system (EMCS) provides the supervisory control that coordinates the electric chillers, cooling towers and pumps used in the overnight charging of the tank.

Figure 60: Chilled water storage system



7.1.3 Problem definition

The goal of controller development for this case is to minimize the cooling plant energy consumption, primarily by choosing the optimal times to charge the tank, and by intelligently controlling other cooling plant setpoints such as the condenser water temperature setpoint. Equation 21 provides a high-level view of the control response that must be calculated each night before 10pm.

$$\begin{aligned} \text{control signals} &= \text{controller}(\text{conditions}) \\ [t_{start}, t_{stop}, T_{cws}] &= \text{controller}(\mathbf{T}_{wb}, \mathbf{T}_{amb}, \dot{\mathbf{Q}}_{campus}, T_{a0}, h_0) \end{aligned} \quad (21)$$

The control variables are the charging start time (t_{start}), the charging stop time (t_{stop}), and the condenser water temperature set-point (T_{cws}) for the night. The conditions variables are as follows: the initial temperature in the warm part of the chilled water storage tank (T_{a0}), the initial height of the thermocline in the storage

tank (h_0), the predicted overnight ambient wet bulb temperature profile over the prediction horizon (\mathbf{T}_{wb}), the predicted full day ambient dry bulb temperature profile over the prediction horizon (\mathbf{T}_{amb}), and the predicted full day campus load profile over the prediction horizon ($\dot{\mathbf{Q}}_{campus}$). A pre-calculation predicts the campus load profile ($\dot{\mathbf{Q}}_{campus}$) as a function of the predicted ambient temperature, solar position, cloud cover, and a schedule of internal loads based on the day of week.

The base case controller (based on how the system was operated before the online MPC experiments) ignores all of these conditions variables except the initial height of the thermocline (h_0) and the ambient wet bulb temperature (\mathbf{T}_{wb}) for just the upcoming night. The heuristic rules used in the base case are as follows: if $h_0 > 60\%$ then do not charge at all that night, otherwise $t_{start} = 9\text{pm}$ and charge until full; and T_{cws} varies between 14.5°C and 20°C as a function of the average overnight wet bulb temperature.

The controller described herein responds to all of the conditions variables, and performs an optimization with a system model. The system model is based on the system component models developed and used in the previous research project, as described below, but the overall model structure is modified by the problem decomposition described in section 7.2.

7.1.4 System component models

A detailed description of the UC Merced chilled water system, and the component models and optimization configuration used in the initial study, are available in the project report (Haves et al., 2011). Detailed descriptions of the campus load and return temperature models are also included herein as the Appendix. For the purposes of this case study description, only a high level view of them is required.

The three main blocks in Figure 60 are the cooling plant, the storage tank and the campus. In this case study, the cooling plant model is a steady-state model and can be considered as a black box with the inputs and output for any given timestep as shown in Equation 22. Note that in this case study, the number of chillers ($n_{chillers}$) is fixed at 2, the flow rate through the chillers ($\dot{m}_{chillers}$) is fixed at its maximum allowable flow rate when charging, and the chilled water supply temperature ($T_{chwsChillers}$) is held constant at 4°C .

$$\dot{Q}_{plantConsumption} = \text{plantModel} (T_{wb}, T_{cws}, T_{chwrChillers}, T_{chwsChillers}, \dot{m}_{chillers}, n_{chillers}) \quad (22)$$

The storage tank is modeled as two well-mixed volumes with a sharp division between them (the ‘thermocline’) that is at a variable height (thus their individual volumes can change but their sum is constant). Note that in both the original MPC implementation and in this approximated one, the thermal losses through the tank wall are ignored. Because the tank is well insulated and has a low surface to volume ratio, tank losses are not significant, and this assumption simplifies the problem. The campus block predicts the cooling load, return temperature and flow rate through the secondary loop.

7.2 Problem decomposition

The problem is parsed into a subproblem and a main problem, where the subproblem pertains only to components modeled as steady-state and whose dimensionality can be made small enough to allow for the production of a lookup table through offline optimizations. This results in a much simpler online optimization problem (using the subproblem lookup table in its objective function), that can be quickly solved with GenOpt. To keep the dimensionality of the lookup table small enough to be tractable, conditions parametrization is used, as discussed in section 7.3.

The system model contains just three state variables: the tank thermocline height and the temperatures of the hot and cold nodes in the tank. The rest of the model is treated as steady-state. (The campus load model is dynamic, but it is external to the system model, acting as a disturbance model.) The control problem can be parsed in such a way to decouple the parts of the problem that rely only on the cooling plant model from the parts that rely on the tank model as well. Consider a divisions of days and nights over the prediction horizon, with the daytime period for a given day i defined as 8am to 10pm, and the night period defined as 10pm to 8am the following day. Note that tank charging can only occur during the night period. The controller must determine the charge amount Q_{charge_i} , the charging start time t_{start_i} , the condenser water temperature setpoint T_{cws_i} , and the number of chillers $n_{chillers}$ for each night in the prediction horizon. The problem decomposition is as follows:

- Main problem: A higher level optimization problem determines the charge amounts Q_{charge_i} for each night over the horizon, using the solution to the subproblem to relate the optimal energy consumption of the plant for an overnight period i to any possible value of Q_{charge_i} .
- Subproblem: For any overnight period i , given the charge amount Q_{charge_i} , a lower level optimization problem (solved offline over a conditions grid) deter-

mines the optimal value of the start time t_{start_i} and condenser water temperature T_{cws_i} , and the associated optimal energy consumption of the plant for that overnight period.

7.2.1 Subproblem definition

Note that the solution of the subproblem for each night is nearly independent of the solution for the other nights. It is primarily dependent on the given values of charge amount Q_{charge_i} , the predicted overnight T_{wb} values, and the overnight values of the hot node in the tank T_a . Since the solution of the higher level optimization problem depends on testing different values for the charge amount Q_{charge_i} for each night, the subproblem may be called many times in the course of solving the whole problem. The solution of this subproblem requires only the cooling plant portion of the system model, which is fast running and free of state variables. This makes the subproblem amenable to offline solution over a conditions grid, resulting in a lookup table that may be used in place of online optimization.

For one overnight charging period i , with a constant overnight T_{cws} , the overnight plant energy consumption is as shown in Equation 23,

$$\begin{aligned} & [Q_{plantConsumption_i}, t_{stop_i}, T_{a8am_{i+1}}, h_{8am_{i+1}}] \\ & = \text{plantSimulation} (\mathbf{T}_{wb_i}, \mathbf{T}_{chwrCampus_i}, \dot{\mathbf{m}}_{chwrCampus_i}, T_{a10pm_i}, h_{10pm_i}, Q_{charge_i}, \\ & \quad t_{start_i}, T_{cws_i}) \end{aligned} \quad (23)$$

where $\mathbf{T}_{chwrCampus_i}$ is the overnight campus return temperatures, $\dot{\mathbf{m}}_{chwrCampus_i}$ is the overnight campus flow rates, T_{a10pm_i} and h_{10pm} are the hot node tank temperature and thermocline height at the start of the night, and T_{a8am_i} and h_{8am} are the tank warm node temperature and thermocline height at the end of the night. The optimization problem in Equation 24 defines the subproblem for one charging period.

$$\begin{aligned} & \min_{t_{start_i}, T_{cws_i}} Q_{plantConsumption_i} \\ & \text{with } \mathbf{T}_{wb_i}, \mathbf{T}_{chwrCampus_i}, \dot{\mathbf{m}}_{chwrCampus_i}, T_{a10pm_i}, Q_{charge_i} \text{ given} \end{aligned} \quad (24)$$

This subproblem can be solved offline over a grid of values for \mathbf{T}_{wb_i} , $\mathbf{T}_{chwrCampus_i}$, $\dot{\mathbf{m}}_{chwrCampus_i}$, T_{a10pm_i} and Q_{charge_i} . A number of input approximations are used, as described below, to keep the grid dimensionality low.

the normalized version of the dashed black line. This normalized curve is used to approximate the hourly T_{wb} predictions given the predicted T_{wbMin} and $T_{wbRange}$.

Figure 61: Overnight T_{wb} minimum and range over a typical year

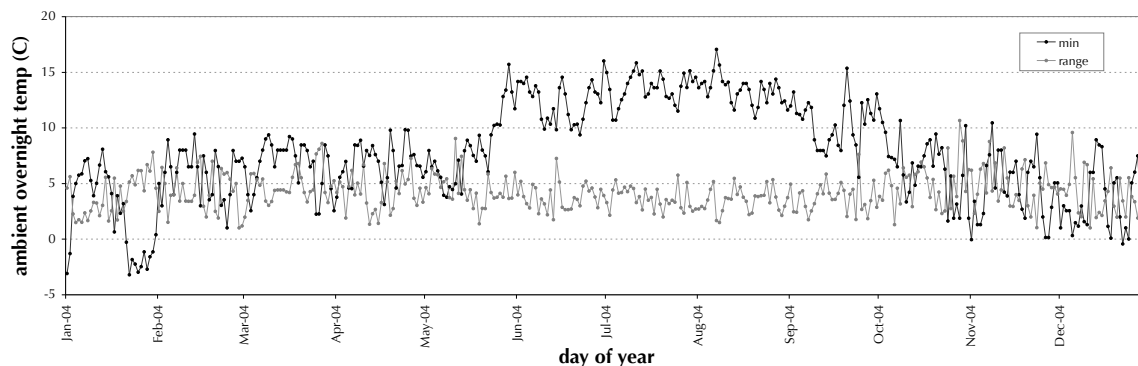
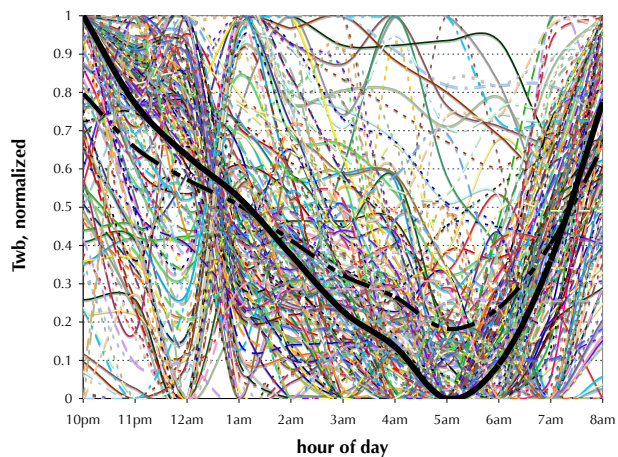


Figure 62: Normalized overnight T_{wb} values



The other necessary approximation relates to the overnight temperatures of the warm node in the storage tank \mathbf{T}_a . To capture this as accurately as the online MPC does, hourly predictions for the overnight campus return flow $\dot{\mathbf{m}}_{chwrCampus}$ and temperature $\mathbf{T}_{chwrCampus}$ would be required, along with the initial temperature T_{a10pm} and initial thermocline height h_{10pm} , thus requiring 24 dimensions in the

lookup table. However, the overnight temperature values are dominated by the initial temperature T_{a10pm} , since the overnight campus flow rates are much smaller than the daytime campus flow rates. So the overnight temperatures may be approximated by using an initial value T_{a10pm} and simulating the progression of overnight T_a by using average hourly values for $\mathbf{T}_{chwrCampus}$ and $\mathbf{\dot{m}}_{chwrCampus}$, and by assuming a value for h_{10pm} . These average values are shown in Figure 63. Note that the campus return temperatures are lower overnight than during the day, so the warm node of the tank gets colder as the night progresses. The annual range of T_{a10pm} values (derived from measured data) is shown in Figure 64.

Figure 63: Overnight campus return temperatures and flow rates

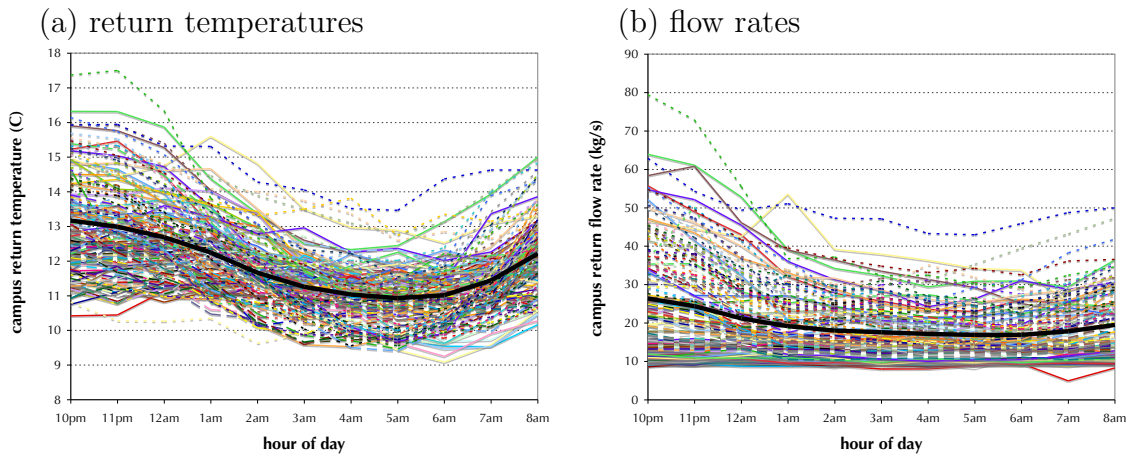
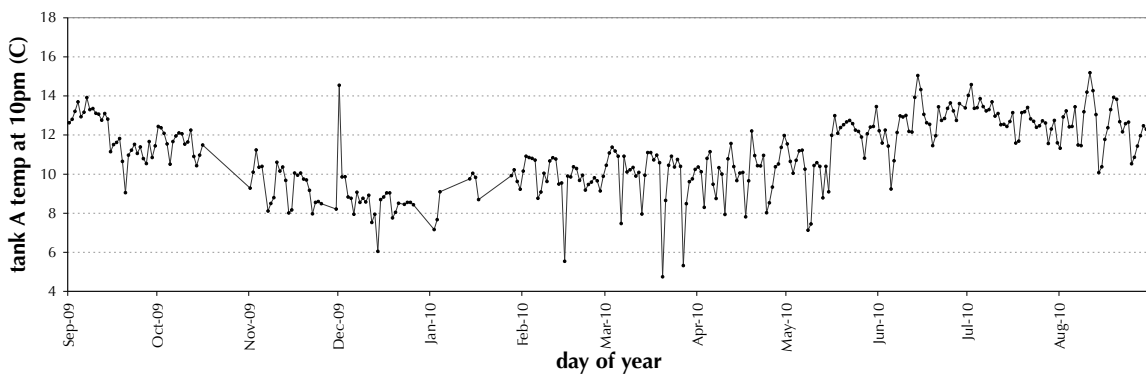


Figure 64: Annual 10pm temperatures of the storage tank hot node



7.4 Subproblem: Optimization configuration

7.4.1 Grid definition

The subproblem was solved over the grid of conditions shown in Table 11. For each point in the conditions space, the subproblem solution vector contains a value for start time, T_{cws} and overnight energy consumption.

Table 11: Grid of conditions for subproblem solution lookup table

Parameter	Min	Max	Resolution
Q_{charge}	10 MWh	90 MWh	10 MWh
T_{wbmin}	0 C	18 C	2 C
$T_{wbrange}$	1 C	11 C	1 C
T_{a10pm}	8 C	16 C	2 C

7.4.2 Computation methods

Solution over the four dimensional grid in Table 11 requires 4950 optimizations. The GPS Hookes-Jeeves optimization algorithm in GenOpt was used. To perform the optimizations, virtual machines on the Amazon EC2 cloud were used. Before the optimizations, the Modelica plant model was gridded into a lookup table, with the inputs and outputs shown in Equation 22 (similar to how it was used in the original MPC implementation), and this lookup table was used as the plant model in the optimizations (using the same lookup table interpolation coded in java). Nine single-processor virtual Windows machines were used, each requiring approximately 30 hours, so a total of approximately 270 processor-hours, which at \$0.12 per processor-hour costs about \$32.

7.5 Subproblem: Lookup table results

Figures 65 and 66 show the optimal start time values (t_{start}^*), versus Q_{charge} , T_{wbMin} , $T_{wbRange}$ and T_{a10pm} . Of interest in interpreting the scatter plot in Figure 65 is the fact that with the conditions parametrization in use, if the optimization were simply always looking for the lowest T_{wb} charging period, there would be very little scatter in the graph, since the charge period would always straddle the time with the lowest T_{wb} (7am in the parametrization), and thus the start time would be a monotonic function of just the overnight charge amount. On the other hand, if the optimization

were searching just for the highest $T_{chwrChillers}$ values, the start time would always be 0, since the overnight campus return temperature is assumed constant and is lower than T_{a10pm} (with the exception of the $T_{a10pm} = 8C$ case), so $T_{chwrChillers}$ is decreasing overnight. So the scatter in Figure 65 between a line at $t_{start}^* = 0$ and a line that is roughly linear downward from an intersection around $t_{start}^* = 7$ at zero charge amount reflects the fact that the optimum value is a trade-off between the desire for a higher $T_{chwrChillers}$ and a lower T_{wb} when charging, which depends on all four of the conditions in the conditions grid.

Figure 66 shows a set of slices through the grid for t_{start}^* versus Q_{charge} and T_{a10pm} . Note that with lower T_{a10pm} values, a given charge amount requires a longer charging time, which is reflected in the different slopes in the start time graphs, and in the gradients in the surface plot at the bottom of the figure.

