Mixed-integer Model Predictive Control with Applications to Building Energy Systems

By
Michael J. Risbeck

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

Doctor of Philosophy
(Chemical and Biological Engineering)

at the
University of Wisconsin–Madison
2018

Date of final oral examination: July 23rd, 2018

The dissertation is approved by the following members of the Final Oral Committee:
James B. Rawlings, Professor, Chemical and Biological Engineering
Christos T. Maravelias, Professor, Chemical and Biological Engineering
Victor Zavala, Associate Professor, Chemical and Biological Engineering
Ross E. Swaney, Associate Professor, Chemical and Biological Engineering
James R. Luedtke, Associate Professor, Industrial and Systems Engineering

To my family.

Acknowledgments

First and foremost, I would like to acknowledge my Ph.D. advisers Profs. James B. Rawlings and Christos T. Maravelias. Without their knowledge and guidance, I would not have been able to accomplish this work. I would additionally like to thank Profs. Victor Zavala, Ross E. Swaney, and James R. Luedtke for serving on my defense committee.

For my research, I've been very lucky to work on a project for Johnson Controls, Inc. The main application of building energy optimization is a rich problem with many challenges and opportunities for which standard control solutions are not sufficient. A big thanks to Bob Turney, who was our liaison and fearless leader, and who was always a good source of feedback. Thanks also to Mike, Matt A., Mohammad, and Matt E. for their helpful discussions throughout the project. I was able to spend a summer at JCI on an internship, and it was a perfect mix of working on my research problem and analyzing some of their deployment strategies ("pioneering" and "establishment" in the language of Bob's famous four bubbles of technology development). I'd additionally like to thank Kirk Drees for his helpful insights and especially for hiring me to work full-time at JCI after I finish my Ph.D. As a grad student, it was great to work on an application that was making use of cutting-edge academic theory, and I'm excited to continue working on it as a practitioner.

Throughout my time at UW, I've had the pleasure to share an office with many of my Rawlings Group colleagues. I thank Megan, Luo, and Ankur for helping bring me up to speed on the group's Linux machines. I also thank Nishith, Min, Travis, and Doug for being constant sources of information and for what I would characterize as the appropriate amount of time-wasting discussions about which lunch catering place has the best pickles (Panera, due to both quality and quantity) and about whether there exists a dessert item for which the addition of nuts does not immediately make it worse (there does not). I wish Doug, Travis, Koty, and Pratyush the best of luck as they follow Professor Rawlings to Santa Barbara. I'd also like to thank my Maravelias Group colleagues for their help with GAMS and other optimization issues.

Of course, I would not have made it through five years of grad school without the great friends I've made. I thank Frank, Parth, Kevin, Loukas, Travis, Paul, and Joe (when he's feeling sociable) for all the beers, board games, brunches, magical movie rewards, and sporting events we've shared at UW. It's crazy to think how far we've come from being

trapped in the crack as first-years. I'd also like to give a shout-out to everybody who ever took part in our Friday Happy Hours; there are too many to name, so I'll take the easy way out and avoid forgetting anybody. Finally, a big thanks to everybody in ChEGS for weathering the storm and hosting social events (hopefully wine and cheese night will make its triumphant return at some point). Of course, we didn't come here to have fun, but it certainly helped.

On the software side, I would like to acknowledge Python and GNU Octave for both being free-source programming languages useful for numerical computing. GNU Octave is generally compatible with Matlab and enables all users (not just those with $1,000 to spare on a license) to perform control simulations and calculations. I used Octave primarily for the building energy optimization in Chapters 5 and 6, as well as for some of the examples in Chapters 2 and 3. Python provides useful data types and a flexible data model that leads to concise and maintainable code. The NumPy package provides a multidimensional array type that addresses many of my frustrations with Matlab's "everything is a matrix" attitude, and the SciPy package provides useful statistical and linear algebra functions. I used Python for the scheduling examples in Chapter 4, as well as for plots and numerous utility scripts used to build this thesis. I'd also like to thank Joel Andersson, the main developer of CasADi, which provides a powerful set of symbolic tools and links to solvers that are useful for optimal control. CasADi is the backbone of our own MPCTools package, which has been used in various MPC short courses (and in this thesis) to facilitate setting up and solving MPC problems in Octave and Python.

For typesetting this thesis, I have used LaTeX. Despite the fact that it was a bit of a struggle to get double spacing active in the right places (and only in the right places), I shudder to think how much frustration there would have been had I used more traditional word-processing software. For diagrams, I have used the LaTeX package TikZ, which provides a very useful set of macros for describing diagrams. For plots, I am very thankful for the Python package Matplotlib, which provides an object-oriented plotting interface. Without Matplotlib, the plots in this document would be nowhere near as beautiful as they are. All of the source code to run calculations and typeset this document are hosted in a Mercurial repository on BitBucket and can be made available upon request.

Last and certainly not least, I thank my family for their constant love and support. Without their nurturing of my curiosity and desire to learn, I doubt I would have pursued a Ph.D., and I am ever grateful for their help and advice.

Michael Risbeck

Madison, WI

August 2018

Contents

List of Figures

List of Tables

Abstract

With utility markets moving toward more complicated and time-varying rate structures, it is becoming increasingly difficult to operate energy-intensive processes at low cost. As the capacity of intermittent renewable generation grows, price volatility is likely to increase, and so current heuristic strategies can lead to expensive and inefficient operation. Fortunately, with improvements in computational power and greater availability of raw data, there exists a significant potential to use optimization techniques *in real time* to reduce costs and improve performance. Model predictive control (MPC) is an advanced process control technique whereby process inputs are chosen by solving, in real time, an optimization problem with an embedded process model. While MPC has traditionally been limited to steady-state tracking problems, recent advances have enabled its application to a wider class of systems.

To this end, we present a formulation of mixed-integer MPC (MIMPC) that allows the inclusion of *discrete-valued* decision variables in addition to standard continuous variables, and we show that MIMPC possesses the same stability properties as standard MPC. In addition, we formulate *suboptimal* MIMPC, which relaxes the requirement that true optimal solutions be found at each timestep, which significantly improves computational tractability. We then present extensions of tracking MIMPC to demonstrate inherent robustness and consider economic (rather than tracking) objective functions. Using these techniques, we formulate standard production scheduling problems as an instance of economic MIMPC, and we show how the inclusion of terminal constraints in the open-loop problem leads to improved *closed-loop* performance.

As a specific application, we discuss real-time cost optimization for large-scale energy systems systems in commercial buildings. Such systems are significant consumers of electricity and are subject to both time-varying prices and peak demand charges assessed based on a customer's maximum instantaneous use of electricity over a month. Using MIMPC and approximate equipment models, the operation of central energy plants (including discrete on/off decisions) can be optimized online. By using various forms of energy storage, time-varying price and efficiency differences can be exploited by the optimizer, reducing utility costs and electricity usage. Through examples, we demonstrate that these techniques can be applied to large-scale building energy systems to lower costs and reduce energy usage.

How beauteous mankind is!
O brave new world,
That has such [utility pricing structures] in it!