Figure 65: Scatter plot of optimal start time versus charge length

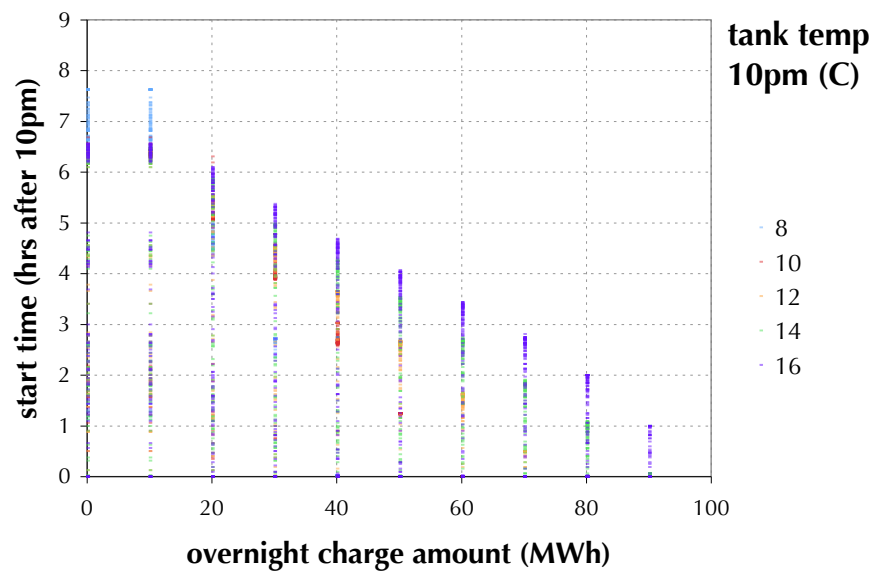


Figure 66: Detailed slices of optimal start time versus charge length and tank temp

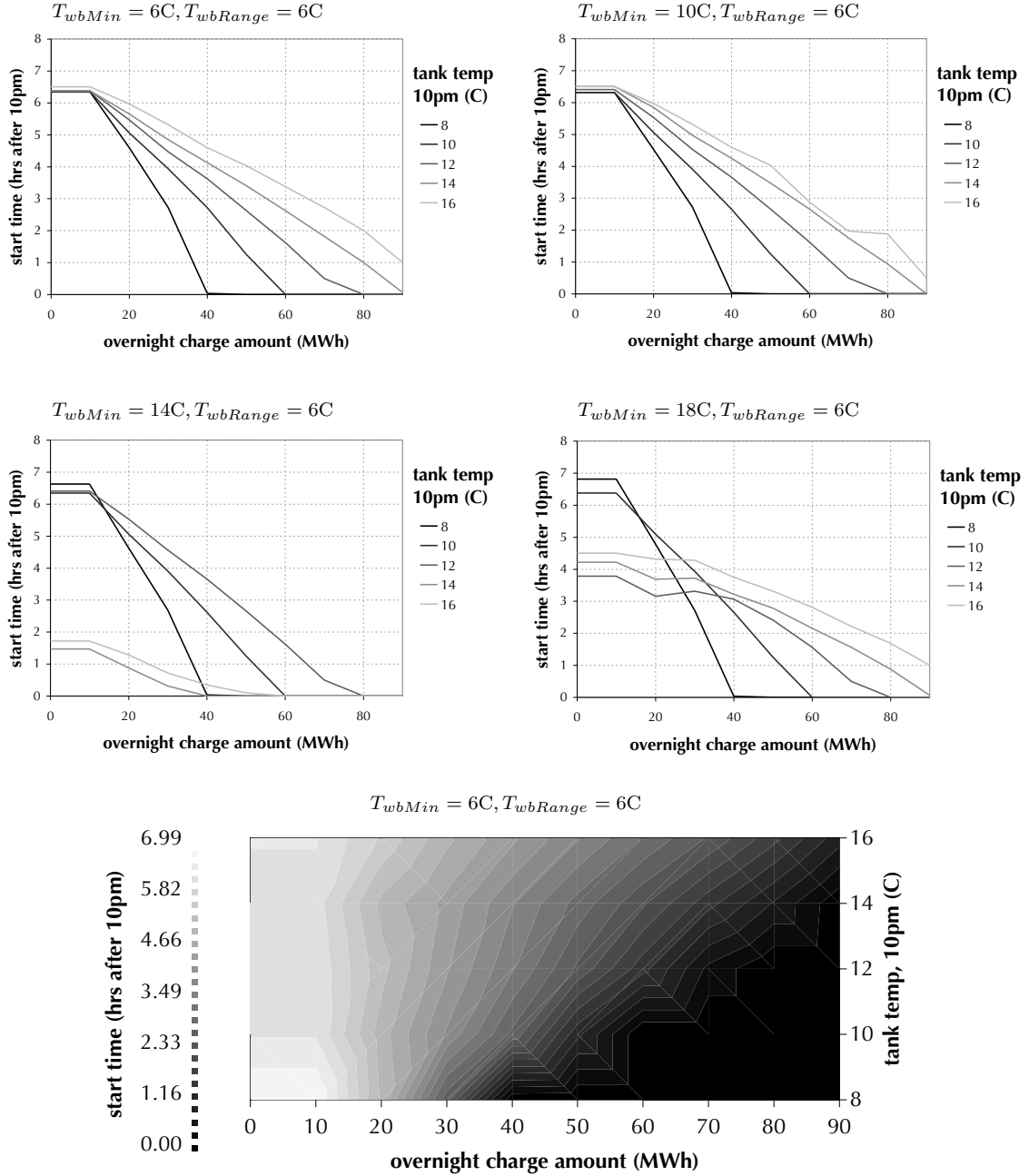


Figure 67 shows a scatter plot of T_{cws}^* versus T_{wbMin} , Q_{charge} , $T_{wbRange}$ and T_{a10pm} , and Figure 68 shows a set of slices through the grid for T_{cws}^* versus T_{wbMin} and T_{a10pm} . Note that, in the scatter plot, the optimal values vary throughout the entire range within its constraints. In the line graphs, the general trend of increasing slope with higher T_{a10pm} , which shows as steeper gradients in the surface plot at the bottom of the figure, show that the optimal value of T_{cws} is more sensitive to T_{wb} when $T_{chwrChillers}$ is higher.

Figure 69 shows slices through the grid for the optimized plant COP versus T_{wbMin} and T_{a10pm} . Note that the optimal COP increases as T_{wbMin} decreases, and increases as T_{a10pm} increases, as expected. But it is much more sensitive to changes in T_{a10pm} than to changes in T_{wbMin} . This means that the higher level MPC problem will want to charge more on nights with higher T_{a10pm} and with lower T_{wbMin} , but it will respond to any changes in T_{a10pm} from night to night much more than to changes in T_{wbMin} .

Figure 67: Scatter plot of optimal condenser temp versus minium wet bulb temp

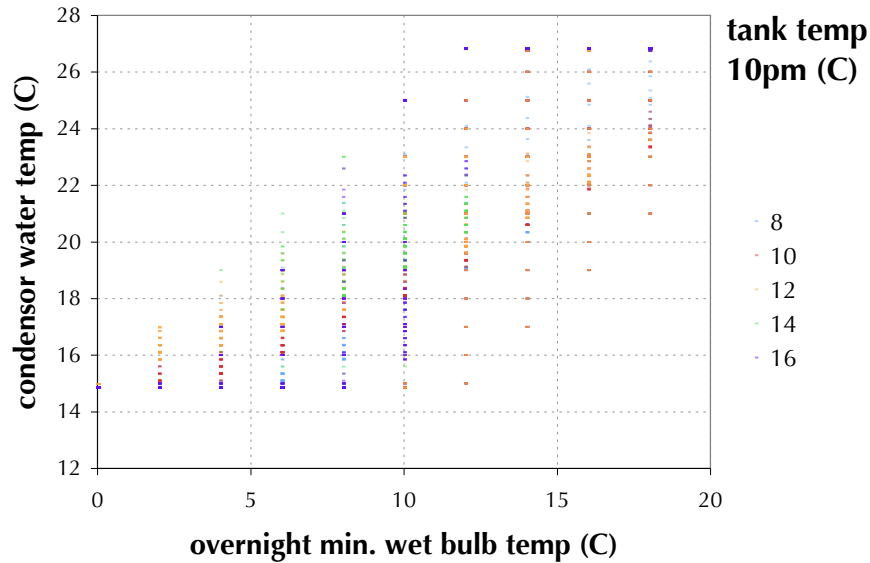


Figure 68: Detailed slices of optimal condenser temp versus T_{wbMin} and T_{a10pm}

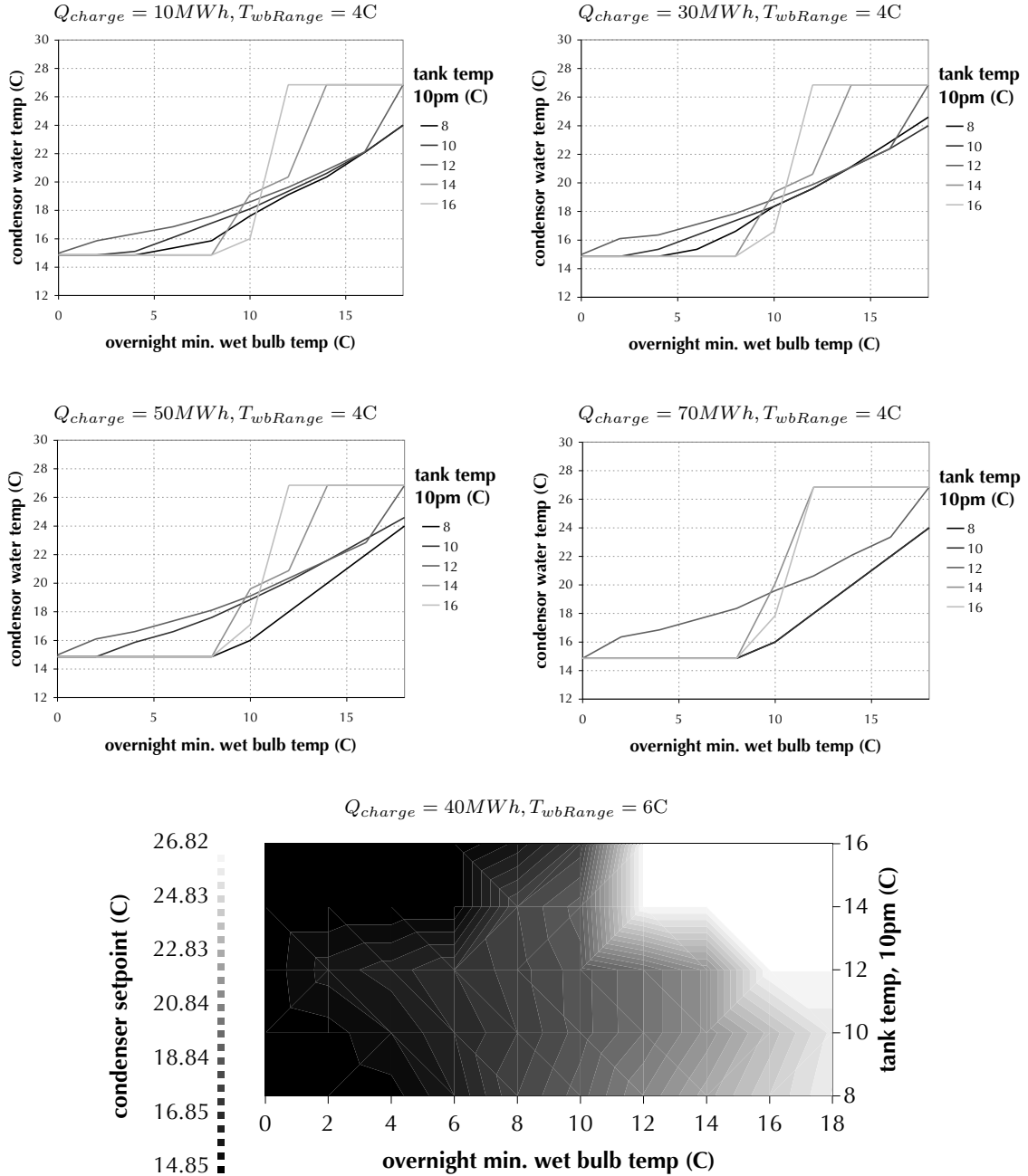
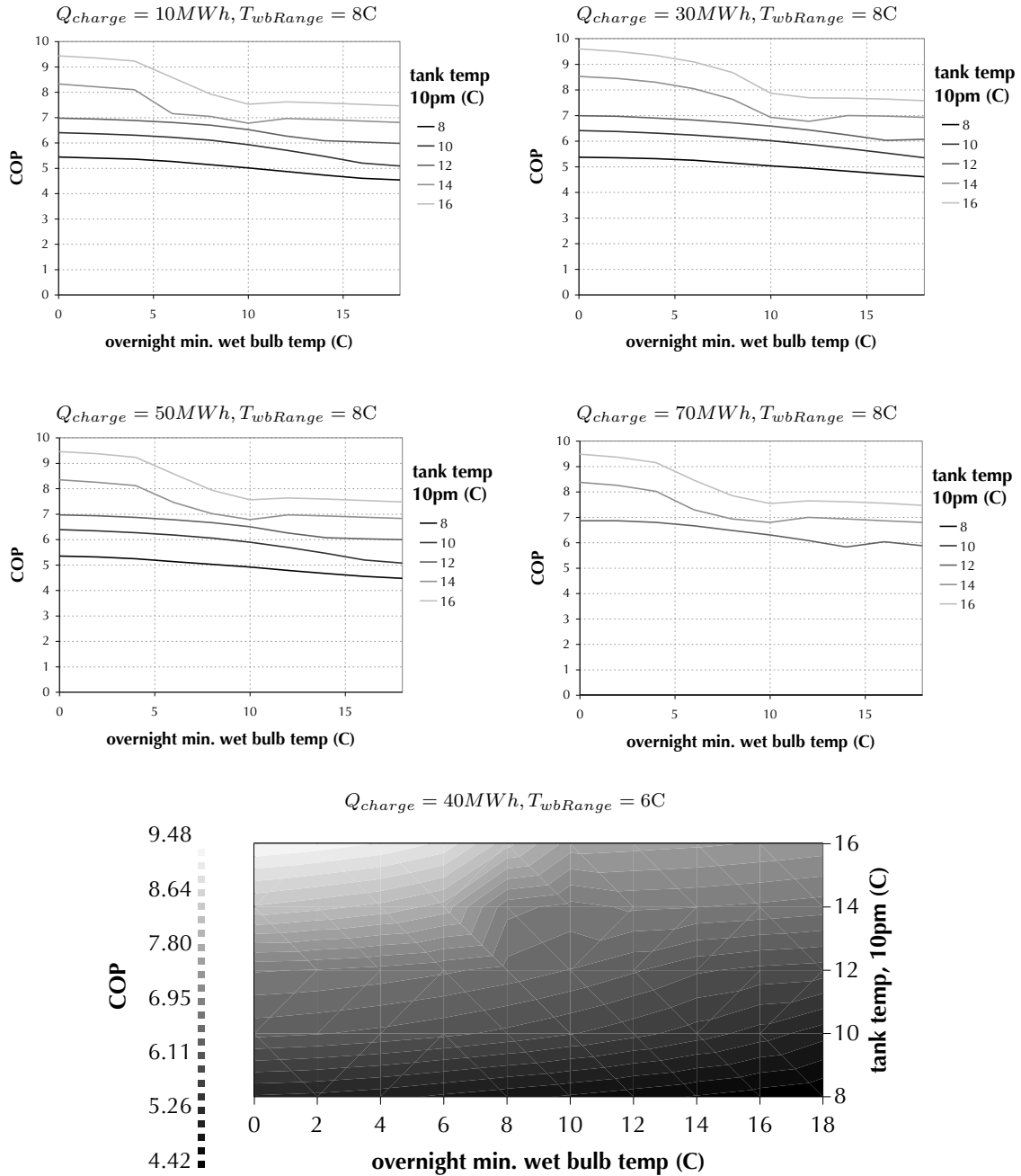


Figure 69: Detailed slices of optimal plant COP versus T_{wbMin} and T_{a10pm}



7.6 Main problem: Online solution using the subproblem lookup table

Given the lookup table of solutions to this sub-problem, the online MPC problem is simply to determine the charge amount for each night over the prediction horizon. The objective function for this online problem is coded in java, using linear interpolation over the lookup table. Penalty functions are used to enforce the thermocline height constraint. The java program takes as input a text file containing values of the following for each night over the horizon: the specified \dot{Q}_{charge} , T_{wbMin} , $T_{wbRange}$, and the number of charge hours required to stay above the minimum thermocline height (which is calculated externally based on the initial thermocline height and the projected campus loads). The prediction horizon can be varied between 1-6 days. The program outputs the total energy consumption plus any penalties for violating constraints, along with the optimal T_{cws} values and charging start times from the lookup table. This objective function program is called iteratively by GenOpt to find the optimal charge amounts.

7.7 Annual simulations

Annual simulations were run with Merced TMY data, with the hourly loads as output from the campus model used in the original MPC implementation. Figure 70 shows the disturbances over the year. The base case control was coded in java and run through the BCVTB configuration in the same way that the approximated MPC implementation was.

For the annual simulation, the approximated MPC used 6.42% less energy than the base case. Figure 71 shows the control decisions, comparing the base case control with that of the approximated MPC. Figure 72 compares the resulting system states over the course of the simulation. Figure 73 shows the charging length (stop time minus start time) over the year. Figure 74 shows scatter plots of the charge length versus the overnight wet bulb temperature, tank temperature and day-ahead return temperature from campus. Figure 75 shows this same data averaged over bins for easier visualization. Note that, relative to the base case control, the approximated MPC generally charges for longer when the overnight wet bulb temperature is lower and less when it is higher. Also note that the approximated MPC generally charges less when next day's campus return temperature is higher (thus when the following night's COP is higher), and more when it is lower. Figure 76 shows the monthly average wet bulb temperature and tank temperature during charging, comparing the base case with the MPC. Figure 77 compares the base case and approximated MPC

in terms of monthly COP and monthly energy use. There are still some unexpected cases during the year when the base case is performing better than the approximated MPC (e.g. December in Figure 77).