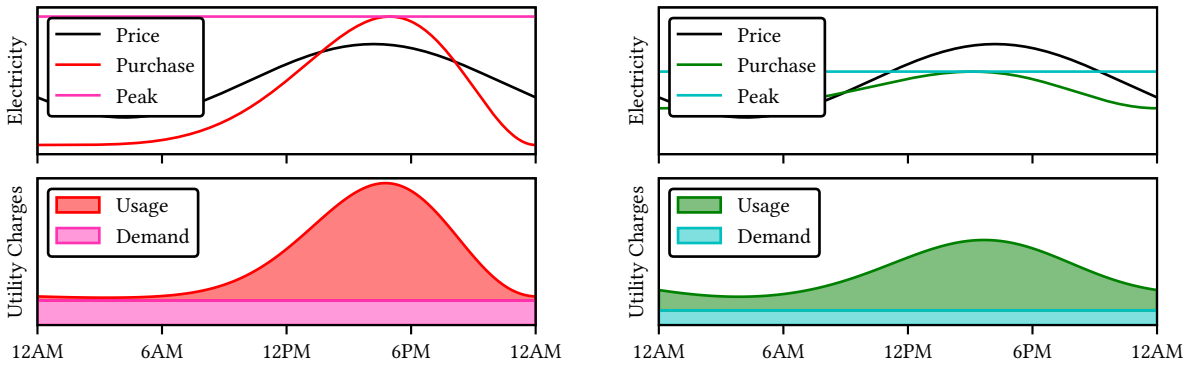
— Miranda
The Tempest

Chapter 1

Introduction

With increased attention toward the negative impact of carbon and other greenhouse gas emissions on global climate, it is apparent that broad changes need to be made to primary energy consumption habits. Within the United States, energy use in buildings accounted for 41% of energy consumption and 40% of carbon emission in 2010. Of these figures, residential buildings constituted 54% of energy use, with commercial buildings consuming the remaining 46% (US Department of Energy, 2011). With such a large contribution to national energy consumption, even modest efficiency improvement would have a large effect in absolute terms.

While across-the-board reduction would be ideal, existing architecture often precludes any sweeping cuts without sacrificing performance. Furthermore, due to cyclic occupation, resource consumption in buildings is highly time-varying (Touretzky and Baldea, 2014), and specifically for electricity, the onus of meeting transient demand profiles is placed directly on suppliers. Since the efficiency of electricity generation is highly dependent on load, aggregate energy efficiency can decrease significantly in times of high demand (Gyamfi et al., 2013). To combat this effect, electricity suppliers are transitioning from flat rate structures to more complicated real-time market utility prices and peak demand charges (Albadi and El-Saadany, 2007). As a result of these demand response strategies, consumers are encouraged to purchase additional electricity when it is abundant and discouraged from purchasing when scarce. The intended effect is that flexible customers reduce their utility bill, while network-wide primary efficiency is increased. An example demand response effort is shown in Figure 1.1. In Figure 1.1a, the nominal use profile leads to high utility costs, and the consumer's peak corresponds to the grid-wide peak. By contrast, Figure 1.1b shows shifted consumption to flatten demand and reduce peak purchase. Although the optimized profile consumes slightly more electricity overall, grid-wide primary energy efficiency increases due to lower demand variance.



(A) Nominal consumption scenario. The high purchase peak coincides with the highest electricity price, leading to significant time-of-use and peak demand charges.

(B) Optimized consumption scenario. The peak has been reduced and shifted to a time of lower electricity price, leading to lower electricity prices and demand variance.

FIGURE 1.1: Example demand response scenario. By shifting usage, significantly lower electricity costs can be achieved.

Unfortunately, it is difficult for consumers to determine exactly *how* to modify their consumption in accordance with these economic signals. Complicated and possibly uncertain pricing structures leave building operators largely adrift with only minimal feedback in the form of a monthly utility bill. Thus, without a means to reliably and optimally make electricity usage decisions, the ultimate result is higher cost to end-users and lower network-wide efficiency.

As a means of realizing these cost savings and efficiency improvements, we propose the application of model predictive control (MPC). Under MPC, an approximate system model is combined with forecasts of relevant external parameters to create an optimization problem that is solve online and in real time to make process decisions (Rawlings et al., 2017b). Traditionally, application of MPC has been restricted to systems in which the end goal is setpoint tracking and for which the decision space is continuous (often, convex). However, recent results (Amrit et al., 2011; Angeli et al., 2012) have extended applicability to systems with arbitrary objective functions rather than just measures of distance to setpoint. This capability allows MPC to optimize tangible measures of system performance, e.g., the complicated electricity price structures described in the previous paragraph. We wish to further extend MPC theory to cover systems with discrete-valued actuators, which allows higher-level decisions, such as switching a piece of equipment on or off, to be made optimally rather than via heuristic methods. As mixed-integer optimization techniques are steadily improving (Belotti et al., 2013), the scope of tractable online optimization problems is increasing as well. Especially with the growth of intermittent renewables in the electricity grid, it is important to have dispatchable loads that can respond to pricing changes to avoid stranded power. Thus, a strategy like MPC that enables buildings to react and adjust utility demand in real time can help address major challenges facing the ever-changing electricity grid, all while reducing costs for end consumers.

1.1 Outline

To advance our goal of tractable real-time optimization strategies for energy systems in commercial buildings, we highlight five major contributions of this work as follows:

- Extension of MPC stability theory to time-varying systems with discrete actuators (Chapter 2)
- Expansion of economic MPC theory to include time-varying systems, discrete actuators, and peak charges (Chapter 3)
- Formulation closed-loop scheduling as an instance of economic MPC with provable nominal closed-loop properties (Chapter 4)
- Creation of a tractable MILP model for real-time cost optimization of central energy plants (Chapter 5)
- Development of two decomposition strategies for large-scale airside/waterside optimization (Chapter 6)

Each of these topics is briefly introduced in the following sections.

1.1.1 Mixed-Integer Model Predictive Control

To incorporate the additional class of discrete-valued decisions, we extend standard MPC theory to include systems with both continuous- and discrete-valued actuators. We refer to this development as mixed-integer model predictive control (MIMPC). Despite the addition of discrete decisions, the basic structure of MPC remains unchanged. At *each* timestep, an optimization problem is solved to determine an optimal *trajectory* of states and inputs for the system to follow, but only the *first* input is actually used. After waiting for the system to evolve (possibly being affected by disturbances), the process is repeated. We illustrate the structure of an MIMPC controller in Figure 1.2. Using an approximate system model, the optimizer determines how to drive the system's output variables y to their setpoints by manipulating inputs u. As shown in this example, these inputs can include both continuous "how much" decisions, as well as discrete "yes or no" choices. A typical use case would be to not only choose at what level to operate a piece of equipment, but also to decide whether to keep it on or switch it off.

The addition of discrete-valued actuators raises a number of stability and robustness issues for closed-loop control. In particular, discrete actuators violate a (historically very common) assumption of local convexity and controllability. However, as we will demonstrate, careful attention to the assumptions regarding the geometry of the input set will reveal that many results already apply to systems with discrete actuators, and many more results can be extended by relaxing the restrictive assumptions. Since our intended application is building energy systems, which are inherently time-varying, we also extend standard results to the arbitrary time-varying case, which covers periodic and time-invariant systems as a special case. Finally, because of the challenges associated with finding

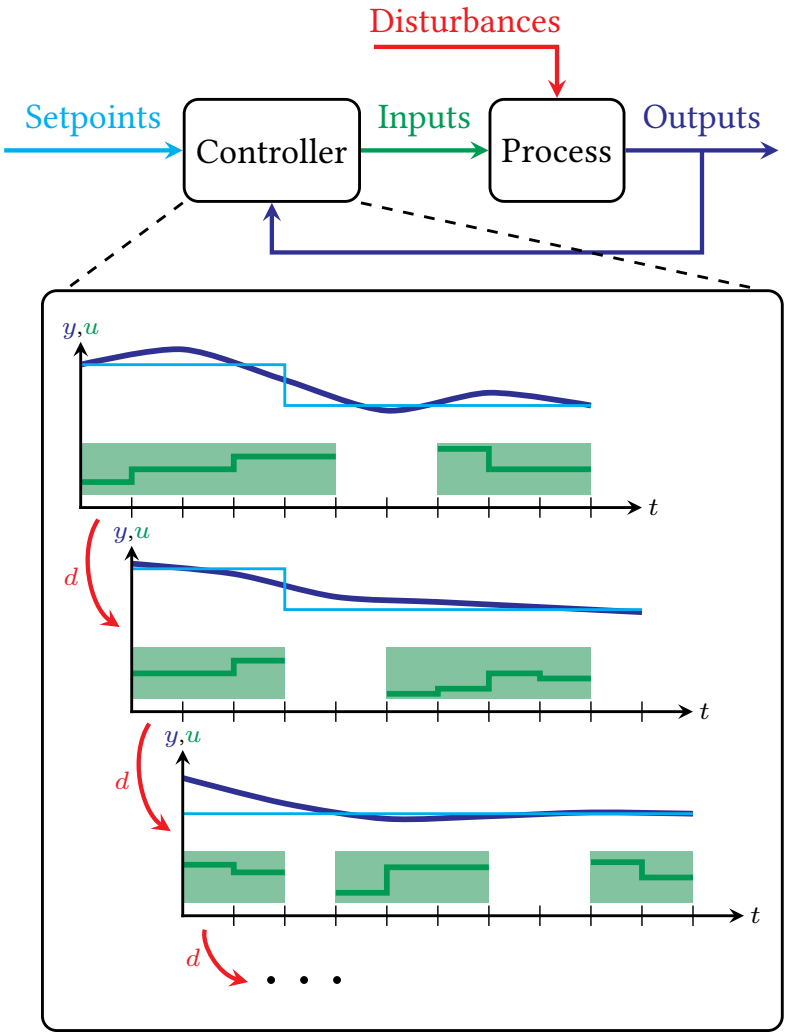
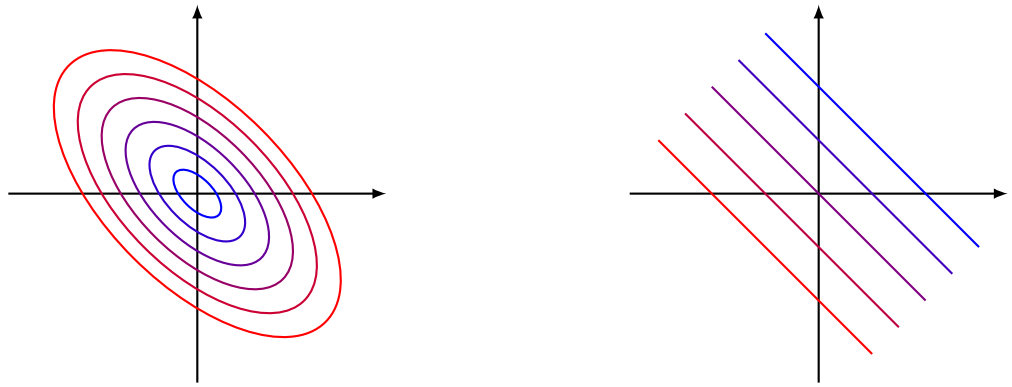


FIGURE 1.2: Diagram of an MIMPC controller. At each timestep, an optimal control problem is solved to determine the value of inputs u based on the measured outputs y. Note that the optimization problem includes both continuous and discrete decisions.



(A) Diagram of standard tracking MPC objective function. Deviations from the origin are penalized symmetrically.

(B) Diagram of economic MPC objective function. Asymmetric costs often reflect tangible performance methods.

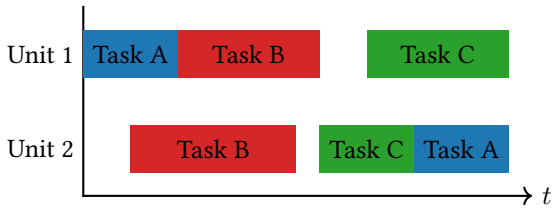
FIGURE 1.3: Comparison of tracking MPC vs. economic MPC objective functions.

globally optimal solutions to mixed-integer optimization problems, we also present a suboptimal formulation for MPC that, among other things, does not require globally-optimal solutions to be found at any time.

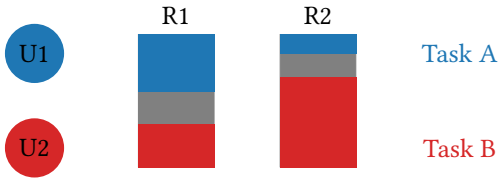
1.1.2 EXTENSIONS OF MIMPC

Although the results for tracking MIMPC provide a characterization of nominal performance, they raise a handful of practical implementation questions. Chief among them are what happens when the system model is not exact and how are the system's setpoints chosen. For the first question, we can consider any system behavior that differs from the model as the effect of *disturbances*. While bounded disturbances can be addressed at design time via robust MPC, the resulting control laws are often conservative. Instead, a useful property to have is *inherent* robustness, which means that the nominal control law cannot be destabilized by arbitrarily small disturbances. We demonstrate that this property holds under only slightly stronger assumptions that were required for stability in the previous chapter.

For the second question, operating setpoints are generally chosen by performing some sort of economic analysis at a higher layer. While this decomposition works for many systems and can be justified by timescale separation, it is not always clear that a system can be optimally operated at steady state. For example, in time-varying systems, there may not exist a feasible steady-state operating point. To address these cases, we extend results about *economic* MPC to the time-varying discrete-actuator case. In contrast to standard tracking MPC, the objective function can be chosen arbitrarily to represent tangible quantities of interest, rather than having to be designed specifically to stabilize a given setpoint. We illustrate these differences in Figure 1.3. With this modification, it is no longer necessary to know the optimal operating point a-priori, and economic performance is optimized dynamically and online, rather than statically and offline.



(A) Example optimal schedule. With discrete representation of time, decision variables are inherently discrete-valued.



(B) Typical constraints for scheduling problems. Units can only perform one task at a given time, and total shared resource constraints must be respected.