Figure 70: Disturbances, Merced

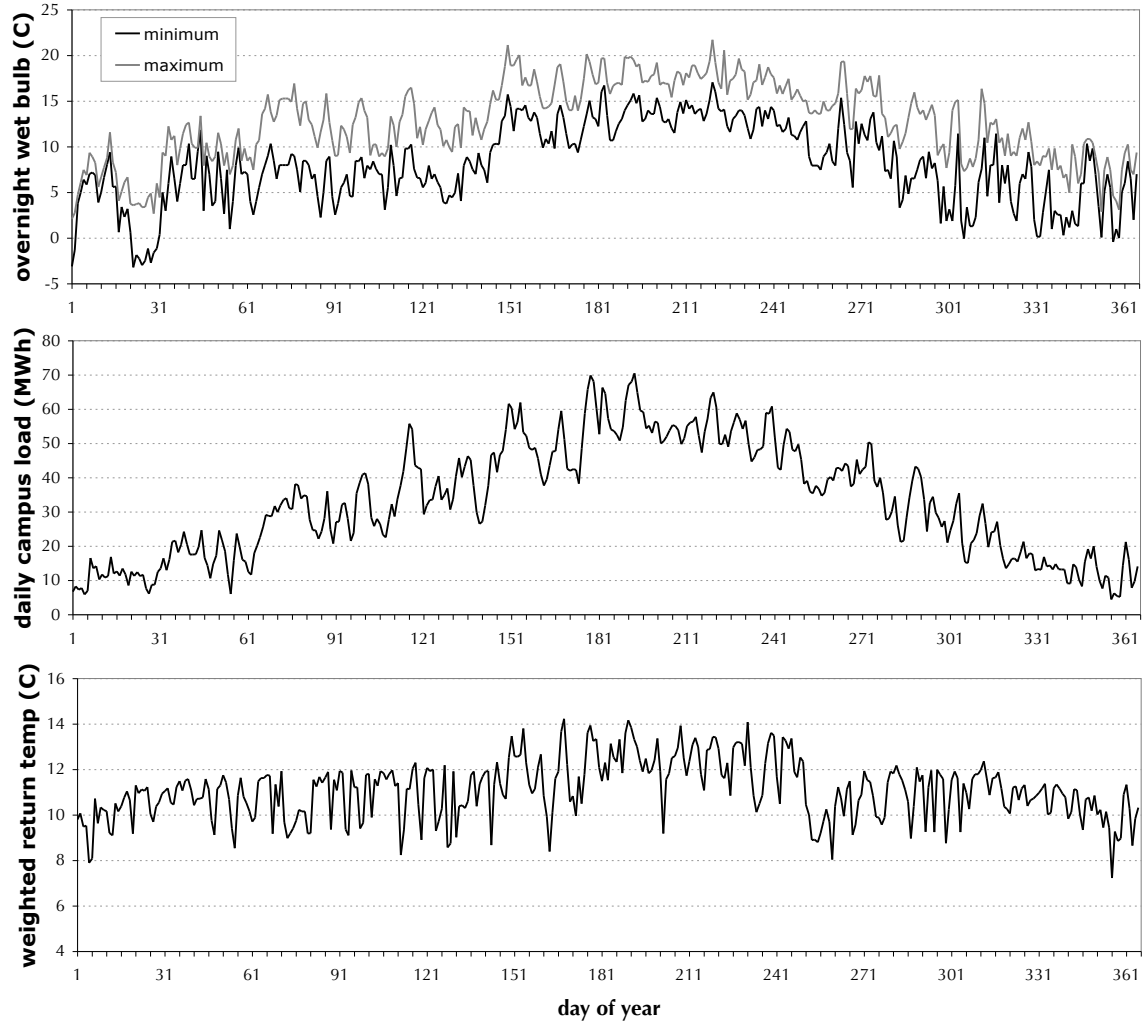


Figure 71: Daily control decisions, Merced

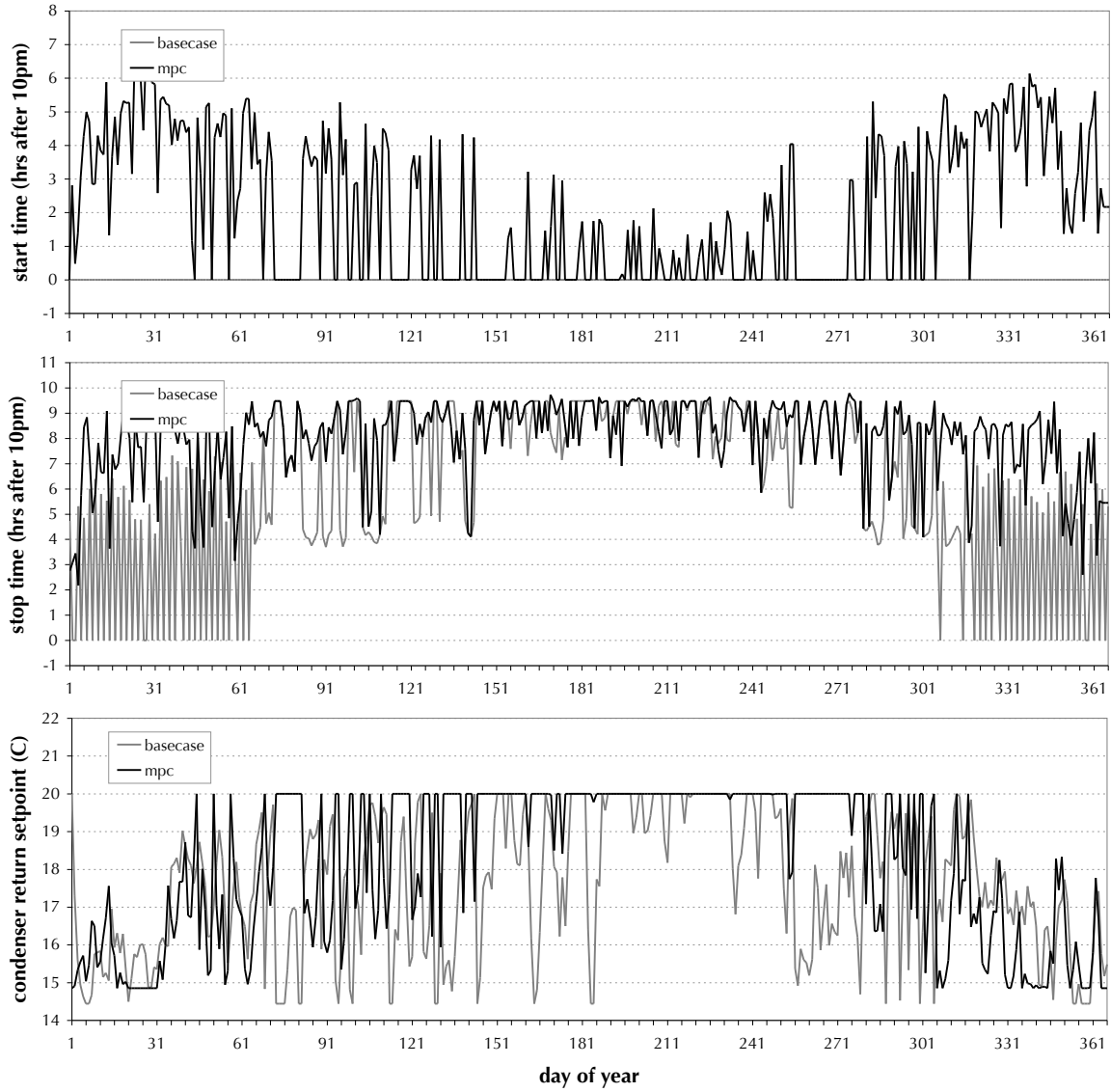


Figure 72: System states, Merced

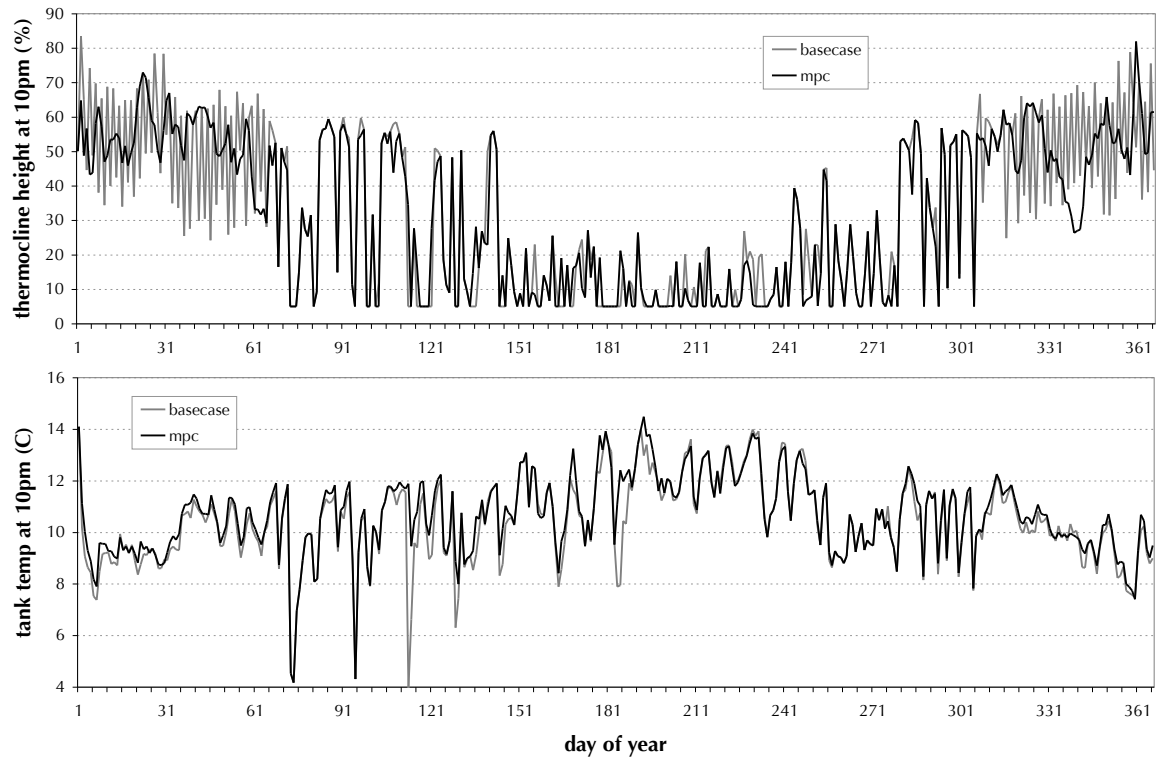


Figure 73: Daily charge length, Merced

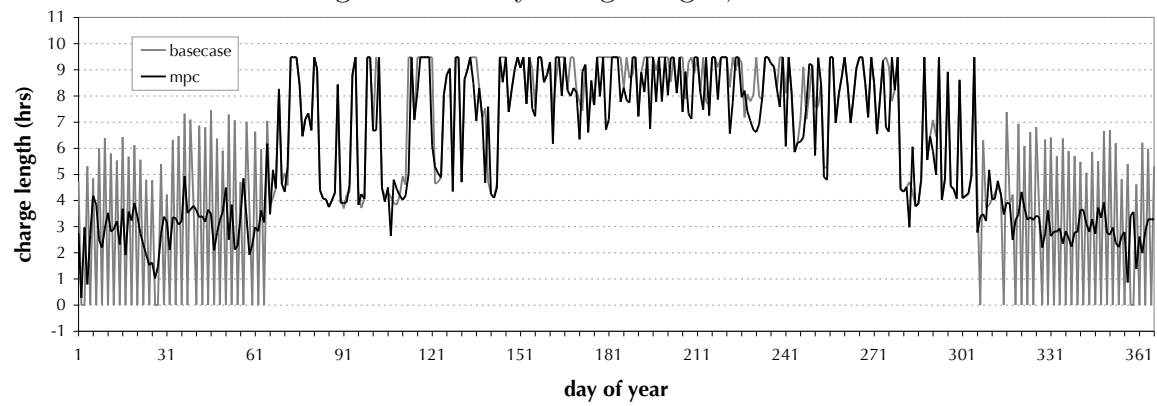


Figure 74: Charge length versus disturbances and states, Merced

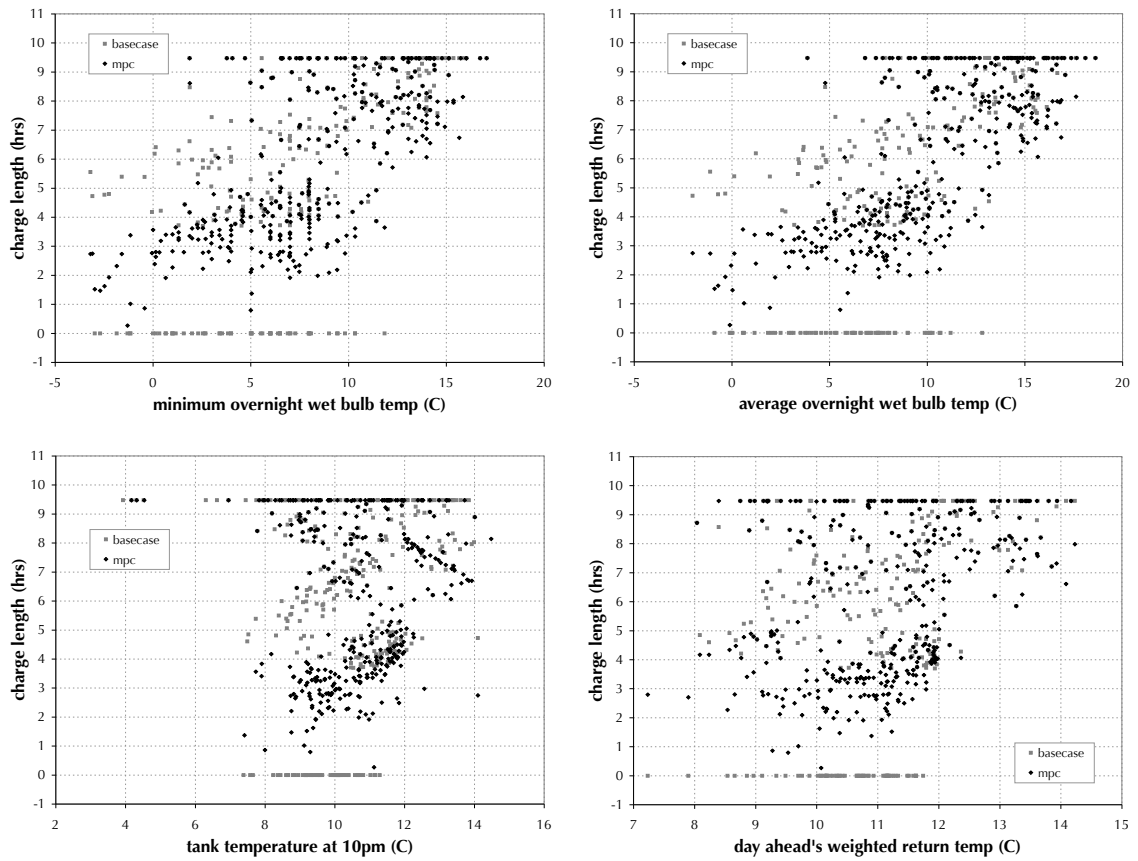


Figure 75: Charge length versus disturbances and states, averaged values, Merced

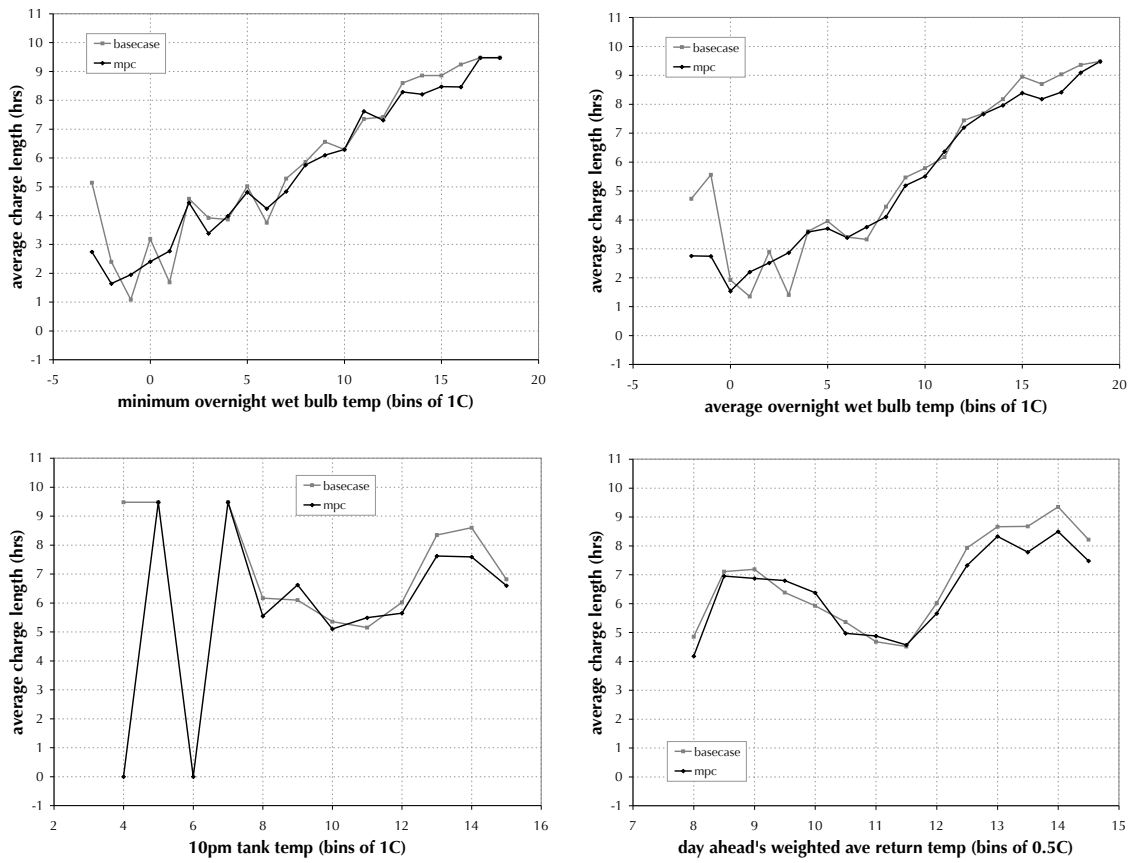


Figure 76: Monthly average wet bulb temperature and tank temperature during charging, Merced

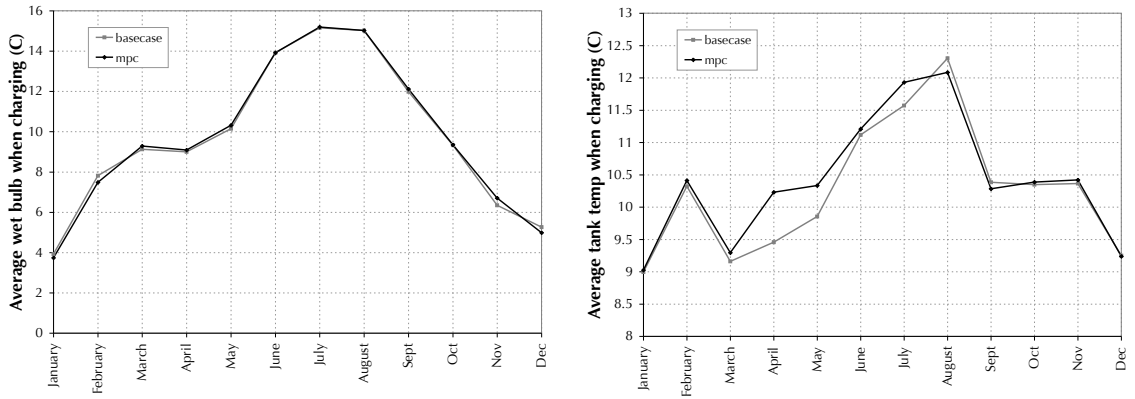
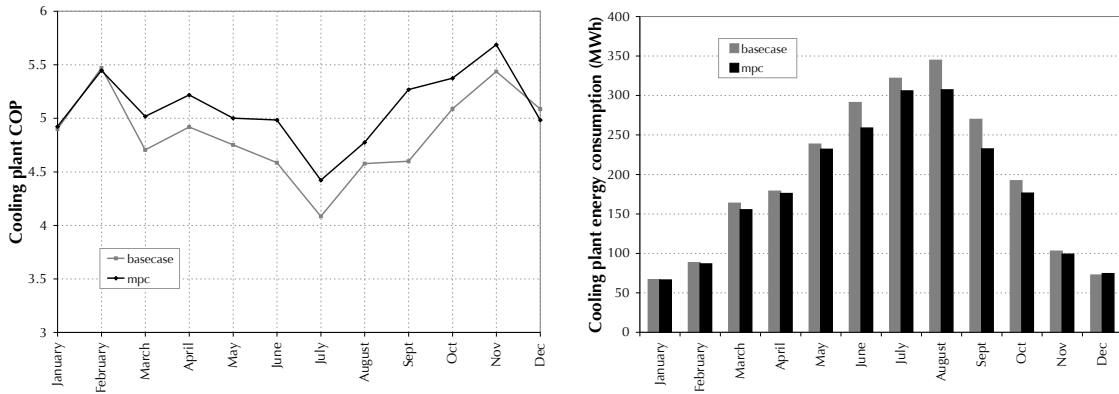


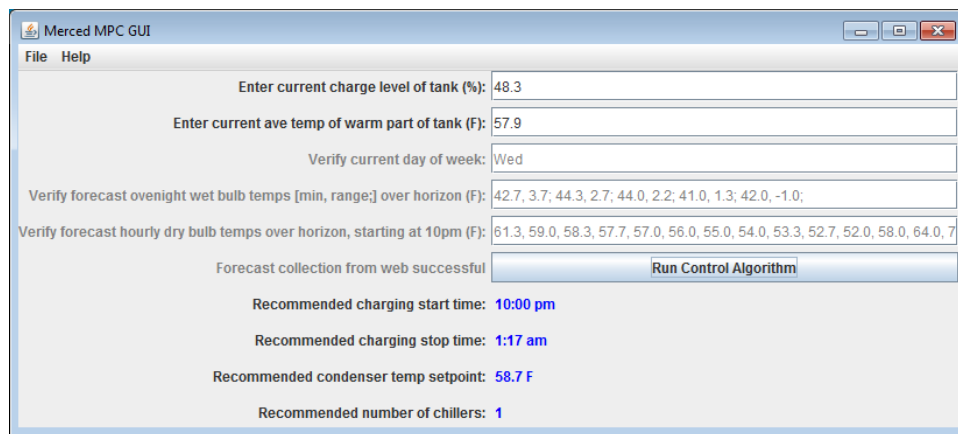
Figure 77: Monthly plant COP and energy consumption, Merced



7.8 Packaged software for use by operators

The interface version automatically downloads the necessary weather forecasts from the National Digital Forecast Database website, so the operators need only input the current charge level of the tank and the average temperature of the tank above the thermocline. (These last two could eventually be pulled from the central energy management system directly, but for now this approach is much simpler and places only a small burden on the operators.) The program is written entirely in java and can be installed and run directly on the user's machine. The computation to determine the optimal setpoints with a 5-day prediction horizon takes about 3-5 minutes on a standard PC.

Figure 78: User interface for approximated MPC implementation



The screenshot shows a Java Swing window titled "Merced MPC GUI" with a menu bar containing "File" and "Help". The interface consists of several input fields and a button:

- Input field: "Enter current charge level of tank (%):" with value "48.3"
- Input field: "Enter current ave temp of warm part of tank (F):" with value "57.9"
- Input field: "Verify current day of week:" with value "Wed"
- Input field: "Verify forecast overnight wet bulb temps [min, range:] over horizon (F):" with value "42.7, 3.7; 44.3, 2.7; 44.0, 2.2; 41.0, 1.3; 42.0, -1.0;"
- Input field: "Verify forecast hourly dry bulb temps over horizon, starting at 10pm (F):" with value "61.3, 59.0, 58.3, 57.7, 57.0, 56.0, 55.0, 54.0, 53.3, 52.7, 52.0, 58.0, 64.0, 7"
- Text: "Forecast collection from web successful"
- Button: "Run Control Algorithm"
- Output: "Recommended charging start time: 10:00 pm"
- Output: "Recommended charging stop time: 1:17 am"
- Output: "Recommended condenser temp setpoint: 58.7 F"
- Output: "Recommended number of chillers: 1"

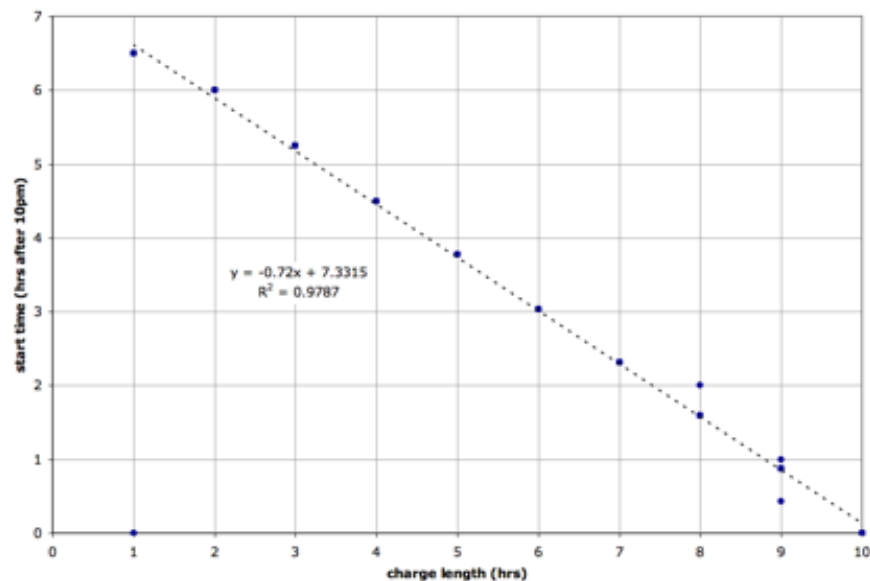
7.9 Discussion

7.9.1 Lessons learned in previous iterations

This case study has undergone a few iterations, and could likely still benefit from at least one or two more (as discussed below). The first iteration was with the assumption of a constant campus return temperature. The difference between the results of that optimization (Figure 79) and the results of the optimization with variable campus return temperature is interesting, and an example of how the ability to visualize the lookup tables can lead to insights into the nature of the problem. In the constant-return-temperature case, the optimal start time is just a function

of the charge length, as it is just trying to minimize the T_{wb} while charging, and thus always charges in a window around the lowest T_{wb} hour. But in the variable-return-temperature case, because the overnight return temperature is lower than the daytime return temperature and thus the tank warm node temperature gets colder as the night progresses, the optimizer will make a trade-off between a lower in T_{wb} and a higher $T_{chwrChillers}$, and find the best start time, which is usually somewhat earlier in the night than the start time in the constant-return-temperature case.