FIGURE 1.4: Elements of scheduling problems. Like MPC, scheduling optimizes decisions over a finite prediction horizon and can be implemented in closed-loop.

1.1.3 Closed-Loop Scheduling

Scheduling problems generally concern the allocation of resources to complete a given set of tasks. As illustrated in Figure 1.4, the key decisions of unit/task assignments are inherently discrete in nature, and in making those choices, constraints on shared resource consumption must be respected. Determining even a single optimal schedule can be computationally challenging, but recent advances in problem formulation and optimization methods are deceasing the required effort.

Unfortunately, determining a single schedule is generally not sufficient for practical use. Due to the finite-horizon nature of scheduling problems, schedule quality often deteriorates in later time points, for example exhausting inventory that would be useful for future production. In addition, due to unit breakdowns, modified orders, etc., a schedule can quickly become suboptimal simply because the underlying parameters have changed. Therefore, scheduling applications can benefit from the same closed-loop implementation used in MPC, in which schedules are reoptimized online at each timestep. By applying the economic MPC theory developed in the previous chapter, we can provide certain nominal closed-loop guarantees to avoid the possible pitfalls of naive closed-loop optimization. We also address the case of integrated scheduling and control, in which each unit and task is defined by an underlying dynamic model. This problem can also be cast as an instance of economic MPC, although the resulting optimization problem is challenging and may require suitable approximation or decomposition.

1.1.4 Central Energy Plant Optimization

As a specific application of of closed-loop scheduling, we consider the optimization of central energy plants. In campuses or large commercial buildings, building heating and cooling is performed most efficiently by producing hot and chilled water in a centralized facility using large, high-efficiency equipment. Because of their large size, these plants are often subject to the electricity pricing structures discussed at the beginning of this chapter. In addition, due the presence of multiple parallel units and variable unit efficiency, even determining the optimal configuration to

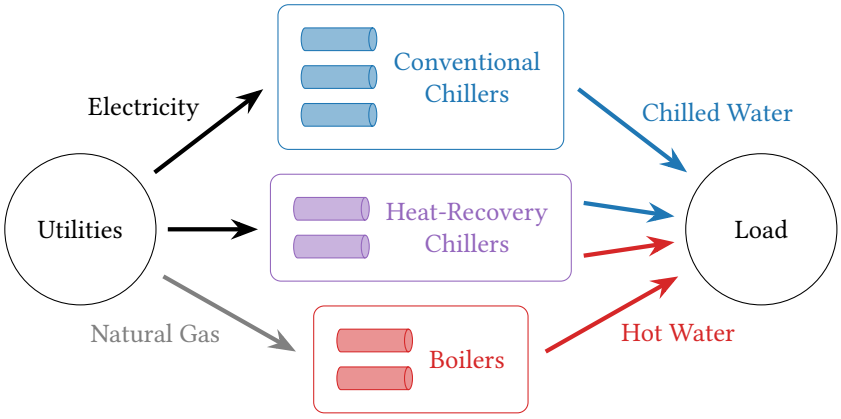


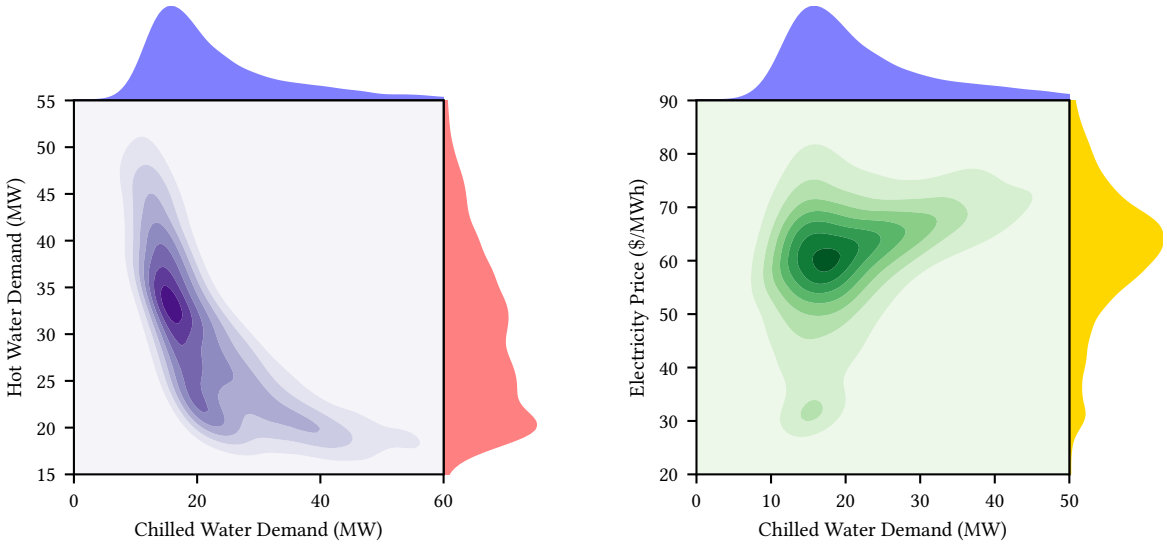
FIGURE 1.5: Representative diagram of central energy plant. Heat recovery chillers increase overall efficiency when there is coincident demand for chilled and hot water.

meet a given total load is nontrivial. Thus, we present a mixed-integer linear programming (MILP) formulation that is suitable for real-time optimization of central energy facilities.

Figure 1.5 shows a simple diagram of major central plant equipment. These units consume electricity or natural gas (purchased from utility markets) to produce hot and chilled water that is eventually delivered to the load. Distributions of key problem parameters are shown in Figure 1.6. We note from these plots that there is significant overlap between heating and cooling demand, which means the use of heat-recovery chillers (HRCs) can lead to significant efficiency improvements, as these units produce hot and chilled water simultaneously. In addition, electricity price and cooling demand are correlated, which means meeting demand just in time leads to significant electricity charges. Thus, large systems often employ water storage tanks so that resources can be produced ahead of time when electricity is cheap and then consumed later without needing to run equipment. This additional layer of flexibility can lead to significant improvements in both utility cost and energy efficiency, but it also significantly complicates optimal operation. Therefore, the application of the optimization strategies developed in this chapter can lead to significant improvement over manual or heuristic scheduling.

1.1.5 LARGE-SCALE HVAC OPTIMIZATION

Although central energy plants are a significant opportunity for cost optimization as just discussed, focusing only on the central plant leaves excludes additional source of energy storage. The most significant part of the central plant's heating or cooling demands are generally used by the building's heating, ventilation, and air-conditioning (HVAC) systems. For comfort purposes, the focus of temperature control is for the air temperature, but the temperatures of the solid building components fluctuate as well. Because buildings are bound by the laws of physics, these temperature changes do not happen instantaneously, and these dynamics can be exploited as a source of additional storage (and



(A) Joint and marginal distribution for chilled water demand and hot water demand.

(B) Joint and marginal distribution for chilled water demand and electricity price.