Figure 79: Optimal start time vs charge amount, assuming constant return temp



7.9.2 Possible extensions in future research

In analyzing the annual results, it was found that the chilled water supply setpoint was not always being met (in both the base case and the lookup table case), because the chilled water plant was exceeding its capacity. To avoid this problem, the flow rate must be allowed to modulate. As such, it can be added to the problem, and the subproblem solution grid re-solved. This adds an optimization variable to the subproblem, making it more expensive to solve, but it does not add any dimensions to the conditions grid.

The choice of 1 versus 2 chillers can also be added to the model. Because the number of chillers is a discrete control variable, it cannot be interpolated. So the

subproblem must be solved over this grid twice, once with the number of chillers fixed at 1 and once with it fixed at 2. The subproblem solution can be calculated at any point by interpolating over the two grids independently and then choosing the case with the least energy consumption.

A variable T_{chws} can be added back into the problem by including it as a given in the subproblem, increasing the dimension of the lookup table by one. And the heat exchanger model developed for use in the initial study at Merced would then be built into the main problem as a lookup table (similar to the way it was gridded for the experimental MPC implementations). The main problem would thus be modified such that $\dot{m}_{chwrCampus}$ and $T_{chwrCampus}$ are no longer extraneous inputs to the problem, but rather are functions of T_{chws} and \dot{Q}_{campus} , calculated by the campus return temperature model in the Appendix.

This study has varied the T_{cws} from night to night, but has kept it constant over each night. It should ideally be allowed to vary in real-time (if the physical control implementation allows for it). The chiller flow rate should ideally also be treated this way (if allowed). As such, these variables could be removed from the MPC problem (making it easier to solve) and treated as a separate optimization to be carried out at a lower level, leaving just the charging start and stop times (and T_{chws} if applicable) to be solved in the subproblem.

Various annual simulations could be run with different horizon lengths, as well as with different hourly loads and weather data, different storage capacities and with different levels of model mismatch and prediction inaccuracies. The same general configuration could also be used with a different gridded plant model and a different campus loads model, and thus could be used for other campuses by simply re-calculating the lookup table with the new plant model.

8 Detailed Case Study #2: Integrated Facade and HVAC Control

8.1 Case description

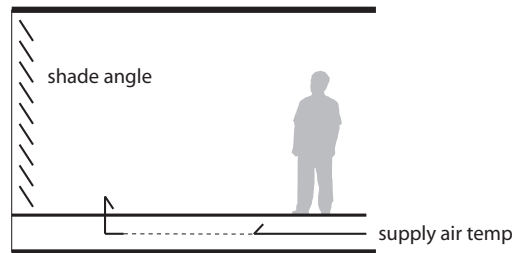
8.1.1 Background

This case study considers the integrated control of an underfloor air distribution (UFAD) system and an automated shading system. Automated shades and UFAD systems are usually controlled independently. The question at hand is whether there might be any energy savings available through integrated control of the two systems.

8.1.2 Problem definition

Figure 80 shows the two control setpoints considered in this study. The problem is considered for two cases: one with internal shades and the other with external shades.

Figure 80: Shading and HVAC control variables of interest



The optimization must consider various complexities, including:

- the general tradeoffs between thermal loads and lighting energy for different shade positions
- the effects of solar gains to the floor on thermal decay in the UFAD system, which effects supply air flow rates and feasible supply air temperature ranges, and thus cooling COP and the potential for economizer operation
- capturing the solar gains at the shades creates a plume that drives thermal stratification in the zone, which affects cooling COP.

8.2 Model description

An EnergyPlus version 6.0 model was constructed with a core zone and a perimeter zone with a UFAD system and automated shades (Figure 81). Table 12 shows some of the key model parameters and their values in the model.

Figure 81: EnergyPlus model for case study

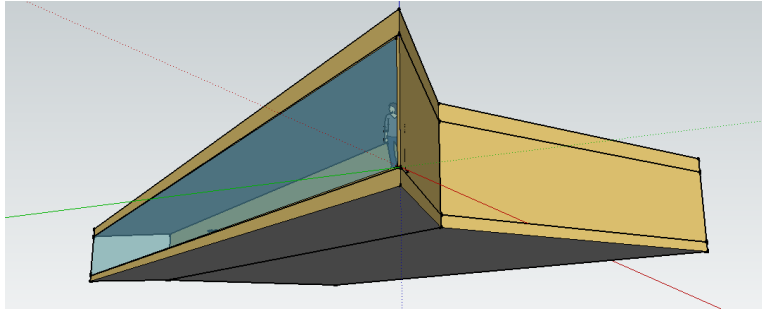


Table 12: Some key model parameter values

lighting power density	4.0 W/m^2
equipment power density	10.0 W/m^2
people density	15 $m^2/person$
chiller COP at rated conditions	3.0
pump total efficiency at rated conditions	62%
supply air fan efficiency at rated conditions	70%
cooling tower fan power at design air flow rate	1 kW

The control setpoints are the blind angle (allowed to vary in the range of 90 degrees (horizontal) to 10 degrees (closed)) and the supply air temperature (allowed to vary in the range of 12-16°C). In the base case for comparison, the control values are kept constant at a 90 degree blind angle and a 12°C supply air temperature.

Figure 82 and Table 13 show the annual energy consumption by end use for the current model configuration with the base case control. (The ‘heat rejection’ category captures the energy consumption of the cooling tower, while the ‘cooling’ category covers just the chiller.)

Figure 82: Annual energy basecase, end-use breakdowns

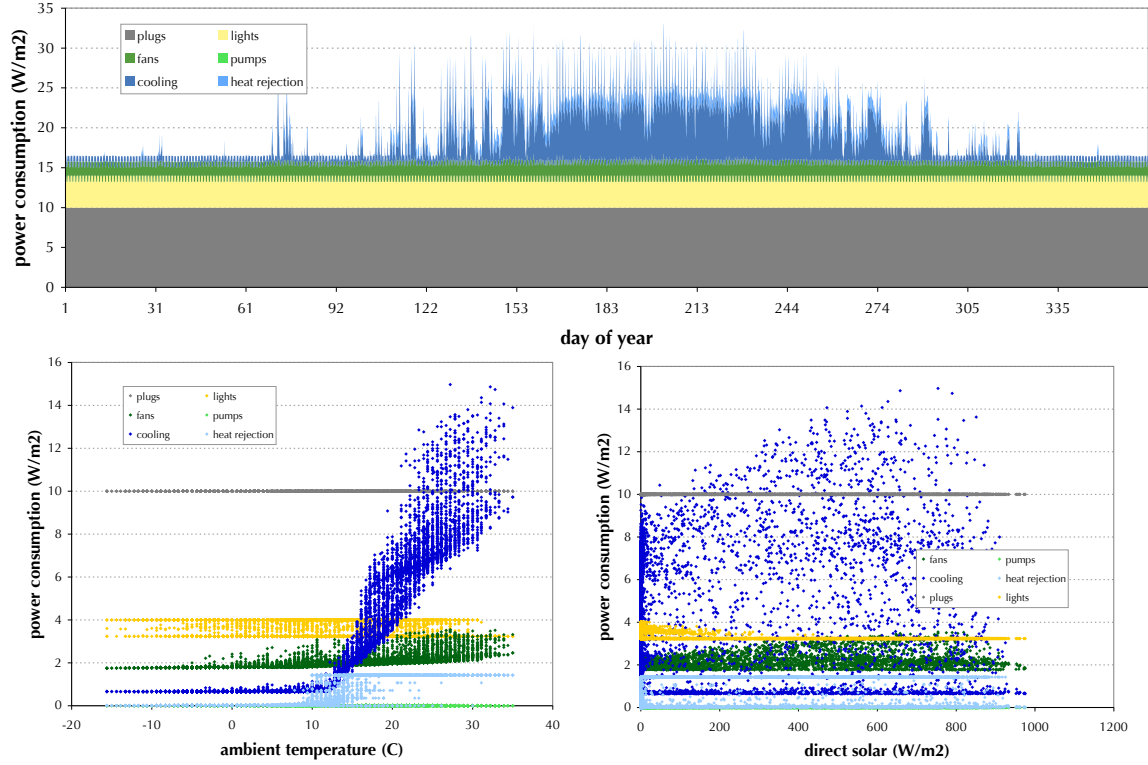


Table 13: Base case annual energy consumption by end use, W/m^2

	total	lights	plugs	fans	pumps	cooling	heat reject.
base case	170.92	31.79	87.60	17.65	0.03	27.74	6.11
% of total		19%	51%	10%	0%	16%	4%

The EnergyPlus model uses the EPMacro language extensively, structured as a set of include files (e.g. different files for geometry, internal loads, HVAC distribution, etc) and one main imf file that references these files and also lists key global parameters, such as the control setpoints. This facilitates model debugging and makes parametric analysis, calibration and optimization easier to configure.

8.2.1 Assumptions and overview

Because the building envelope is primarily glass and the zone temperatures are kept relatively constant, it was assumed that thermal mass had little effect on the optimal values of the blind position and supply air temperature. Neglecting thermal mass simplifies the control optimization analysis significantly because it eliminates the need for a prediction horizon.

Five conditions variables have been chosen a priori for the grid, as follows:

- ambient temperature
- direct beam radiation
- day of year
- diffuse horizontal radiation
- time of day

The day of year and time of day variables are used primarily to capture solar position, but may also be used to estimate internal loads in the model with calibrated schedules. Note that the EnergyPlus weather files have many more variables than just the five conditions listed above. The remaining variables are estimated as functions of these five conditions, as described in the conditions parametrization section below.

8.2.2 Model configuration for control optimization

Although EnergyPlus is generally used for annual simulations, it may also be used for simulation lengths as short as one day. In this analysis, the simulation would ideally only be run for one timestep (usually 15 minutes or less). But since this is not possible, it is run for one day and the objective function output is limited to just the hour of interest by using a schedule in the output definition. The month, date (derived externally by a pre-processing step in SimCon) and hour of day parameters are thus listed in the main imf file and used by the EPMacro language to define the start and stop days for the run period and the schedule used in the output.

The main imf file of the EnergyPlus model also contains the two control variables, which are used by the EPMacro language to set these values in the appropriate places in the model. The GenOpt and SimCon variables are thus demarcated at the top of the main imf file as follows:

```
##set1 SupplyAirTempVal = %supplyAirTemp%
##set1 BlindsAngle = %blindAngle%
##set1 MonthNum = $monthNum$
##set1 DayNum = $dayNum$
##set1 HourNum = $hourOfDay$
```

The EnergyPlus output reports are limited to just one variable:

```
Output:Variable,*,Total Electric Demand,daily,ReportSched;
```

The weather file, on the other hand, requires more significant modifications. In addition to the monthNum and dayNum values, there are 14 weather variable values that must be entered in the weather file, as shown below. The parametrization to go from the 5 conditions variables to these 14 weather inputs is described in the next section.

```
LOCATION,New York Central Prk Obs Belv,NY,USA,TMY3,725033,40.78,-73.97,-5.0,40.0
...
DATA PERIODS,1,1,Data,Sunday, $monthNum$/$dayNum$, $monthNum$/$dayNum$
...
1987,$monthNum$, $dayNum$,6,0,[long flag value],$Tamb$,$Tdp$,$RH$,101500,$EtHorRad$, $EtDirNorRad$, ...
$HorIRsky$, $GlobalHor$, $DirectNorm$, $DiffuseHor$, $GlobalHorIll$, $DirectNormIll$, $DiffuseHorIll$, ...
$ZenithLum$,190,$WindSpeed$,0,0,40.2,77777,9,999999999,110,0.243,0,88,999,999,99
```

8.3 Conditions parametrization

Many of the EnergyPlus weather variables are closely coupled, such as the diffuse horizontal radiation (W/m^2) and the diffuse horizontal illuminance (lux), where one may be reasonably well approximated by a linear correlation with the other. With this in mind, a spreadsheet tool is used to graph and linearly correlate any particular weather variable with each of the 5 chosen conditions variables, as shown in Figure 83. In this particular screenshot, the horizontal infrared radiation (W/m^2) is compared against (from left to right and top to bottom) the day of year, hour of day, ambient temperature, direct normal radiation and diffuse horizontal radiation. The horizontal infrared radiation shows a good correlation with the ambient temperature (shown in closer detail in Figure 84), so the linear curve fit relating these two variables is used in the conditions pre-processor in SimCon to derive the value for infrared horizontal radiation given the ambient temperature. For cases where none of the conditions variables correlate well with the particular weather variable (as is the case with wind direction, shown in Figure 85), then the average value for that weather variable is used throughout. Table 14 summarizes the correlations that are used in the implementation.

Figure 83: Screenshot of hourly weather parametrization, horizontal infrared radiation

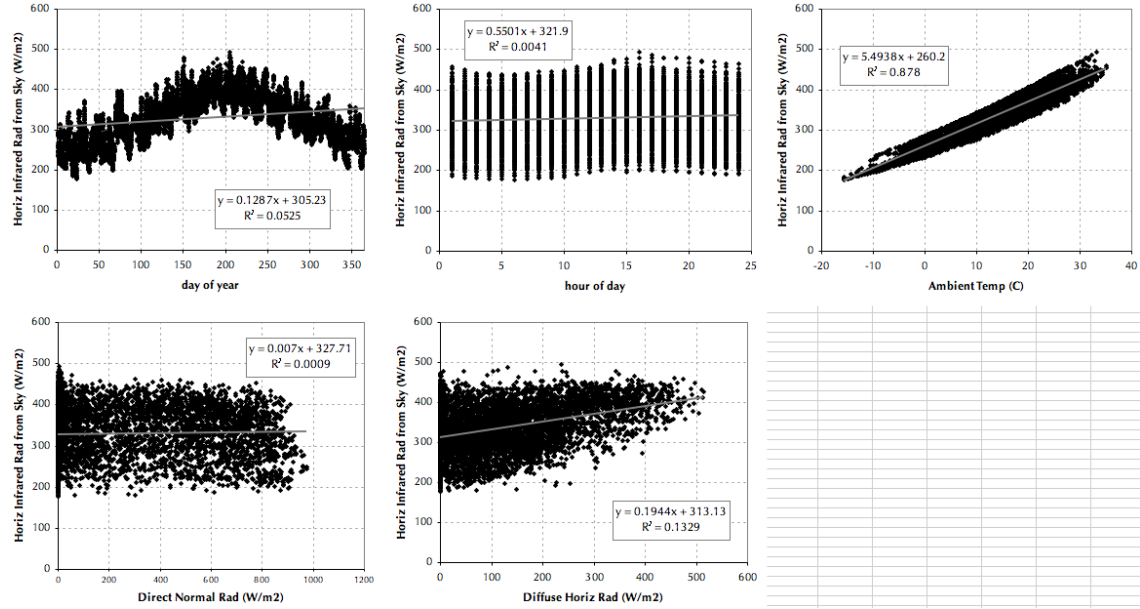


Figure 84: Horizontal infrared radiation versus ambient temperature

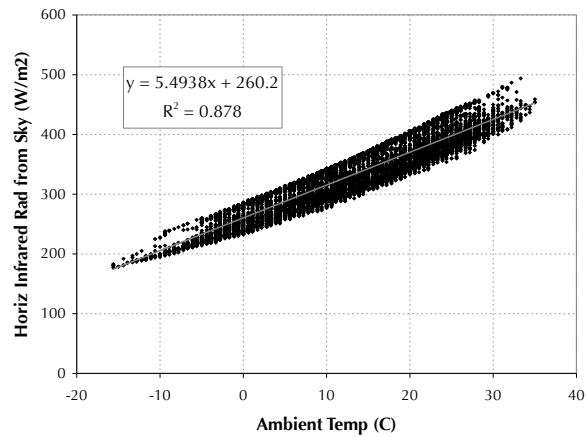


Figure 85: Screenshot of hourly weather parametrization, wind direction

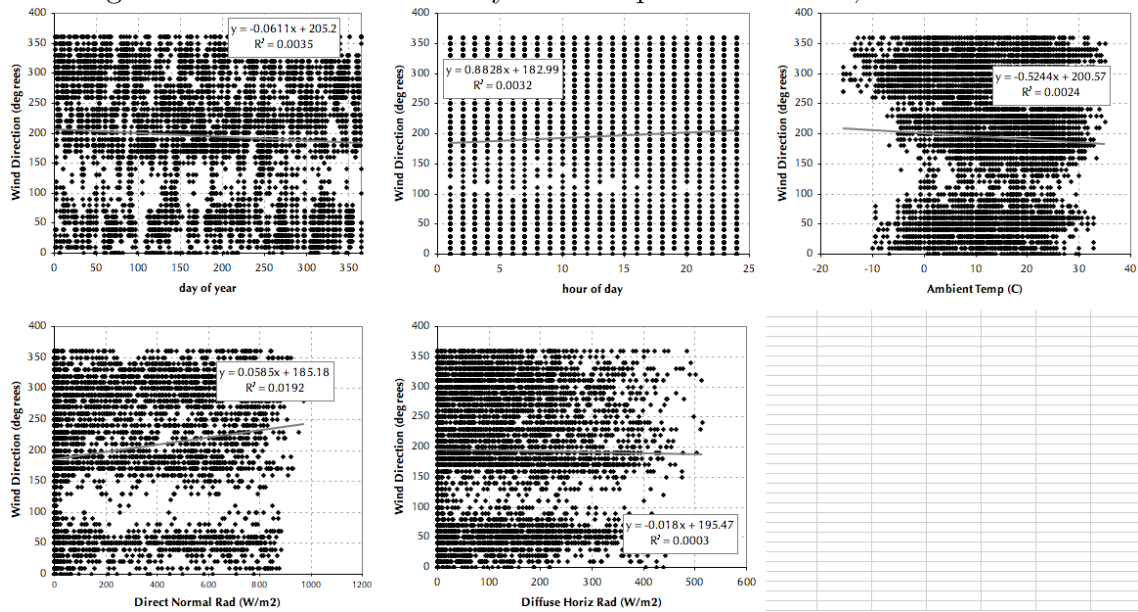


Table 14: Parameter correlations

variable	correlate	equation	R^2
Tdp	Tamb	$y = 1.0082x - 7.0768$	0.8557
RH	use Tdp	psychrometric function	
Extraterr Horiz Rad (Wh/m2)	Diffuse	$y = 3.3948x + 53.85$	0.8516
Extraterr Direct Normal Rad (Wh/m2)	hr	$y=1400$ if $7 < hr < 18$, 0 otherwise	
Horiz Infrared Rad from Sky (Wh/m2)	Tamb	$y = 5.4938x + 260.2$	0.878
Global Horiz Rad (Wh/m2)	Direct	$y = 0.7989x + 47.26$	0.7039
Global Horiz Ill (lux)	Direct	$y = 83.749x + 5397.5$	0.6804
Direct Normal Ill (lux)	Direct	$y = 98.645x - 312.56$	0.9931
Diffuse Horiz Ill (lux)	Diffuse	$y = 113.14x + 216.25$	0.9962
Zenith Luminance (Cd/m2)	Diffuse	$y = 44.955x + 63.426$	0.4138
Wind Direction (degrees)	average	190	
wind speed (m/s)	Tamb	$y = -0.044x + 5.9043$	0.0393
Sky cover	average	4	
Opaque sky cover	average	4	
Visibility (km)	average	15	
Ceiling height (m)	average	77777	

This process can be improved for some variables in future iterations, by considering higher-order and multi-variable correlations. For example, the global horizontal radiation could be directly calculated from the diffuse horizontal and the direct beam by using the solar angle, which can be derived from the time of day and day of year.