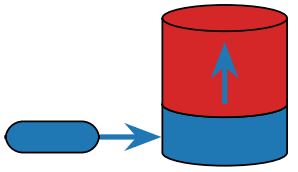
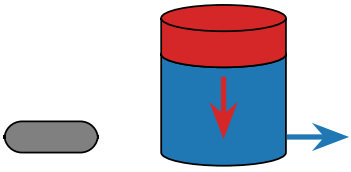
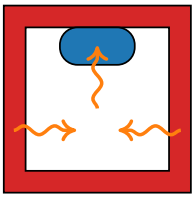
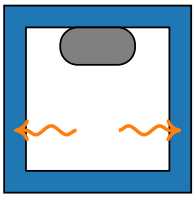
FIGURE 1.6: Distribution of key problem parameters. From one year of sample data provided by Johnson Controls.

thus cost savings). We refer to this effect as "passive TES," with the water storage tanks in central plants denoted as "active TES."

As shown in Table 1.1, both active and passive TES operate by the same principle. At night (or whenever electricity is cheap), equipment is run to produce extra cooling that is stored. Under active TES, the produced chilled water is simply held in a storage tank, while under passive TES, the storage occurs by leaching heat from the building solid components, thus lowering its tamperature. In the daytime (when electricity is expensive), this stored energy is then used to provide cooling without needing to run any equipment. For active TES, the water is simply withdrawn from the tank, while in passive TES, the lower temperature of the building mass causes it to naturally absorb heat from the air. Thus, by exploiting both active and passive TES simultaneously, energy storage capacity is significantly increased.

To this end, we extend the central plant optimization model from the previous section to optimize building temperature trajectories as well. With this change, a portion of the heating and cooling demand is now a decision variable, determined implicitly by the temperature trajectory of the buildings. Assuming building models capture the relevant dynamics of thermal mass temperature change, use of passive TES is implicit, and the optimizer naturally decides to pre-heat and pre-cool as necessary to reduce operating costs. Because there can be a significant independently varying (but interacting) building temperatures, we also present decomposition strategies so that realistically sized systems can be optimized online and in real time. These ideas thus further increase the scope of mixed-integer economic MPC and capture an additional source of cost or energy savings.

TABLE 1.1: Pictograms of TES strategies. Active TES stores energy in fluids, while passive TES stores energy in building mass.

	Nighttime	Daytime
Active TES		
Passive TES		

1.2 MATHEMATICAL NOTATION

Throughout this thesis, we make use of the following notation. The set of real numbers is denoted by $\mathbb{R}$ and the set of integers by $\mathbb{I}$. Subscripts on these sets denote domain restrictions (e.g., $\mathbb{R}_{\geq 0}$ for nonnegative reals or $\mathbb{I}_{[a,b]}$ for integers between a and b). To index discrete times, we use the $t \in \mathbb{T}$ with $\mathbb{T} \coloneqq \mathbb{I}_{\geq 0}$. The standard inner product between two vectors a and b is denoted $a^\mathsf{T} b$. The Euclidean norm of a vector x is denoted as $|x|$ with $|x|_Q \coloneqq \sqrt{x^\mathsf{T} Q x}$ for positive-semidefinite matrix Q. We use $\|x\|$ to denote a *generalized norm* of x, which can be any function $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$. In particular, all norms and seminorms are generalized norms (a seminorm $\|\cdot\|$ satisfies scalability $\|\alpha x\| = |\alpha| \, \|x\|$ for any scalar α, as well as the triangle inequality $\|x + y\| \leq \|x\| + \|y\|$, but not necessarily positive-definiteness, which means it is possible that $\|x\| = 0$ for $x \neq 0$). The set $\mathbb{B}_\epsilon(x)$ denotes the n-ball centered at x, i.e., $\mathbb{B}_\epsilon(x) \coloneqq \{y \in \mathbb{R}^n : |y - x| \leq \epsilon\}$; when the argument x is omitted, it is understood that $x = 0$. Sequences are denoted using boldface symbols, e.g., $\mathbf{x} \coloneqq (x(0), x(1), \dots)$, with $\mathbf{x}_{i:j}$ giving the subsequence $(x(i), \dots, x(j))$. They may be finite or infinite. The norm of a sequence is defined as $\|\mathbf{x}\| \coloneqq \sup_{k \geq 0} \|x(k)\|$ (for finite sequences, substituting $\max$ is equivalent). For a function $f(\,\cdot\,)$, its sublevel sets are denoted $\operatorname{lev}_\gamma f(\,\cdot\,) \coloneqq \{x : f(x) \leq \gamma\}$. For two sets A and B, $A \oplus B \coloneqq \{a + b : a \in A, b \in B\}$ is their Minkowski sum. A function $\alpha : [0, a] \to \mathbb{R}_{\geq 0}$ with $a > 0$ is said to be class $\mathcal{K}$ if it is zero at zero, continuous, and strictly increasing; it is class $\mathcal{K}_\infty$ if in addition its domain is all of $\mathbb{R}_{\geq 0}$ and it is unbounded. A function $\beta : \mathbb{R} \times \mathbb{T} \to \mathbb{R}_{\geq 0}$ is said to be class $\mathcal{KL}$ if for each $t \in \mathbb{T}$, the partial $\beta(\,\cdot\,, t)$ is class $\mathcal{K}_\infty$ and for any $s \in \mathbb{R}_{\geq 0}$, $\lim_{t \to \infty} \beta(s, t) = 0$. See Kellett (2014) for mathematical properties of $\mathcal{K}$ and $\mathcal{KL}$ functions, which are used throughout.

If you don't know where you're going, you probably won't get [asymptotic stability].

— Yogi Berra

Chapter 2

Mixed-Integer Model Predictive Control

2.1 Introduction

As computer hardware becomes faster and mixed-integer optimization techniques become more advanced, it becomes increasingly possible to solve mathematical programming problems in real time. Advanced control strategies such as model predictive control (MPC) leverage this possibility by optimizing a mathematical model to predict future behavior of the system of interest. By performing these calculations in real time, it is possible to react quickly to changing internal or external conditions. In this chapter, we provide a mathematical formulation for and prove theorems about state-space MPC in which decision variables can be both continuous and discrete. Notation is stated in Section 1.2 and is generally standard. The only nonstandard notation is that $|\cdot|$ is used for the Euclidean norm on $\mathbb{R}^n$, while $\|\cdot\|$ is used to denote any *generalized norm*, which includes all seminorms and point-to-set distance functions $\|x\|_X := \inf_{x' \in X} |x - x'|$ for a set $X \subseteq \mathbb{R}^n$.

2.1.1 State-Space Dynamic Systems

As a mathematical framework for process optimization and control, we consider dynamic systems in state-space form. In this formalism, the system is described at each instant by a state vector $x \in \mathbb{R}^n$. This vector holds all of the information about past history of the system (insofar as it can affect future behavior). The state of the system can be altered by inputs $u \in \mathbb{R}^m$, which are chosen by a controller in order to achieve certain objectives. The system may also be affected by disturbances $w \in \mathbb{R}^l$. These values are essentially inputs to the system that come from external sources and thus cannot be chosen by the controller. The convention is that the *nominal* system behavior is defined

by $w \equiv 0$. It is often the case that the states and inputs are restricted to lie in sets $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbb{U} \subseteq \mathbb{R}^m$ respectively. Typically, $\mathbb{X}$ is assumed to be closed, and $\mathbb{U}$ is assumed to be compact. In general, the constraints $u \in \mathbb{U}$ must always be satisfied (e.g., due to physical limitations of the actuators), while the constraints $x \in \mathbb{X}$ are desirable but could be potentially violated due to the effect of the disturbance w. In the case of shared state and input constraints, we use the set $\mathbb{Z} \subseteq \mathbb{X} \times \mathbb{U}$.