8.4 Optimization configuration

The multi-dimensional grid used for the disturbances is as shown in Table 15. This grid configuration results in 1500 sets of disturbances for which optimal control was to be determined. The Hookes-Jeeves algorithm in GenOpt was used, with 1 step size reduction, and non-parallelized, so running on just one processor. The sequential optimizations required approximately 24 hours on a single Windows virtual machine.

Table 15: Conditions grid

	min	max	spacing
day of year	1	183	91
hour of day	9	15	3
ambient temperature (C)	5	30	2.5
direct beam radiation (W/m^2)	0	400	100
diffuse horizontal radiation (W/m^2)	0	400	100

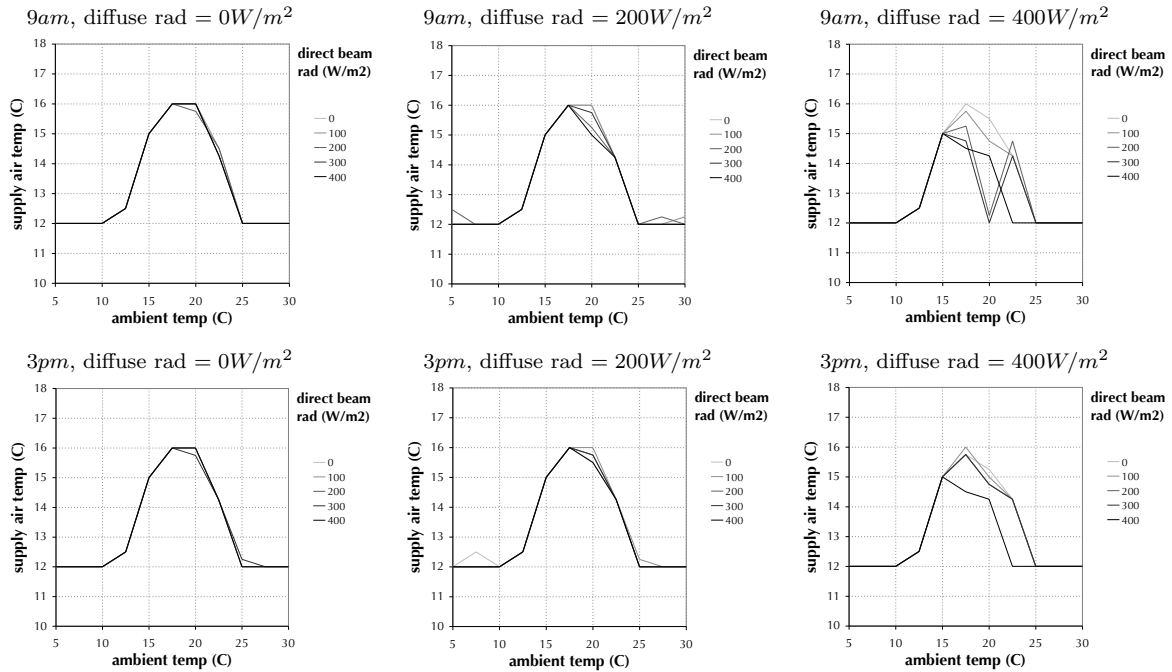
8.5 Lookup table results

8.5.1 Interior blind case

Figures 86 through 88 show some of the many possible slices through the calculated lookup table (day = 183 in graphs). Figure 86 shows the optimal supply air temperature as a function of ambient temperature and direct radiation, for various values of diffuse radiation and time of day. The shape of the curve when the ambient temperature is greater than or equal to $17.5^{\circ}C$ is as expected, with the supply air temperature at or near the upper bound when the ambient temperature is $17.5^{\circ}C$, and decreases to its lower bound with the ambient temperature is at or above the zone temperature setpoint. The specifics of this part of the curve depend on the solar gains variables. The shape of the curve when the ambient temperature is less than $17.5^{\circ}C$ is less intuitive, but it is reasonable: the supply air temperature setpoint falls in keeping with the ambient air temperature, which is likely because (1) lower supply air temperatures require lower air flow rates and thus lower fan power consumption,

and (2) no cooling energy is required to provide the supply air temperature at the ambient air temperature plus the fan heat-gain delta temperature.

Figure 86: Interior blind case: Optimal supply temperature versus ambient temp.



The optimal blind position, graphed in Figure 87 for the same conditions as those in Figure 86, does not show the same consistency. In particular, it shows a lot of variation when the solar gains are low. The variations at low loads may be because under these conditions the energy use is not very sensitive to shading position. (This variation at low loads should be investigated in greater detail in future research.) But the general trends are as expected, with the blinds tending towards closed when it is hot and sunny, and tending more towards open when it is cold or less sunny. Figure 87 shows the optimal blind position as a function of the direct and diffuse solar gains - here it shows somewhat less variation, and tends towards closed when either the direct or diffuse gains are high, and tends towards open when both are low.

Figure 87: Interior blind case: Optimal blind angle versus ambient temperature

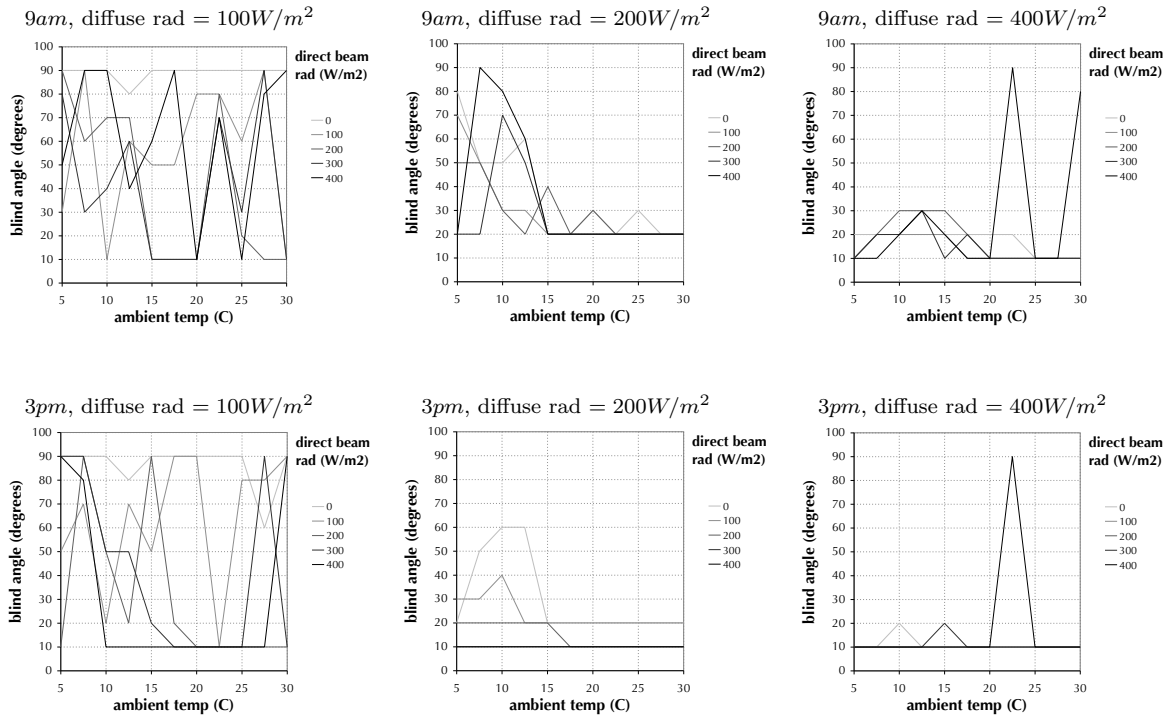
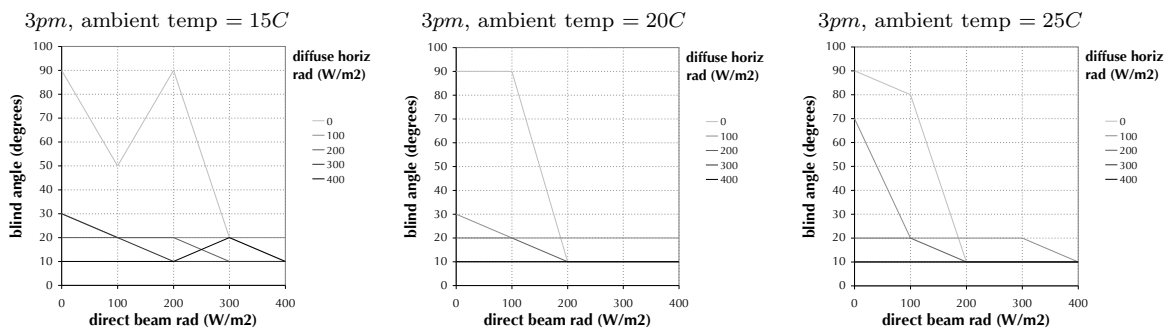


Figure 88: Interior blind case: Optimal blind angle versus direct and diffuse solar values



8.5.2 Exterior blind case

Figures 89 through 91 repeat the previous figures but for the exterior blind case. Note that the optimum supply air temperature curve is more consistent in this case, deviating only very slightly from a constant curve shape versus ambient temperature. The optimal blind position is still highly variable at low solar loads, but the trends towards closed under higher solar gains is stronger, and the results also show the optimal blind position to be more responsive to the ambient temperature than it is in the internal blind case.

Figure 89: Exterior blind case: Optimal supply temperature versus ambient temperature

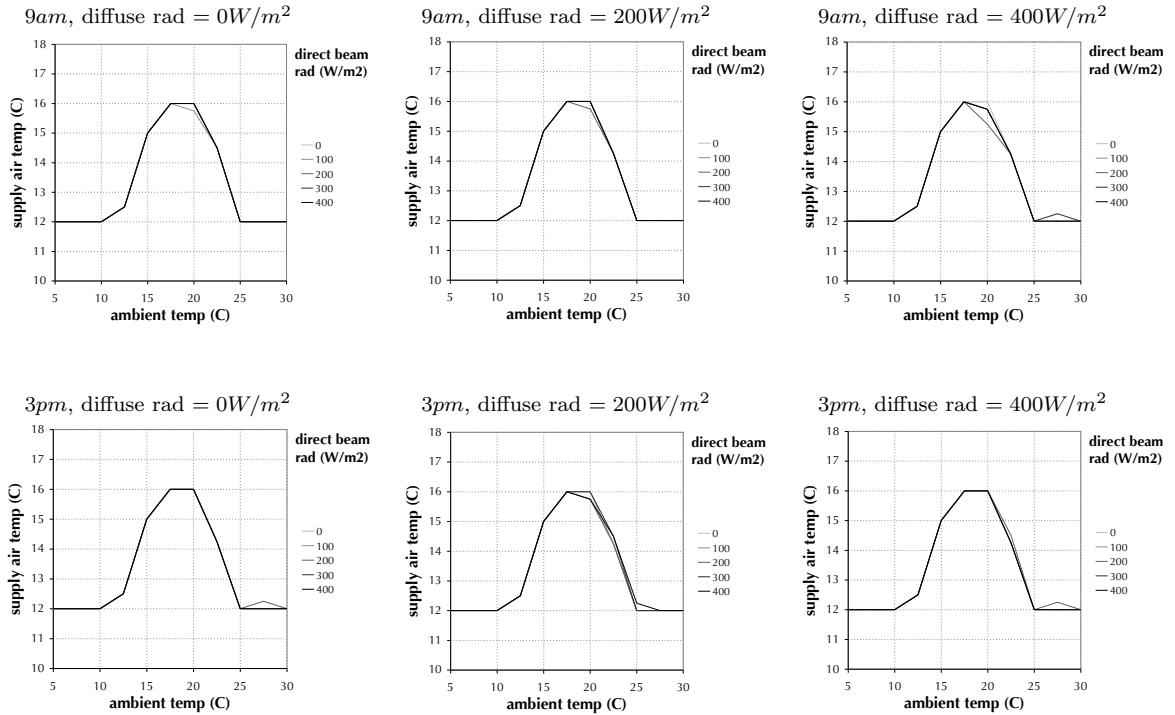


Figure 90: Exterior blind case: Optimal blind angle versus ambient temperature

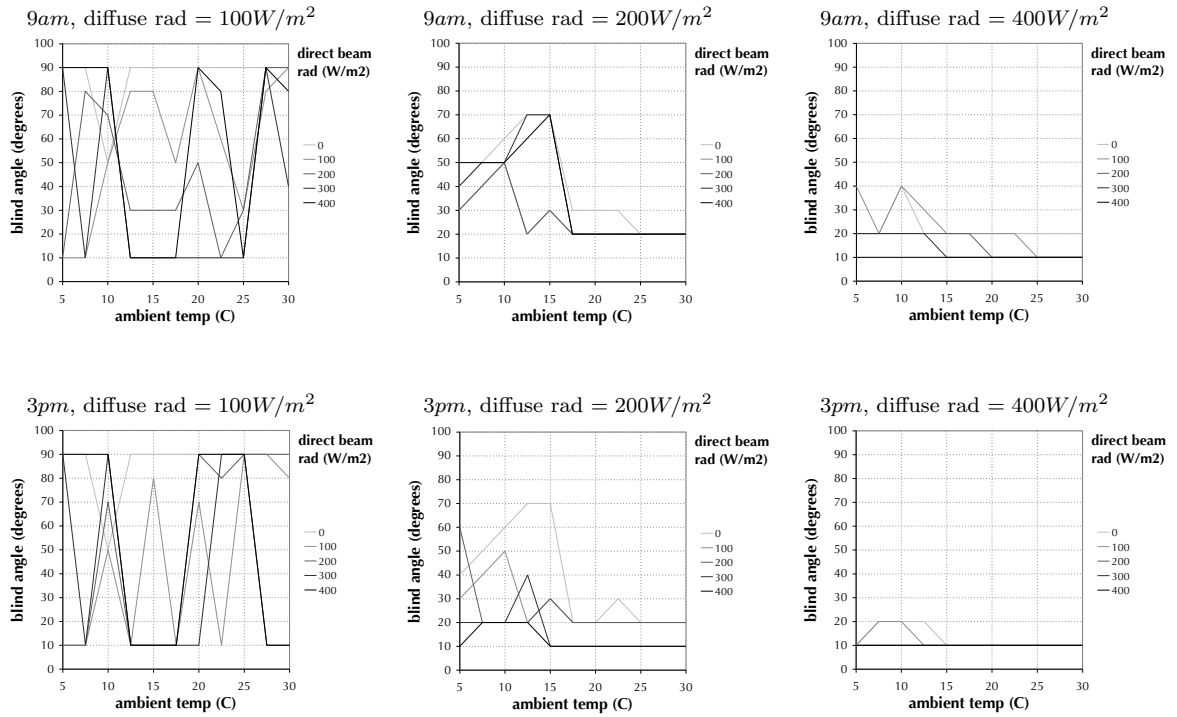
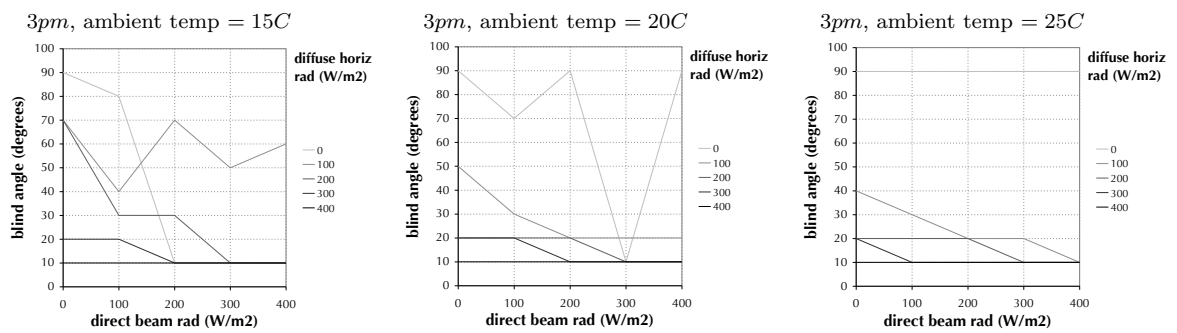


Figure 91: Exterior blind case: Optimal blind angle versus direct and diffuse solar values



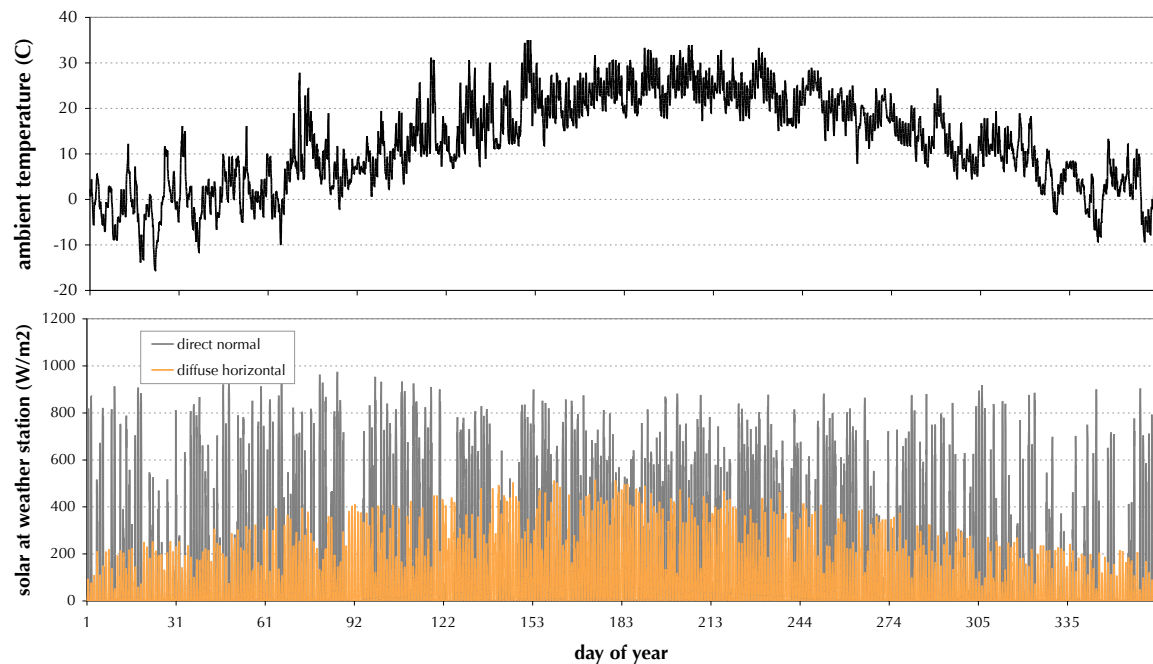
8.6 Annual simulations

8.6.1 Annual simulation configuration

To test the lookup table control, its use is simulated through the Building Control Virtual Test Bed (BCVTB). The ‘real building’ simulation model in EnergyPlus receives its inputs through the Schedule:ExternalInterface object in the main imf file. Otherwise, the EnergyPlus model is identical to the one used in the optimizations to derive the lookup table (although in future iterations the effects of model mismatch could be investigated by varying parameters in this annual simulation model). The timestep of the BCVTB simulation is 15 minutes.

The annual simulations were run with the NY Central Park TMY3 weather file. The annual ambient temperatures and solar radiation are shown in Figure 92.

Figure 92: Annual weather variables



8.6.2 Annual simulation results: Interior blind case

Figure 93 shows the control setpoints over the course of the year, and Figure 94 shows a more detailed view of just the month of March. The x-y plots (Figure 95) of the control values versus the ambient temperature are somewhat more informative. The supply air temperature behaves as expected, and the blind angle generally trends towards closed when the ambient temperature is higher and open when it is colder, but there is more variation than expected.