Continuous-Time Systems

In continuous time, the system evolves according to the ODE model

$$\frac{dx}{dt} = F(x, u, w, t). \tag{2.1}$$

Here, $F : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the system model. From this form, we see that knowledge of the current x and the future values of u and w are sufficient to predict the future trajectory of the system state. Note that in this case, $t \in \mathbb{R}_{\geq 0}$ is a continuous variable, and $x(\,\cdot\,)$, $u(\,\cdot\,)$, and $w(\,\cdot\,)$ are all functions of t.

For implementation purposes, it is common to assume that the input is a piecewise-constant function (a "zero-order hold") or a piecewise-linear function (a "first order hold") of time with a fixed sample time Δ. In a zero-order hold, $u(t)$ is constant on each interval $[k\Delta, (k+1)\Delta)$, while for a first-order hold, the input changes linearly on each interval. It is also possible to apply orthogonal collocation methods (Villadsen and Michelsen, 1978) which use a fixed-order polynomial for $u(t)$ within each sampling interval. All three cases allow the time-varying function $u(\,\cdot\,)$ to be described in terms of a finite number of parameters.

Discrete-Time Systems

Due to the mathematical subtleties associated with ODE systems, it is common to instead use a discrete-time representation of the system. In the discrete-time setting, the functions of (continuous) time become sequences $\mathbf{x} \coloneqq (x(t), x(t+1), \dots)$, with similar definitions for $\mathbf{u}$ and $\mathbf{w}$. In discrete time, the system evolution is given by

$$x(t+1) = f(x(t), u(t), w(t), t). \tag{2.2}$$

We will often use the shorthand notation $x^+ = f(x, u, w, t)$. Under nominal conditions, we will omit the disturbance w (i.e., $x^+ = f(x, u, t)$) for brevity. Note that for a constant sample time and a specified hold, the continuous-time system (2.1) can be exactly transformed to a discrete-time system of the form (2.2). That is, the corresponding discrete-time states $x(t)$ will agree with the continuous-time states $x(k\Delta)$ for nonnegative integers k.

In many cases, it is not possible to measure all the values of the state x. In such cases, it is common to define a vector of outputs $y \in \mathbb{R}^p$ given by

$$y = h(x, u, t) + v,$$

in which v is a specific type of disturbance commonly referred to as measurement noise. It is assumed that these quantities can be measured directly, and under certain conditions, measurements of y along with knowledge of u are sufficient to determine the state x. Note that throughout this chapter (and indeed throughout the majority of this thesis), we will work directly with the state vector x, rather than the outputs y.

A common class of discrete-time system is the linear system, in which the model $f(\,\cdot\,)$ is of the form

$$f(x, u, t) = A(t)x + B(t)u + c(t), \tag{2.3}$$

in which $A(t)$, $B(t)$, and $c(t)$ are time-varying matrices and vectors of the appropriate size. In the time-invariant case, the parameters no longer depend on time. As written, (2.3) is technically an affine function rather than a linear function, and so it is common to shift the system so that $f(0, 0, t) = 0$ (i.e., so that $c(t) \equiv 0$). Two generalizations of this form are piecewise affine (PWA) systems (Camacho et al., 2010) and mixed-logical dynamical (MLD) systems (Bemporad and Morari, 1999). In the form of (2.2), PWA systems are defined by

$$x^+ = A_i x + B_i u + c_i, \qquad (x, u) \in \mathcal{Z}_i \tag{2.4}$$

with $(\mathcal{Z}_i)_{i=1}^I$ a partitioning of feasible (x, u) space and $(A_i, B_i, c_i)_{i=1}^I$ the corresponding affine models for state evolution. Such models can be used to give linear approximations of nonlinear systems with multiple equilibria. Note that if each $\mathcal{Z}_i$ is polyhedral, then the entire system can be described in terms of linear relationships. MLD systems are defined similarly by

$$x^+ = Ax + B_1 u_1 + B_2 u_2 \tag{2.5}$$

in which $u = (u_1, u_2)$ is explicitly partitioned into continuous-valued and discrete-valued subvectors. Note that the MLD formulation includes additional "auxiliary" states as needed to enforce logical propositions that define the feasible set $\mathbb{Z}$. A final class of system is the switched system (Heidarinejad et al., 2013), which is of the form

$$f(x, u, t) = f_{u_1}(x, u_2, t), \quad u_1 \in \{0, 1, \ldots, S\}.$$

In this form, the discrete input u_1 chooses from among a finite set of system models $f_i(\,\cdot\,)$, while the remaining continuous inputs u_2 function as usual.

Hybrid Systems

Within the control literature, it is common to refer to any system with discrete-valued actuators as a "hybrid system" (Baotić et al., 2006; Bemporad and Morari, 1999; Camacho et al., 2010; Di Cairano et al., 2014; Karer et al., 2008; Lazar et al., 2006; Pregelj and Gerkšič, 2010; Rivotti and Pistikopoulos, 2015). Whether these types of systems are called "hybrid" is perhaps a matter of personal preference, but for clarity we refer to these as mixed-integer dynamical systems and reserve the term "hybrid" for a more general class of systems.

A true hybrid system is essentially the combination of discrete and continuous time, typically defined as follows (Branicky et al., 1998; Goebel et al., 2012):

$$\begin{cases} \dfrac{dx}{dt} = f(x,u) & x \in C \\ x^+ = g(x,u) & x \in D. \end{cases} \tag{2.6}$$

Here, $f(\,\cdot\,)$ is called the "flow map", C the "flow set", $g(\,\cdot\,)$ the "jump map", and D the "jump set." As long as $x \in C$, the system flows in continuous time according to the flow map, while when $x \in D$, the system instantaneously "jumps" according to the jump map. As before, x contains the system states and u contains the system inputs. More details about this formalism can be found in Goebel et al. (2009, 2012).

2.1.2 Model Predictive Control

Thus far, we have discussed mathematical formalism for modeling dynamical systems, but we have not discussed how the inputs u are chosen. For small systems it is common to choose a heuristic control law. For example, the classic PI control law defines a specific output variable $y_i \coloneqq h_i(x)$ and chooses (discrete-time) components $u_i(t)$ of the input according to

$$u_i(t) = u^{\text{sp}}(t) + K\left(\left(y_i^{\text{sp}}(t) - h_i(x(t))\right) + \frac{\Delta}{\tau}\sum_{t'=0}^{t-1}\left(y_i^{\text{sp}}(t') - h_i(x(t'))\right)\right),$$

in which y^{sp} and u^{sp} define the desired setpoint, while K and τ are tuning parameters. Such controllers tend to perform well when there are only one or two inputs, or when the system dynamics are fairly simple. However, if there are significant interactions among the various states and inputs, if there are not an equal number of inputs and outputs, or if there are difficult constraints that must be satisfied, these techniques are often insufficient.

To address the limitations of classical control techniques, a common advanced control method is model predictive control (García et al., 1989; Mayne et al., 2000; Rawlings et al., 2017b). Industrial use of model predictive control (MPC) has grown significantly since its original development (Qin and Badgwell, 2003), and its range of application has been steadily increasing. To make control decisions, MPC takes advantage of the fact that if the process model $f(\,\cdot\,)$ is known (or can be approximated), then it is possible to predict the effect that a given sequence $\mathbf{u}$ of inputs will have on the future state trajectory $\mathbf{u}$. The goal of the control system is transcribed into a mathematical cost function, e.g.,

$$\ell(x,u,t) = \left|x - x^{\text{sp}}(t)\right|^2 + \left|u - u^{\text{sp}}(t)\right|,$$

so that minimizing this cost function will correspond to achievement of control objectives such as $x(t) \to x^{\text{sp}}(t)$ and $u(t) \to u^{\text{sp}}(t)$.

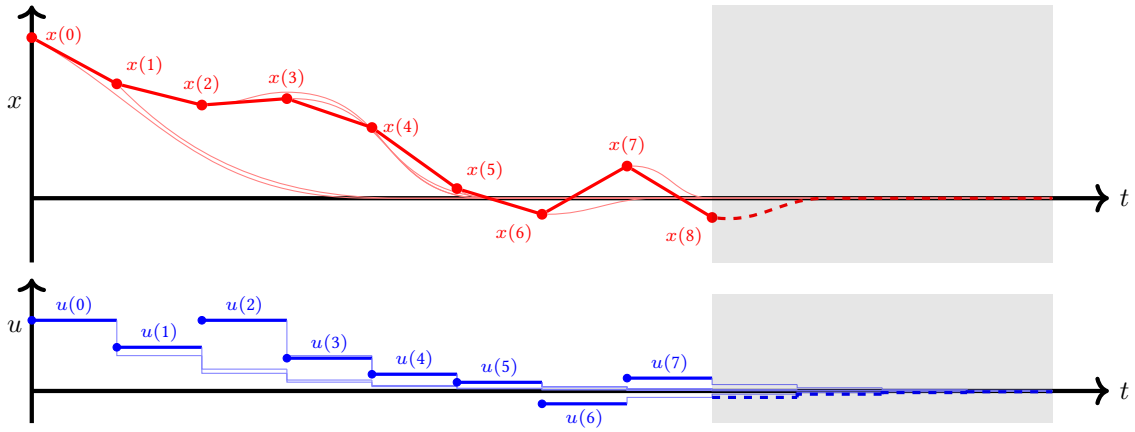


FIGURE 2.1: Illustration of closed-loop MPC. Thick dark lines indicate closed-loop trajectory, while thin light lines show past open-loop predictions. Shaded boxes show current prediction horizon, with thick dashed lines showing incumbent solution.

To choose inputs, the MPC controller solves a finite-horizon optimal control problem of the form

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{u}} \quad & \sum_{k=0}^{N-1} \ell(x(k), u(k), t+k) && && \text{Stage Cost} && (2.7a)\\
& \qquad + V_f(x(N), t+N) && && \text{Terminal Cost} && (2.7b)\\
\text{s.t.} \quad & x(k+1) = f(x(k), u(k), t+k), \quad && k \in \mathbb{I}_{[0,N)} && \text{System Model} && (2.7c)\\
& (x(k), u(k)) \in \mathbb{Z}(t+k) && && \text{Path Constraints} && (2.7d)\\
& x(0) = x && && \text{Initial Condition} && (2.7e)\\
& x(N) \in \mathbb{X}_f(t+N) && && \text{Terminal Constraint} && (2.7f)
\end{aligned}
$$

The various components will be discussed later in this section. This optimization problem finds a predicted sequence of states $\mathbf{x}$ and $\mathbf{u}$ with $N+1$ and N elements respectively. In certain optimal control formulations, these sequences would be followed completely to their termination. However, due to disturbances, model mismatch, changes in setpoints, etc., a solution that is optimal at one timestep may become suboptimal or even infeasible from system's new state. MPC accounts for these effects by sending only the *first* input $u(0)$ to the system and solving problem (2.7) again at the next timestep. Thus, the path that the system follows (the "closed-loop" trajectory) does not necessarily match the optimal predictions (the "open-loop" trajectories), even when the model $f(\,\cdot\,)$ is completely accurate and there are no disturbances. We illustrate this process in Figure 2.1. In this example, the system encounters a large disturbance during the transition from $x(1)$ to $x(2)$, which leads to large revision in the predicted trajectory. At all other timesteps, revisions are much smaller.

Due to the horizon being shifted at each time point, MPC strategies are often referred to as "rolling-horizon" or "moving-horizon" implementations. The key feature is that the horizon always stays the same length, and thus at each timestep, the previous time point is removed from the beginning of the horizon, and a single extra time point is added to the end. Note that this is in contrast to "shrinking-horizon" strategies, common for batch applications,

in which no new time point is added. Due to the principle of optimality, the optimal open-loop solution does not change when the horizon shrinks. By contrast, when the horizon rolls, the extra time point is an additional source of flexibility, and thus the optimizer may revise the solution, even under nominal operation.

Historically, MPC has been developed for and limited to setpoint tracking using continuous-valued inputs (García et al., 1989). Although constraints could be considered, they were (and sometimes still are) limited to linear or convex constraints. One of the main goals of this chapter is to show that the (implicit or explicit) assumption of continuous-valued inputs is unnecessary, and standard MPC formulations and theorems are compatible with discrete-valued actuators. This extension allows additional, often higher-level, discrete decisions (e.g., yes or no, on or off, how many) to be optimized within the same rolling horizon framework. A fairly recent extension of economic MPC allows the objective function to be a tangible measure of process profitability rather than am abstract measure of distance from the setpoint (Amrit et al., 2011). We hold off on discussing this and other developments until the next chapter, and instead focus on tracking-oriented MPC here. Note that we are not considering the full class of hybrid systems as discussed in Section 2.1.1, but rather focusing on discrete-time systems in which some of the inputs (or states) are constrained to be discrete-valued. However, as we will see in the examples (and throughout the rest of this thesis), a wide variety of systems can be described using this class of system. In analogy with "mixed-integer" optimization (Belotti et al., 2013), we refer to this case as mixed-integer model predictive control, or MIMPC.

2.2 Tracking Model Predictive Control

In this section, we formulate and present stability properties of tracking-oriented MPC in which the goal is to bring the state of the system to the origin and keep it there. For full generality, we allow the system model, constraints, and objective function to be time-varying. As will be discussed later, this formulation also allows tracking an arbitrary steady state, periodic cycle, or general feasible trajectory. We start by providing a mathematical definition of the stabilization goal, and then we present an MPC formulation to achieve said goal.