Figure 93: Annual control outputs

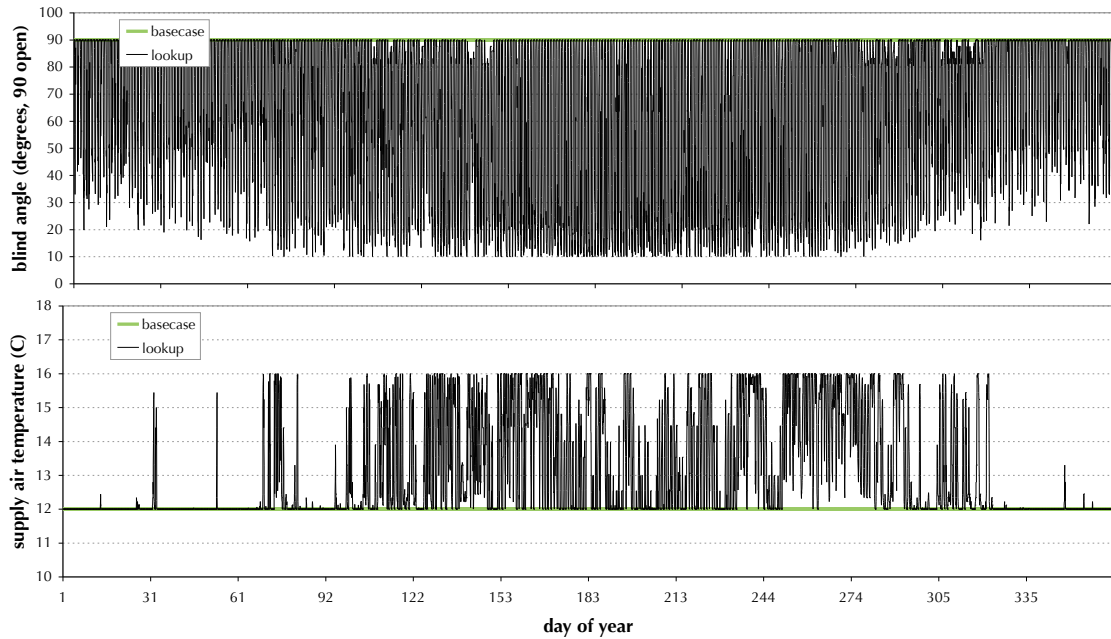


Figure 94: March control outputs

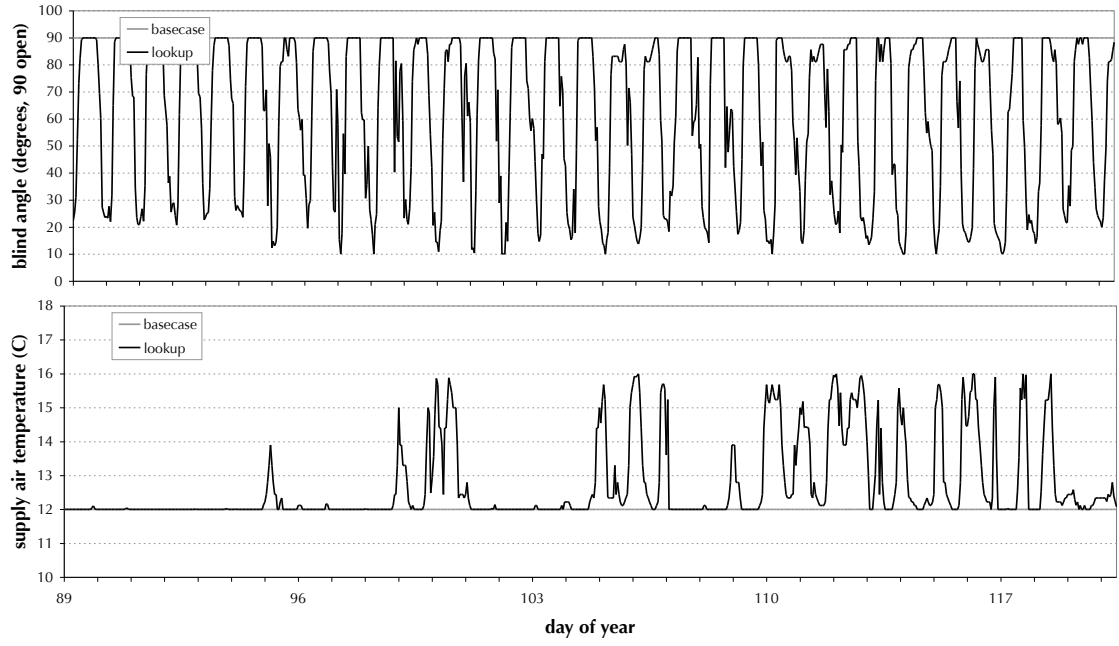


Figure 95: Hourly control outputs vs ambient temperature

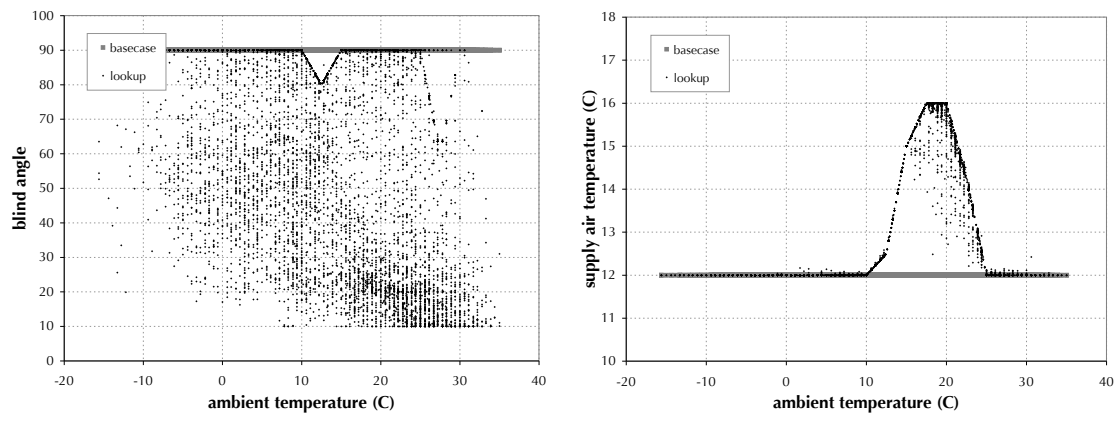
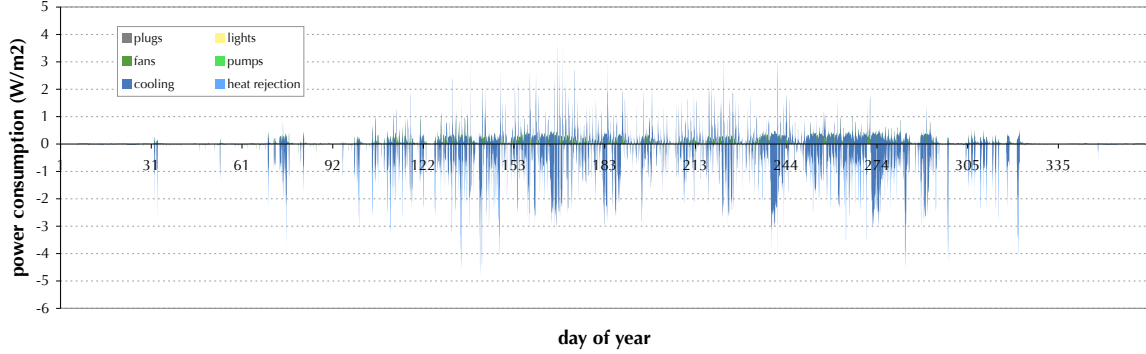


Figure 96 shows the energy savings of the lookup table control case over the base case. In general, the lookup table case is using slightly more fan power, but is saving more cooling energy.

Figure 96: Annual energy savings, end-use breakdowns



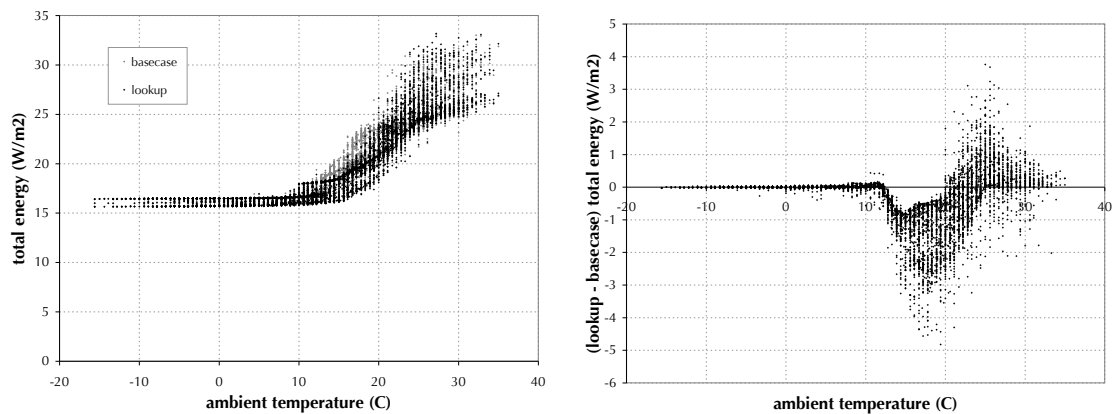
The annual energy consumption is summarized in Table 16. The lookup controller is generally decreasing cooling energy compared to the base case by using more shading, and is trading off higher fan energy consumption for lower cooling energy by increasing the supply air temperature. There is only a very slight (less than 0.005%) increase in lighting energy in the lookup case. It was suspected that this may be pointing to an error in the daylighting controls in the model, but it was thoroughly checked and was working properly. It seems that the lookup table controller is avoiding increasing lighting use in the way that it is controlling the blind, which would make sense if the cost of increasing lighting use always outweighs whatever cooling savings may be gained by increasing shading.

Table 16: Interior blind case: Annual energy consumption, W/m^2

	total	HVAC	lights	plugs	fans	pumps	cooling	heat reject.
basecase	170.92	51.53	31.79	87.60	17.65	0.03	27.74	6.11
lookup	168.37	48.98	31.79	87.60	18.54	0.03	24.72	5.69
savings	2.55	2.55	0.00	0.00	-0.88	0.00	3.01	0.42
saving %	1.49%	4.95%	0.00%	0.00%	-4.99%	0.56%	10.87%	6.83%

As with the control values, it is often instructive to view x-y plots of the energy consumption versus the ambient temperature (and against other variables). Figure 97 suggests that the control is not making much difference relative to the baseline when the temperature is less than 10C, and is generally saving energy in the 10-23C range, but seems to be often performing worse than the base case in the 23-30C range. More detailed investigation of the hourly outputs reveals that these times of poor performance usually occur at times when the control setpoints for the supply air temperature are changing quickly - this loss in performance is thus likely because the MPC model is not considering the thermal mass of the air in the UFAD zones. Any further extensions of this research should investigate this further, and consider including a state variable for the UFAD zone temperature and using a prediction horizon of a few hours.

Figure 97: Hourly energy consumption vs ambient temperature



8.6.3 Annual simulation results: Exterior blind case

The results for the exterior blind case are very similar to the interior blind case, but with slightly higher energy savings compared to the baseline. Table 16 shows the details; the annual HVAC energy savings were 5.56%, rather than the 4.95% in the interior blind case.

Table 17: Exterior blind case: Annual energy consumption, W/m^2

	total	HVAC	lights	plugs	fans	pumps	cooling	heat reject.
basecase	168.95	49.56	31.79	87.60	17.25	0.03	26.20	6.08
lookup	166.20	46.80	31.79	87.60	18.07	0.03	23.06	5.64
savings	2.75	2.76	0.00	0.00	-0.82	0.00	3.14	0.44
saving %	1.63%	5.56%	0.00%	0.00%	-4.77%	0.58%	11.99%	7.19%

8.7 Discussion

This study could be extended in various ways in further research. The controller performance loss due to the thermal lag of the UFAD zones should be investigated further and a state variable and prediction horizon should be added to avoid this problem; an interesting follow-up research question would be to look at how much performance improvement is gained relative to the increase in controller complexity. The study could be repeated for other climates and with other design parameter values (such as the reflectivity of the roller shades or the use of active external shading or changes to the building geometries or materials), to search for the most promising application configurations. In addition, the controller could be redesigned to consider a case where the change in blind position over time is constrained such that the movements are not perceptible by the occupants, and the goal is still to minimize glare and total energy consumption over the day (which would thus require prediction).

9 Discussion

9.1 Lessons learned from the case studies

The case studies have considered a variety of problems and simulation tools, providing good testing grounds for refinement of the methods and analysis of their effectiveness. In general, the case studies have highlighted the importance of iteration when using this approach: start with a lower resolution grid and use it to debug the model and optimization configuration, then re-run the optimizations and build up to a higher resolution only when comfortable with the set up.

Each of the case studies also highlighted particular aspects of the control approach. The abstract case study #1 (solar shading and natural ventilation) showed that the impact of grid size on controller performance is not necessarily straightforward, and suggested that irregular grids should be considered in future research (as discussed below).

The abstract case study #2 (massive slab pre-cooling) allowed for the demonstration and testing of various details of conditions parametrization over a prediction horizon, and provided a test case for the approach with EnergyPlus. In general, EnergyPlus can be used effectively for cases with just one or two state variables of interest, but it is more challenging to use for this than tools like Modelica or TRNSYS that allow for explicit state initialization; having to use a work-around to approximate initial state variable values is onerous and problematic, and is what keeps EnergyPlus from being useful when more than a couple of state variables are of interest. The three different online MPC configurations proved useful in teasing apart the performance impacts of grid spacing, disturbance parametrizations and initial state parametrizations; the three configurations may be useful in any future research to further quantify the performance impacts of these three different aspects of the approach.

The abstract case study #3 (cogeneration dispatch) showed the use of problem decomposition, as intended, but also strengthened the argument for using irregular grids. It also pointed to the potential use of analytical solutions for control switch points to augment the performance of the lookup table approach in some cases.

The two detailed case studies (chilled water storage control and integrated shading-HVAC control) provided tests of the approach with more detailed modeling and real-world systems, grounding the investigation by ensuring that the methods can be applied to more than just abstract problems. The chilled water storage study also provided an opportunity to wrap the controller into a human-in-the-loop interface and connect it to an online source of weather prediction.

9.2 Evaluation of the approach

9.2.1 Range of feasibility and suitability

The figure in the Methods chapter that relates grid size to computational time (Figure 4) suggests that the approach is only feasible for problems that can be simplified to 5-6 dimensions (potentially slightly more if one's computation budget is higher). The case studies have shown that a wide variety of building control problems can be effectively simplified to this size through conditions parametrization and problem decomposition, with reasonably small performance losses relative to online MPC and heuristic rule-based control.

Two of the five case studies involved a prediction horizon, the other three were steady-state problems. For the two case studies with prediction, the horizons were relatively short. Longer prediction horizons would likely make the problems infeasible by ballooning their dimensionality. In general, the approach is more often feasible for steady-state problems than for predictive problems, but as shown in those two case studies this is not strictly so. Further parametrizations or other developments may make feasible predictive problems with longer horizons than the ones considered herein.

The case studies demonstrated the use of the approach with Modelica, TRNSYS and EnergyPlus. Modelica and TRNSYS are better suited to the approach because of their greater flexibility in model construction, shorter simulation time-steps, more variable simulation lengths, and explicit state initialization. But EnergyPlus was also shown to be effective for some problems. It could be made more broadly effective if future versions of the program allow for explicit state initialization and shorter run-periods than one day. Given that EnergyPlus can be used with the approach, it is likely that DOE-2 could also be used. Similarly, most other simulation tools could likely be used, given that they share common characteristics with at least one of three tools tested.

The case studies have also shown that the approach is better suited to control problems whose solutions are generally smooth and continuous, rather than problems whose solutions have many discontinuous jumps. However, it could be made better at the latter through more sophisticated sampling techniques.

9.2.2 Benefits

Although the approach was first investigated because of its potential to simplify implementation in existing control systems, the most beneficial aspect of it seems to be the feedback it can provide to the design process. Graphs of slices through the

lookup table provide the system designer or analyst with a way of visualizing the optimal control response for the system. If the control response is nearly always on a constraint boundary, then the system designer may want to loosen that constraint if possible. On the other hand, if the control response never ventures into particular areas of the control space, then the system designer may want to eliminate that flexibility.

The lookup table slices graphs also provide a very useful debugging tool for the controller development process. As such, the approach may also provide online MPC developers with a good way of testing and debugging their implementations.

The lookup table visualizations and annual simulated implementations can also help system designers to estimate the potential for energy savings with near-optimal control with a particular system. This can help system designers to more accurately compare different systems in terms of energy savings and payback periods. In some cases, this approach may also be useful in estimating potential savings with a full online MPC configuration with a given system, thus aiding in the decision of whether or not to investment in more advanced controls for the system.

In general, the approach is well suited to existing processes in building design and operation: building designers or consultants could develop lookup tables using their design models and their own processing power or cloud computing, and then hand over lookup tables with interfaces to building operators or have them implemented directly in buildings' control systems. It could thus offer improvements to the control of both new and existing buildings. The approach could also be useful for design studies, including analyses of which building elements (e.g. external shades, operable windows) in a particular design are more important as dynamic elements, and which ones could be designed as static elements without much loss in performance. Additionally, the approach could fit in well with other uses of simulation during building operations. The same model used in the controller could be used for benchmarking, fault detection and diagnosis, and for retrofit analysis, and this single model could be updated with changes to the building or periodically re-calibrated to account for building and system deterioration over time. Updates to the model would then require re-running the optimizations over the conditions grid, but this could be easily automated.

For research purposes, the approach could be used to help determine the energy savings potential of integrated dynamic systems in buildings in general, and it could be used for the development of better simplified control rules for common systems of particular types of buildings.

9.2.3 Challenges

The biggest challenge at this point is in streamlining the approach and making it accessible to practitioners. The methods may need to be simplified somewhat in order to become commonly used. A good user interface will be required to walk people through the process. For most practical problems, cloud computing is necessary for the problem to be solved in a reasonable amount of time - a good interface will have to account for this and facilitate the process of partitioning the grid and sending it to cloud machines for processing.

9.2.4 Prospects for market uptake

If a good interface can be developed, the methods as they currently stand could likely find their way into use at a handful of consulting firms, as an additional service offering for some projects. Chilled water storage systems and cogeneration systems are the most likely systems to see practical benefits from this approach. If such initial applications prove successful, and/or if the methods and interface can be simplified, then it could conceivably find its way into more design firms and more applications, particularly if building design continues to trend towards more complex and integrated systems.

If the approach were to see significant market uptake, it is likely that it would become more heuristic in nature and less academic, with the optimization approach used to inform aspects of otherwise heuristic control laws, or heuristics used to fill in gaps in the optimization or to speed up the process.

9.3 Areas to consider in future research

As noted previously, an important area for future research is the use of irregular grids. Various sampling techniques (such as adaptive grids, or even random sampling) should be considered, and they should be evaluated not only for the resulting controller's performance, but also for the parallelizability of the technique - a fully serial sampling method is unlikely to be appropriate for this use. The integration of user-supplied heuristics for control switch points, as discussed at the end of Chapter 6, should also be considered in future research.

Other areas to consider in future research include the use of nonlinear interpolations, and how best to deal with uncertainties. The use of optimization starting points based on solutions to neighboring grid points should also be considered as a way to speed up the optimization process. Again, successful techniques must ensure that the approach remains massively parallelizable, so that the conditions grid can

still be solved quickly through the use of many processors available through cloud computing services.

A good idea gained from feedback by researchers at NRCan-Varenes (personal communication, 2010) is to find ways of structuring optimization configurations such that variables that change often (e.g. electricity rate structures) can be exposed either as conditions variables or as the online MPC parts of decomposed problems, while aspects that are more likely to remain unchanged are embedded in the offline solution.

The other main area in need of future research and development is to devise appropriate software interfaces to facilitate the use of this approach in research and practice, to study its integration into practice (possibly through interviews with users or potential users), and to integrate this software and methodology with other operations-phase uses of building simulation tools.

10 Conclusions

A method for near-optimal supervisory control using building simulation tools and offline optimization is outlined herein. Through case studies, the approach is shown to be feasible for problems that can be suitably expressed as a function of 5-6 conditions variables (or slightly more, depending on one's computation budget), or for problems that can be broken down into subproblems of this size. The case studies demonstrated the use of the approach with Modelica, EnergyPlus and TRNSYS, and considered the following application areas: integrated solar and natural ventilation; massive slab pre-cooling; cogeneration dispatch; chilled water storage; and integrated solar and HVAC control.

Three abstract case studies were used to analyze particular aspects of this approach. The first considered the effect that grid spacing has on controller performance. The relationship is complex and case-dependent, but the methods of investigation used herein could be useful in finding appropriate grid spacings in other case studies. The study also pointed to the importance of considering irregular grids in future research. The second abstract case study analyzed the performance losses associated with conditions parametrization methods: for this case they were found to be small. Simple heuristic baseline controls and full online MPC were used as comparison points in these two case studies, with the available savings defined as the difference between the baseline and the full MPC.

In the first case study, the lookup table control captured 83%-94% of the available savings (depending on the grid spacing). In the more complex case #2, with a coarse grid spacing, the lookup table control captured 59% of the available savings.

Problem decomposition was demonstrated in the third abstract case study, and was found to make an otherwise infeasible problem easily solvable, and produced a 5% faster payback on cogeneration system investment, relative to a simple base case controller.

The inability to explicitly specify initial states in many building simulation tools remains a problem. This should be fixed by simulation tool developers. In the meantime, for controls problems with one or two important state variables, the workarounds for EnergyPlus described in Chapter 5 may be used; for larger problems, users are recommended to use simulation tools that do allow for explicit state initialization.

Open-source software to facilitate the use of building simulation tools both in online MPC and in the modified approach described herein is available online. The promise of this software and approach is that it might fit well within existing design and operation processes, with visualization of control responses over the conditions

space providing important feedback during the design process, and the resulting controls could be easy to implement within existing building control systems. Further research, software interface development and early-adopter implementation is needed to move in that direction.

Designing good controls for low-energy building systems is challenging. It is hoped that the concepts and tools described herein will help designers and operators get closer to optimal performance.

References

- K Ackerly and G Brager. Occupant response to window control signaling systems. *Center for the Built Environment Summary Report, University of California, Berkeley*, 2011.
- M Apte, M Sohn, MA Piette, P Berkeley, D Black, P Price, M Najafi, S Narayanan, R Brahme, Z O’Neill, Y Lin, M Spears, A Surana, A Cerpa, V Erickson, A Kamthe, B Eisenhower, and I Mezic. Energy performance visualization and occupancy-based energy management systems for buildings: Implementation and testing at the university of california, merced. *Final report to the US Department of Energy and California Energy Commission-Public Interest Energy Research, LBNL*, 2010.
- ASHRAE. *Handbook of Fundamentals*. ASHRAE, 2005.
- G Augenbroe. Trends in building simulation. in *Advanced Building Simulation (ed. A Malkawi and G Augenbroe)*, pages 4–24, 2003.
- A Bemporad, F Borrelli, and M Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Transactions on Automatic Control*, 47, 2002.
- M Brambley, P Haves, SC McDonald, P Torcellini, D Hansen, DR Holmberg, and KW Roth. Advanced sensors and controls for building applications: Market assessment and potential r&d pathways. *PNNL-15149, Prepared for the U.S. Department of Energy under Contract DE-AC05-76RL01830*, 2005.
- J Clarke, J Crockroft, S Conner, J Hand, N Kelly, R Moore, T O’Brien, and P Strachan. Simulation-assisted control in building energy management systems. *Energy and Buildings*, 34(933-940), 2002.
- B Coffey. A development and testing framework for simulation-based supervisory control with application to optimal zone temperature ramping demand response using a modified genetic algorithm. *M.A.Sc. Thesis, Concordia University*, 2008.
- B Coffey. Simcon download. <ftp://ftp.pwgsc.gc.ca/rpstech/Controls/MPC/SimCon/>, 2011.
- B Coffey and E Kutrowski. Demand charge considerations in the optimization of cogeneration dispatch in a deregulated energy market. *International Journal of Energy Research*, 30:535–551, 2006.

- B Coffey, F Haghghat, E Morofsky, and E Kutrowski. A software framework for model predictive control with GenOpt. *Energy and Buildings*, 42:1084–1092, 2010a.
- B Coffey, P Haves, B Hancey, Y Ma, F Borrelli, and S Bengea. Development and testing of model predictive control for a campus chilled water plant with thermal storage. *ACEEE Summer Study on Energy Efficiency in Buildings*, 2010b.
- EnerSys Analytics. *Energy Profile Tool*.
www.energyprofiletool.com, 2011.
- P Haves, B Coffey, and S Williams. Benchmarking and equipment and controls assessment for a ‘big box’ retail chain. *ACEEE Summer Study on Energy Efficiency in Buildings*, 2008.
- P Haves, B Hancey, F Borrelli, J Elliot, Y Ma, B Coffey, S Bengea, and M Wetter. Model predictive control of hvac systems: Implementation and testing at the university of california, merced. *LBNL TK*, 2011.
- G Henze and M Krarti. Predictive optimal control of active and passive building thermal storage inventory: Final report. *DOE Award Number: DE-FC-26-01NT41255*, 2005.
- G Henze and S Liu. Calibration of building models for supervisory control of commercial buildings. *Proceedings of the 9th International Building Performance Simulation Association (IBPSA) Conference, Montreal, Canada*, 2005.
- G Henze, D 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):153–177, 2004.
- G Henze, D Kalz, S Liu, and C Felsmann. Experimental analysis of model-based predictive optimal control for active and passive thermal storage inventory. *HVAC&R Research*, 11(2):189–213, 2005.
- G 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*, 39:221–235, 2007.
- MJ Holmes. The simulation of heating and cooling coils for performance analysis. *Liege, Belgium: System Simulation in Buildings, Proceedings of the International Conference of the Commission of the European Communities*, 1982.

- G Kelly. Control system simulation in North America. *Energy and Buildings*, 10: 193–202, 1988.
- M Kummert and P Andre. Simulation of a model-based optimal controller for heating systems under realistic hypotheses. *Proceedings of the 9th International Building Performance Simulation Association (IBPSA) Conference, Montreal, Canada, 2005*.
- M Kummert, P Andre, and A Argigiou. Performance comparison of heating control strategies combining simulation and experimental results. *Proceedings of the 9th International Building Performance Simulation Association (IBPSA) Conference, Montreal, Canada, 2005*.
- K Lee and J Braun. Development and application of an inverse building model for demand response in small commercial buildings. *Proceedings of SimBuild*, 2004.
- Y Ma, F Borrelli, B Hancey, B Coffey, S Bengesa, P Haves, A Packard, and M Wetter. Model predictive control for the operation of building cooling systems. *IEEE Trans. Control Systems Technology*, PP(99):1–8, 2010.
- A Mahdavi. Simulation-based control of building systems operation. *Building and Environment*, 36:789–796, 2001.
- A Mahdavi and C Proglhof. A model-based method for the integration of natural ventilation in indoor climate systems operation. *Proceedings of the 9th International Building Performance Simulation Association (IBPSA) Conference, Montreal, Canada, 2005*.
- A Mahdavi, B Spasojevic, and K Brunner. Elements of a simulation-assisted daylight-responsive illumination systems control in buildings. *Proceedings of the 9th International Building Performance Simulation Association (IBPSA) Conference, Montreal, Canada, 2005*.
- D Mayne, J Rawlings, C Rao, and P Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- B Metz, OR Davidson, PR Bosch, R Dave, and LA Meyer, editors. *Climate Change 2007: Mitigation. Contribution of Working Group III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC)*, chapter Summary for policymakers. Cambridge University Press, 2007.

- M Morari and J Lee. Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23:667–682, 1999.
- N Nassif, S Kajl, and R Sabourin. Simplified model-based optimal control of VAV air-conditioning system. *Proceedings of the 9th International Building Performance Simulation Association (IBPSA) Conference, Montreal, Canada*, 2005a.
- N Nassif, K Stainslaw, and R Sabourin. Optimization of HVAC control system strategy using two-objective genetic algorithm. *HVAC&R Research*, 11(3):459–486, 2005b.
- NRCan-Varennes. *Diagnostic Agent for Building Operators (DABO)*. http://canmetenergy-canmetenergie.nrcan-rncan.gc.ca/eng/buildings_communities/buildings/building_optimization/ongoing_commissioning/dabo_software.html, 2011.
- UC Berkeley Dept. of Electrical Engineering and Computer Science. *Ptolemy II*. <http://ptolemy.berkeley.edu/ptolemyII/>, 2011.
- F Oldewurtel, D Gyalistras, M Gwerder, C Jones, A Parisio, V Stauch, B Lehmann, and M Morari. Increasing energy efficiency in building climate control using weather forecasts and model predictive control. *Clima - RHEVA World Congress, Antalya, Turkey*, 2010.
- personal communication. NRCan-Varennes controls research group. verbal feedback after a presentation, October 14, 2010.
- J Petersen, V Shunturov, K Janda, G Platt, and K Weinberger. Dormitory residents reduce electricity consumption when exposed to real-time visual feedback and incentives. *International Journal of Sustainability in Higher Education*, 8: 16–33, 2007.
- PNNL. *Facility Energy Decision System*. www.pnl.gov/feds/, 2011.
- Pulse Energy. *Pulse Energy*. <http://www.pulseenergy.com/>, 2011.
- S Qin and T Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.
- SuPerB and LBNL. *Learn HVAC*. www.learnhvac.org, 2011.

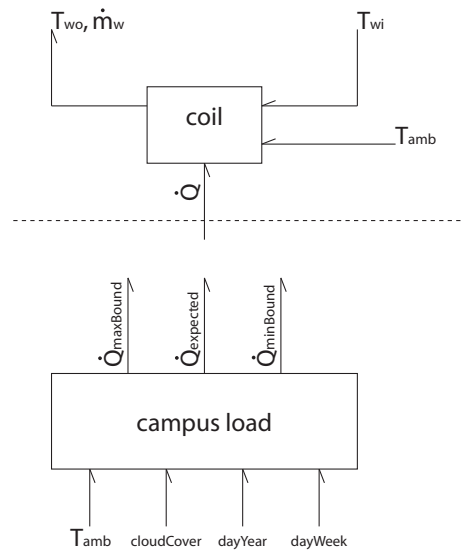
- D Thevenard. Updating the ashrae climate data for design and standards. *ASHRAE Research Project 1453-RP: Final Report*, 2009.
- USDOE. *Buildings Energy Data Book*.
<http://buildingsdatabook.eren.doe.gov>, 2011a.
- USDOE. *Building Energy Software Tools Directory*.
http://apps1.eere.energy.gov/buildings/tools_directory/, 2011b.
- S Wang and X Jin. Model-based optimal control of VAV air-conditioning system using genetic algorithm. *Building and Environment*, 35:471–487, 2000.
- M Wetter. GenOpt: Generic optimization program. <http://gundog.lbl.gov/GO/>, 2009a.
- M Wetter. Simulation model: Finned water-to-air coil without condensation. *LBLN-42355*, 1999.
- M Wetter. Modelica-based modeling and simulation to support research and development in building energy and controls systems. *Journal of Building Performance Simulation*, 2(2):143–161, 2009b.
- M Wetter and P Haves. A modular building controls virtual test bed for the integration of heterogeneous systems. *SimBuild 2008*, pages 69–76, 2008.
- P Xu, P Haves, and M Kim. Model-based automated functional testing - methodology and application to air-handling units. *ASHRAE Transactions*, OR-05-13-4, 2005.

A Appendix: Campus Load and Return Temperature Models

A.1 Overview

As discussed in Chapter 7, the campus chilled water system MPC model is divided into three major parts: the cooling plant, the storage tank, and the campus buildings. At each simulation timestep, given a supply water temperature (T_{wi}) and weather conditions, the campus part of the model must calculate the return temperature (T_{wo}) and the required chilled water flow rate (\dot{m}_w). This calculation is divided into two sub-models, as shown in Figure 98. The campus load is modeled as a function of weather and occupancy disturbances. It is not dependent on any other models for its input values, and thus may be treated as a disturbance model whose values can be pre-calculated at each controller time step and does not need to be repeated iteratively within the online optimization. The flow rate and return temperature from the campus is a function of this load, the ambient temperature, and the chilled water supply temperature, the latter of which is a control variable, so the return temperature model must be included in the fast-running online system model.

Figure 98: Campus load and return temperature models

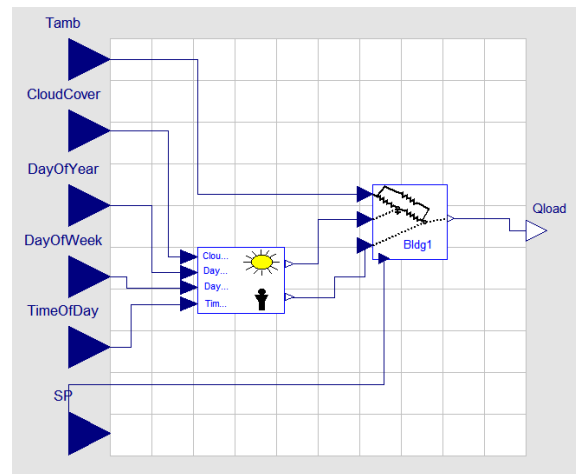


A.2 Campus load model

A.2.1 Model description

The objective of the campus load model is to predict the campus cooling load based on the ambient conditions, time of day, day of the week, and time of year. The building load model has two main sub-components: a Gains component and a Building Thermal Load component.

Figure 99: Building load model in Modelica, showing the two main sub-components



Gains component

The Gains component takes the time of day, day of week, day of year and cloud cover as inputs and outputs a solar load and an internal load. The internal gain is essentially just a scheduled heat addition to the space, representing the heat gains from people, lights and equipment, with different values for nighttime, daytime during the week and daytime on Saturday and daytime on Sunday. The solar load is determined by first calculating the extraterrestrial horizontal radiation (an atmospheric-independent value which is a function of the time of day and day of year), and then using the cloud cover and two parameters for the building's geometry and absorptivity to calculate the inside and outside solar gain values. The model has 12 parameters whose values are determined by calibration (Table 18), and one parameter (site latitude) that is specified by the user. The equations for the inside and outside solar gain calculations are shown in Equations 26 through 30.

Table 18: Gains component parameters

Internal gain	
occDayStart	hour of the day when the daytime internal load starts
occDayEnd	hour of the day when the daytime internal load ends
QintBaseline	a constant load independent of occupancy, ie. the overnight load
QintOccupied	this plus QintBaseline = daytime internal load for a weekday
QintOccupiedSaturday	this plus QintBaseline = daytime internal load for a Sat. daytime
QintOccupiedSunday	this plus QintBaseline = daytime internal load for Sun. daytime
Solar gain	
bldgSolarOutsideQperHorRad	a building-geometry parameter: solar gain on the outer wall mass per unit of horizontal radiation
bldgSolarInsideQperHorRad	a building-geometry parameter: solar gain on the inner wall mass per unit of horizontal radiation
cloudCover.fewClouds	when cloudCover = fewClouds, the ratio of extraterrestrial horiz. solar to horiz. solar at building
cloudCover.scatteredClouds	when cloudCover = scatteredClouds, the ratio of extraterrestrial horiz. solar to horiz. solar at building
cloudCover.brokenClouds	when cloudCover = brokenClouds, the ratio of extraterrestrial horiz. solar to horiz. solar at building
cloudCover.overcast	when cloudCover = overcast, the ratio of extraterrestrial horiz. solar to horiz. solar at building

$$\text{solarHour} = (\text{TimeOfDay} - 12) \cdot 15^\circ \quad (26)$$

$$\text{solarDec} = -23.45^\circ \cdot \cos\left(360^\circ \cdot \frac{(\text{DayOfYear} + 10)}{365}\right) \quad (27)$$

$$\begin{aligned} \text{extraterrHorizSolar} = \max(0, \cos(\text{solarHour}) \cdot \cos(\text{solarDec}) \cdot \cos(\text{latitude}) \\ + \sin(\text{solarDec}) \cdot \sin(\text{latitude})) \end{aligned} \quad (28)$$

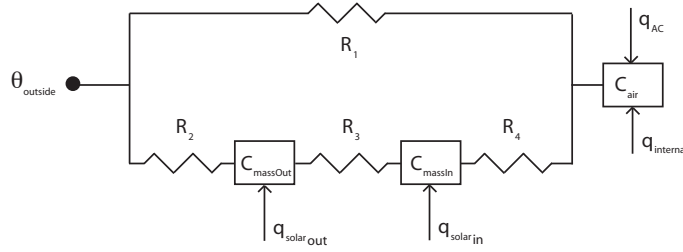
$$\begin{aligned} \text{QsolarOutside} = \text{cloudCover}.\text{[currentCondition]} \cdot \text{extraterrHorizSolar} \\ \cdot \text{bldgSolarOutsideQperHorRad} \end{aligned} \quad (29)$$

$$\begin{aligned} \text{QsolarInside} = \text{cloudCover}.\text{[currentCondition]} \cdot \text{extraterrHorizSolar} \\ \cdot \text{bldgSolarInsideQperHorRad} \end{aligned} \quad (30)$$

Building Thermal Load component

The BuildingThermalLoad component is based on the thermal resistance and capacitance model shown in Figure 100 and Equations 31 through 33. Various lumped parameter models like this have been developed and used in the literature (see Lee and Braun (2004) for a good example and further references). Its inputs are the ambient temperature ($\theta_{outside}$), the outside solar load ($q_{solarOut}$), the inside solar load ($q_{solarIn}$), the internal load ($q_{internal}$) and the indoor temperature set-point. It outputs the cooling load (q_{load}), and stores the temperature state of the thermal mass (T_{mass}). The component has six parameters: UAeff_inst ($= \frac{1}{R_1}$), UAc_outside ($= \frac{1}{R_2}$), UAc_inside ($= \frac{1}{R_4}$), UAc_middle ($= \frac{1}{R_3}$), WallCapInside ($= C_{massIn}$), and WallCapOutside ($= C_{massOut}$).

Figure 100: Thermal resistance and capacitance model



$$q_{load} = \max \left(0, q_{internal} + \frac{1}{R_4}(T_{massIn} - T_{zone}) + \frac{1}{R_1}(T_{amb} - T_{zone}) \right) \quad (31)$$

$$\dot{T}_{massIn} = \frac{q_{solarIn} + \frac{1}{R_3}(T_{massOut} - T_{massIn}) + \frac{1}{R_4}(T_{zone} - T_{massIn})}{C_{massIn}} \quad (32)$$

$$\dot{T}_{massOut} = \frac{q_{solarOut} + \frac{1}{R_2}(T_{amb} - T_{massOut}) + \frac{1}{R_3}(T_{massIn} - T_{massOut})}{C_{massOut}} \quad (33)$$

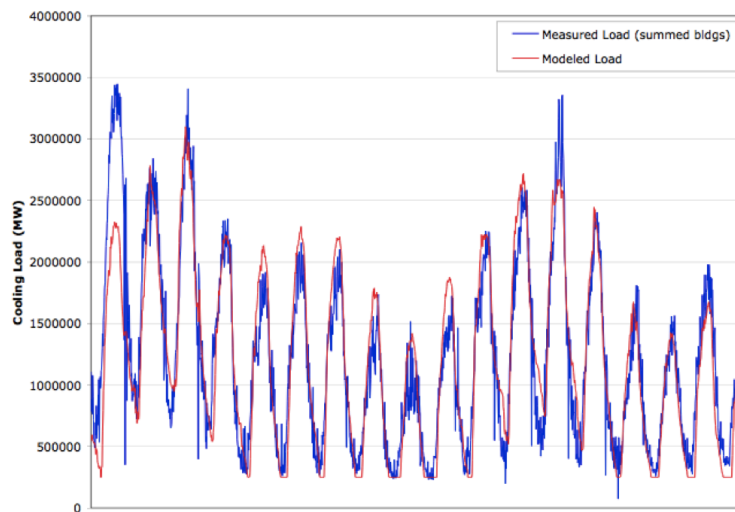
A.2.2 Model calibration

The model was calibrated with annual data during pre-experiment development, then re-calibrated right before each experiment with the two previous weeks to ensure that it was up to date with any changes in occupancy or other attributes. Table 19 shows the calibrated parameter values and performance for the first experimental period.

Table 19: Campus load model calibrated parameter values

Name	Value	Name	Value
cloudCover.fewClouds	0.175	QintOccupiedSaturday	250000
cloudCover.scatteredClouds	0.4	QintOccupiedSunday	125000
cloudCover.brokenClouds	0.6	QintBaseline	527000
cloudCover.overcast	0.8	UA_{inst}	66200
bldgSolarOutsideQperExtratHoriz	400000	UAcOut	100000
bldgSolarInsideQperExtratHoriz	400000	UAcIn	100000
occDayStart	7	UAcMiddle	99600
occDayEnd	20	WallCapInside	5000012800
QintOccupied	500000	WallCapOutside	500000000

Figure 101: Campus cooling load, May 16 - Jun 2, 2009



A.3 Campus return temperature model

A.3.1 Model description

The main purpose of the heat exchange model is to predict the chilled water return temperature and flow rate, given the chilled water supply temperature, cooling load and ambient air temperature. A block diagram of the model is shown in Figure 102. The return temperature and flow rates are important in that they determine the effective capacity of the storage tank in the system, as well as influencing the COP of the chillers and the energy consumption of the system pumps. The model plays a key role within the MPC in deciding if and when the chilled water supply temperature should be increased or decreased.

Figure 102: Block diagram of the heat exchange model

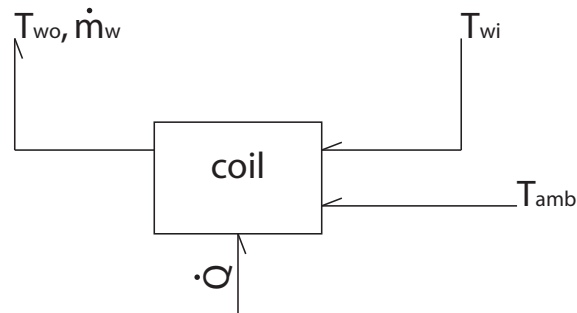
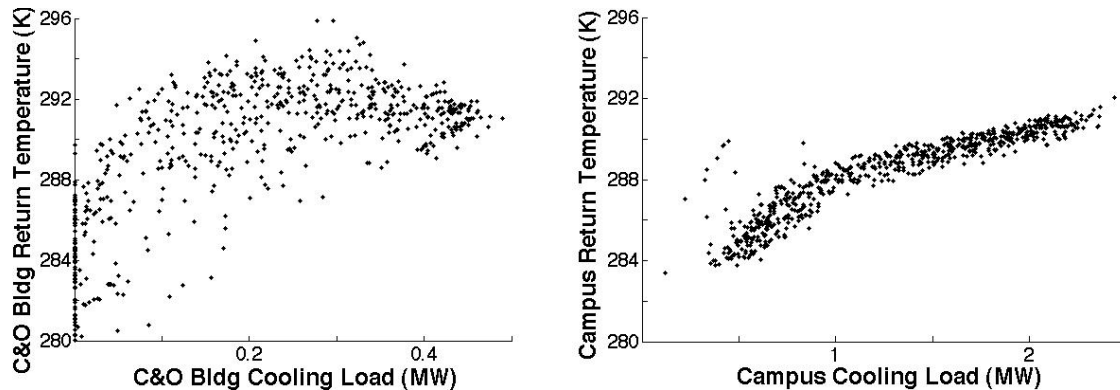


Figure 103 shows hourly measurements of the C&O Building and the campus return water temperatures for the period Aug29-Sep4 2009, during which time the supply water temperature was kept at their standard operating point of 277K. A fully empirical model of return temperature as a function of load and ambient temperature can capture the behavior in Figure 103 well (the curve is a function of the load and the scatter a function of the ambient temperature), but in order for the model to be useful in predicting behavior on non-standard conditions, particularly for different supply water temperatures, some first principles should be incorporated. The challenge is in devising a semi-empirical model to accurately capture the measured data and also to predict the effects of changing the supply water temperature.

Figure 103: Measured return temperatures for (a) C&O Building, (b) entire campus



The model was developed through detailed consideration of the C&O Building, which has a single air handling unit, and its behavior was extrapolated to the campus level (the execution speed requirement for the MPC application prohibited modeling all of the heat exchangers separately, and initial tests indicated that this aggregation would not be a significant source of error). The model is based upon an air-to-water counterflow heat exchanger. The system of Equations 34 and 35 is used, with unknowns m_w and T_{wo} , where the Log Mean Temperature Difference (LMTD) ΔT_m is defined in Equation 36.¹ This system of equations requires values for the air side entering and leaving temperatures. The leaving air temperature T_{ao} is kept as a constant in the model, and the entering air temperature is calculated as in Equation 37, with a constant return temperature $T_{airReturn}$, the measured outdoor temperature T_{amb} , and a calibration parameter γ as the percentage outside air.²

¹Note that these two unknowns are not commonly used with these equations, and a variety of implementation challenges result. Equation 36 is undefined when $T_{ao} = T_{wi} = T_{ai} = T_{wo}$, when $T_{ai} = T_{wo}$, or when $T_{ao} = T_{wi}$. The system of equations is also challenging in that under particular conditions there are no solutions. In the Matlab implementation of the model, two `fminsearch` solution loops are used, one to solve Equation 35 for given a T_{wo} value (made more difficult when Equation 38 below is added to Equation 35), and the other used to find the intersection of Equations 34 and 35. (An explicit formulation of Equation 35 is possible using Matlab's `labrtw` function, but it seems to continually run into solutions that are undefined. Using an implicit solution loop avoids this.) In both cases, the tolerance of the solution loop can be set fairly high (0.1K for the T_{wo} value in our case), and a maximum number of iterations specified (15 in our case), which increases the model speed while still producing acceptable performance.

²Note that if the model is being used with a single air handler and the real-time data is available, T_{ao} and T_{ai} may be used directly as measured.

$$\dot{Q}_{load} = \dot{m}_w c_{pw} (T_{wo} - T_{wi}) \quad (34)$$

$$\dot{Q}_{load} = UA \Delta T_m \quad (35)$$

$$\Delta T_m = \left(\frac{T_{ao} - T_{ai} - T_{wi} + T_{wo}}{\log \left(\frac{T_{ao} - T_{wi}}{T_{ai} - T_{wo}} \right)} \right) \quad (36)$$

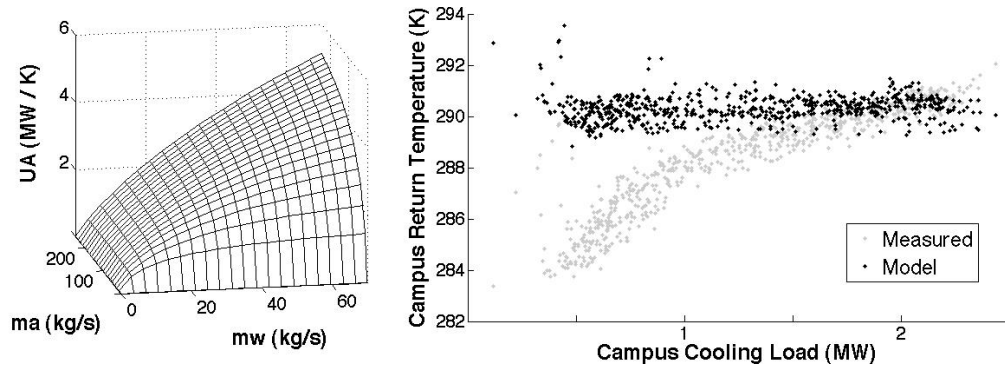
$$T_{ai} = T_{airReturn} + \gamma(T_{amb} - T_{airReturn}) \quad (37)$$

The assumption of a constant UA is not appropriate for variable flow heat exchangers. Such an assumption produces model behavior where the return temperature decreases with an increase in load, opposite to what is observed in Figure 103. So a variable UA is used. Based on the empirical work of Holmes (1982) and as used in the model of Wetter (1999), Equation 38 (shown in three dimensions in Figure 104a) is used for the relationship between the steady state UA and the water and air flow rates (Equation 39), with α_1 and α_2 used as calibration parameters. With this addition, the model performs much better than with a constant UA (Figure 104b), but it still does not accurately capture the observed curve of UA versus load.

$$UA_{steadyState} = (\alpha_1 \dot{m}_w^{-0.85} + \alpha_2 \dot{m}_a^{-0.85})^{-1} \quad (38)$$

$$\dot{m}_a = \dot{Q}_{load} / (c_{pa} (T_{ai} - T_{ao})) \quad (39)$$

Figure 104: (a) UA as a function of m_w and m_a (Equation 38), (b) Performance of model with the variable UA , Equations 34-39



There may be a variety of reasons why the UA is lower at low loads than expected, one of which is an observed increase in flow rate fluctuations at low loads (likely because the local PI gains were set to match design conditions). An adjustment factor ζ is added to the UA calculation to account for it, as shown in Equations 40 and 41. The form of Equation 41 was chosen such that ζ is in the range (0,1) and β is used as a calibration parameter to determine the strength of the effect. The variable ξ is a building-specific empirical function of the load. For the UC Merced case, Equation 42 was used, which is a function in the range (0,1) fit to the curve of flow rate fluctuations (differences in successive 15-min samples) versus load for the C&O Building. But the main purpose of this factor, as discussed further below, is to provide a means of modifying the thermodynamics-based model to better fit the measured data without losing the ability to use the model to reasonably predict the impacts of changes in supply water temperature.

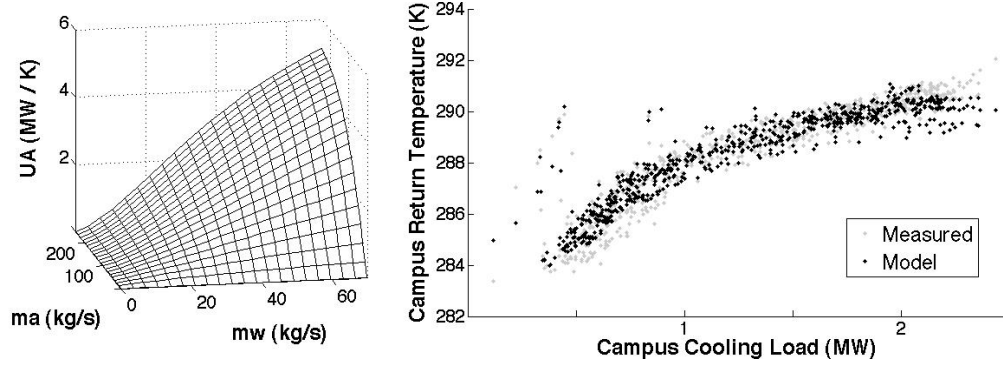
$$UA = \zeta (UA_{steadyState}) \quad (40)$$

$$\zeta = (1 + \beta\xi^2)^{-1} \quad (41)$$

$$\text{for C\&O Building, } \xi = \frac{-0.9074 \ln(1 + \dot{Q}_{load}) + 12.329 - 0.3}{12.329 - 0.3} \quad (42)$$

The UA is thus a function of m_w , m_a and Q_{load} , as shown in Figure 105a (shown in three dimensions by assuming Q_{load} to be a linear function of m_w and m_a), with the calibration parameters α_1 , α_2 and β . The parameters α_1 and α_2 determine the height and skew of the function, and β determines the magnitude of the dip (the difference between Figure 104a and 105a) at low loads. With this addition, the model can match measured data much better, as shown in Figure 105b.

Figure 105: (a) UA varied by Equations 38-42, (b) Model performance, Equations 34-42



A.3.2 Model calibration

The four model parameters, α_1 , α_2 , β and γ , should be calibrated to fit measured data through a global error-minimization method³. The following steps may be used to provide a good initial point for this global calibration:

1. Determine $T_{airReturn}$ and T_{ao} as averages of measured data, or based on set point info. Fit γ using the determined value of $T_{airReturn}$ and the data for T_{amb} and the mixed air temp T_{MA} .
2. Fit the parameters α_1 and α_2 using Equation 38 and only points where the load is greater than some heuristically-determined value (e.g. 0.3MW for C&O Bldg)
3. Fit the parameter β using the α_1 and α_2 values from step 2 and all of the data

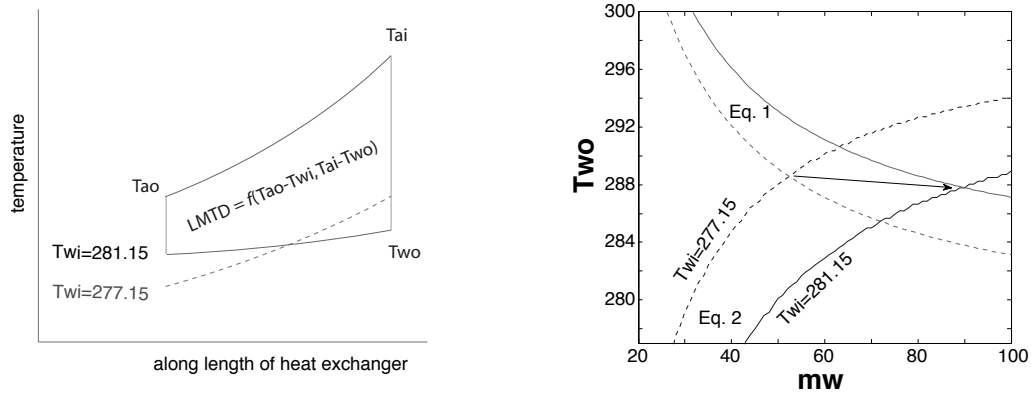
For the UC Merced study, the constant values used were $T_{airReturn} = 296.66$ K and $T_{ao} = 284.84$ K, and the calibrated parameter values were as follows: $\gamma = 0.1355$, $\alpha_1 = 7.3443e-05$, $\alpha_2 = 1.5464e-04$ and $\beta = 56.025$.

³Non-convergence of the system of equations can happen in cases where airflow rate is high relative to the load (high values of ma limit the potential impact of the water flow rate on the UA), and/or if the values of α_1 , α_2 or β are too high, so calibration constraints are necessary.

A.3.3 Model predictions with warmer supply water temperatures

Equation 34 makes one suspect that the return temperature would always increase with an increase in the supply temperature, and for a given water flow rate and load this is what happens with this equation. But for that same given flow rate and load, Equation 35 calculates a lower return temperature, for the following reasons: for given mass and air flow rates, the UA is a given, thus the LMTD is also a given; to maintain a given LMTD when the inlet water temperature is increased, the outlet water temperature must be decreased, as illustrated in Figure 106a. Equations 34 and 35 must always converge at a higher mw when T_{wi} is increased (Figure 106b), and depending on the load and the model parameters, the value of T_{wo} may be either increased or decreased.

Figure 106: (a) Maintaining a LMTD, (b) a new convergence point with a higher T_{wi}

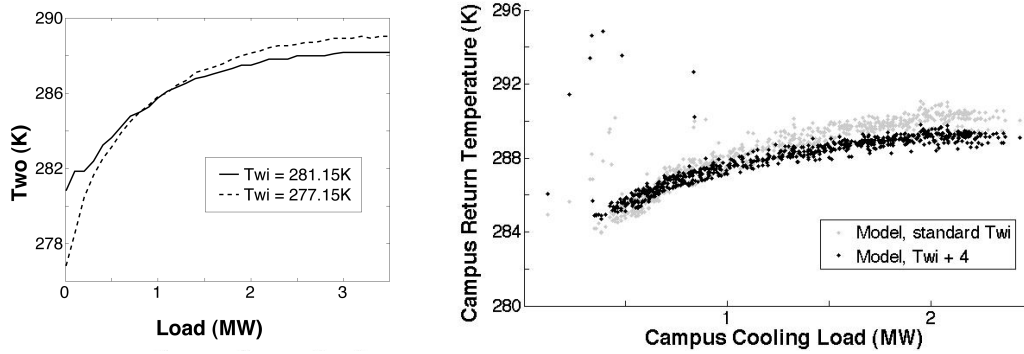


If supply water temperature is increased, the model predicts the somewhat non-intuitive behavior: the return temperature should change very little at most loads, except at very low loads where it should increase slightly, and at high loads it should decrease slightly.⁴ Figure 107a shows the abstracted result with an exaggerated 4K increase in supply temperature, without the scattering effect of the ambient air temperature. And Figure 107b compares the model predictions under normal conditions (as shown in Figure 105b) with the predictions when the supply water

⁴Note that this result is sensitive to the values of the calibrated α_1 and α_2 values, and at some combinations of these values (generally at the extremes) the effect of increasing T_{wi} is always to increase (or decrease) the solution value for T_{wo} .

temperature is increased by 4K. Figure 107 shows a 4K increase to highlight the nature of the effect; the pattern is the same with a 1K increase, but impacts are smaller.

Figure 107: Predicted return temperature values with a higher supply temperature

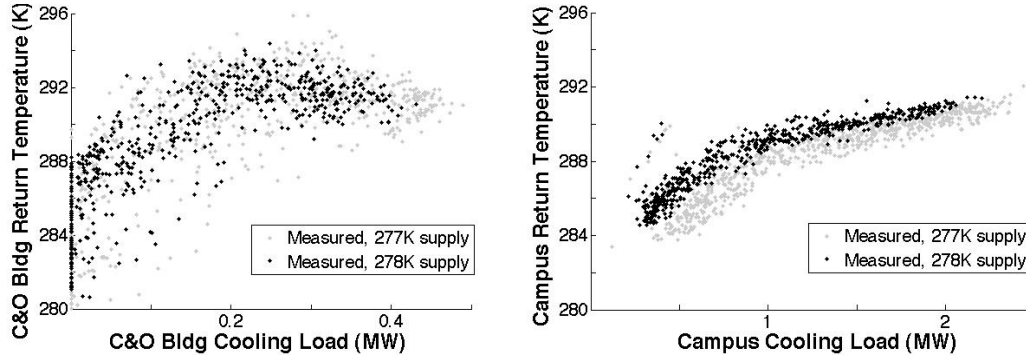


A.3.4 Experimental results

To test the validity of the model, an experiment was conducted where the campus chilled water temperature was increased by approximately 1K for the period Sep6-11, 2009. The model predicted that the return temperature would change very little, increasing by less than 1K at low loads and decreasing by less than 0.25K at high loads. This was correct for the C&O Building (Figure 108a), and for most other buildings on campus. But it was incorrect for the campus as a whole, where there was an approximately 1K increase in return temperature at low loads and an approximately 0.5K increase at low loads (Figure 108b), and the increase of return temperature at the lowest loads was actually more than the increase in supply temperature, which runs counter to the models thermodynamic principles.

The unexpected campus-level behavior was mostly caused by two buildings, including the S&E Building, which accounts for approximately one quarter of the campus load and which showed a pronounced offset in the return temperature versus load graph. While the C&O Building has just one air handler and the other buildings that performed as expected have a few each, but the S&E Building has many, and also has other complicating factors such as on-site outdoor air pre-cooling. The results thus suggest a partial affirmation: the model is perhaps appropriate as is for individual buildings or for individual air handling units, but further modifications

Figure 108: Measured effect of increased T_{wi} on (a) C&O Building, (b) entire campus



may be necessary for it to be used with whole campuses and buildings that deviate significantly from the one air handler configuration upon which the model was based. Further research is required to further validate, modify and determine the range of applicability of the model.

A.3.5 Discussion

One of the downfalls of using this model at a campus level is that the air side becomes a potentially problematic abstraction. It may work in cases where all of the buildings are similar enough to assume similar air side behavior. But in other cases, if the modeling application can handle the additional simulation time required, it may be wise to model each building (or even each air handler) separately using this model. The outside air fraction (Equation 37) can also be better suited to buildings with economizers by using two different γ values based on T_{amb} (the resulting discontinuity does not present any numerical problems since this is a pre-calculation at each time step).

Equations 40-42 were added to the model to allow a better fit to the empirical data (with lower UA than expected at low loads) while maintaining the model's counterflow heat exchanger basis. They could be improved upon in one of two ways. Either (1), digging into more detail: the identification of flow rate fluctuations as a potential cause of a lower UA at low loads is perhaps a useful starting point for this investigation, but the current use of Equation 41 and building-specific Equation 42 is just a convenient stretch of principle to fit the data; further investigation of more potential causes and more explicit relationships between these causes and the model

equations is called for. Or (2), considering a purely empirical fit of UA modification as a function of load: graphing and fitting a curve to Equation 43 would likely result in equations similar to Equations 40-42, but a slightly different form might provide a better fit, and the approach would be more transparent.

$$\zeta(Q_{load}) = UA_{model}/UA_{measured} \quad (43)$$

As often happens with the modeling of building systems in operation, a number of side-benefits resulted from looking at the system and its energy data in detail. Uncovering and fixing various small problems during MPC model development and experimentation produced energy savings for UC Merced that were similar in size to the energy savings resulting from the application of the improved controller. In particular, with the heat exchange modeling, two such problems were identified: some of the building bridges were recirculating their chilled water return when they should not have been; and comparing the return temperatures and flow rates from the various buildings with the return temperature at the central plant pointed to a bypass problem or a faulty flow rate meter somewhere in the system, the latter of which was found and fixed, decreasing pump energy use, improving comfort in one of the buildings and increasing the chiller COP by increasing the campus return water temperature.