2.2.1 Stability

Consider a autonomous discrete-time, time-varying system of the form

$$x(t+1) = f(x(t), t)$$

with the function $f(\,\cdot\,)$ defined on sets $\mathbb{X}(t)$ for each $t \in \mathbb{T}$. We take the following standard definitions:

Definition 2.1 (Positive invariance): Sets $\mathbb{X}(t)$ are said to be positive invariant under the system $x^+ = f(x,t)$ if $f(x,t) \in \mathbb{X}(t+1)$ for all $x \in \mathbb{X}(t)$ and $t \in \mathbb{T}$.

Definition 2.2 (Asymptotic stability): The system $x^+ = f(x,t)$ is asymptotically stable on $\mathbb{X}(t)$ with respect to the generalized norm $\|\cdot\|$ if the sets $\mathbb{X}(t)$ are positive invariant and there exists a $\mathcal{KL}$ function $\beta(\,\cdot\,)$ such that the system satisfies

$$\|x(t+k)\| \leq \beta(\|x(t)\|, k)$$

for any $t \in \mathbb{T}$ and $x \in \mathbb{X}(t)$.

Definition 2.3 (Lyapunov function): A function $V : \mathbb{X} \times \mathbb{T} \to \mathbb{R}_{\geq 0}$ is said to be a Lyapunov function for system $x^+ = f(x,t)$ and generalized norm $\|\cdot\|$ on $\mathbb{X}(t)$ if the sets $\mathbb{X}(t)$ are positive invariant and there exist $\mathcal{K}_\infty$ functions $\alpha_1(\,\cdot\,)$, $\alpha_2(\,\cdot\,)$ and $\alpha_3(\,\cdot\,)$ that satisfy

$$\alpha_1(\|x\|) \leq V(x,t) \leq \alpha_2(\|x\|) \tag{2.8}$$
$$V(f(x,t),t+1) \leq V(x,t) - \alpha_3(\|x\|) \tag{2.9}$$

for all $x \in \mathbb{X}(t)$ and $t \in \mathbb{T}$.

Note that in both definitions, we allow for the possibility that the norm $\|\cdot\|$ is a *generalized norm.* This generalization allows for the possibility that $\|x\| = 0$ for some $x \neq 0$. For example, point-to-set distance $\|x\|_X \coloneqq \min_{x' \in X} |x - x'|$, which assigns $\|x\|_X = 0$ for all $x \in X$.

The concepts of asymptotic stability and Lyapunov functions are related by the following theorem:

Theorem 2.4 (Lyapunov stability theorem): *Suppose the sets $\mathbb{X}(t)$ are positive invariant for the system $x^+ = f(x,t)$ and that a function $V(\,\cdot\,)$ is a Lyapunov function for $f(\,\cdot\,)$. Then, the system is asymptotically stable.*

Proof. For notational convenience, let $f^k(x,t)$ be composition of $f(\,\cdot\,)$ with itself k times, i.e.,

$$f^1(x,t) = f(x,t), \quad f^2(x,t) = f(f(x,t),t+1), \quad f^3(x,t) = f(f(f(x,t),t+1,t+2)),$$

etc. Because the sets $\mathbb{X}(t)$ are positive invariant, we have $f^k(x,t) \in \mathbb{X}(t+k)$ for all $k \in \mathbb{I}_{\geq 0}$. From (2.8), we have that $\|x\| \geq \alpha_2^{-1}(V(x,t))$. Substituting into (2.9) gives the condition

$$V(f(x,t),t+1) \leq V(x,t) - \alpha_3(\alpha_2^{-1}(V(x,t))) \coloneqq \sigma_1(V(x,t))$$

for function $\sigma_1(s) \coloneqq s - \alpha_3(\alpha_2^{-1}(s))$. Now, $\sigma_1(\,\cdot\,)$ is zero at zero and continuous, but it is not necessarily increasing. To provide an increasing bound, we first define

$$\sigma_2(s) \coloneqq \max_{s' \in [0,s]} \sigma_1(s') \geq \sigma_1(s).$$

This function is nondecreasing (obvious) and continuous (because it is the maximum of a continuous set-valued map). For an increasing function, we define

$$\sigma_3(s) \coloneqq \frac{1}{2}(s + \sigma_2(s)) \geq \sigma_2(s) \geq \sigma_1(s),$$

which is a $\mathcal{K}_\infty$ function. Note also that $\sigma_3(s) < s$ for all positive s. Thus, substituting this bound into (2.2.1) and composing k times, we have

$$V(f^k(x,t), t+k) \leq \sigma_3^k(V(x,t)),$$

in which $\sigma_3^k(\,\cdot\,)$ is the k-times composition of $\sigma_3(\,\cdot\,)$. Applying (2.8), we have

$$\left\| f^k(x,t) \right\| \leq \beta(\|x\|, k), \qquad \beta(s,t) \coloneqq \alpha_1^{-1}\left(\sigma_3^k(\alpha_2(s))\right)$$

in which $\beta(\,\cdot\,)$ is class $\mathcal{KL}$ due to the properties of $\mathcal{K}_\infty$ functions. Thus, the system is asymptotically stable. □

The usefulness of Theorem 2.4 is that it provides an indirect route to prove asymptotic stability: rather than examine the global properties $f^k(\,\cdot\,)$ directly, we can instead search for a Lyapunov function. In the following section, we will define an optimization problem whose optimal value function is a Lyapunov function, thus proving asymptotic stablity for the system.

Remark 2.5: As will become apparent in the following sections, the most difficult part of constructing a Lyapunov function is typically the upper bound $V(x,t) \leq \alpha_2(\|x\|)$. However, the decrease condition (2.9) alone is sufficient to prove the convergence result $\|x(t+k)\| \to 0$ as $k \to 0$: the sequence $V(f^k(x,t), t+k)$ is bounded and nondecreasing and thus has a limit (monotone convergence theorem). Therefore, for any $\epsilon > 0$, there exists $K \geq 0$ such that

$$V(f^k(x,t), t+k) - V(f^{k+1}(x,t), t+k+1) \leq \epsilon$$

for all $k \geq K$ Thus, applying (2.9) and choosing $\epsilon' = \alpha_3^{-1}(\epsilon)$, we have $\|x(t+k)\| \leq \epsilon'$ for all $k \geq K$, and the limit follows. Although convergence may be sufficient for some applications, it lacks certain desirable properties of asymptotic stability. For example, asymptotic stability implies the classical Lyapunov stability property that for each $\epsilon > 0$, there exists $\delta > 0$ such that if $\|x(t)\| \leq \delta$ then $\|x(t')\| \leq \epsilon$ for all $t' \geq t$.

2.2.2 Optimal Control Problem

To provide sufficient generality, we consider the case of time-varying discrete-time state-space systems that evolve according to

$$x(t+1) = f(x(t), u(t), t), \tag{2.10}$$

for states $x \in \mathbb{X}(t)$ and inputs $u \in \mathbb{U}(t)$. Choose combined constraints $\mathbb{Z}(t) \subseteq \mathbb{X}(t) \times \mathbb{U}(t)$. Let

$$\mathbb{X} \coloneqq \bigcup_{t \in \mathbb{T}} \mathbb{X}(t), \quad \mathbb{U} \coloneqq \bigcup_{t \in \mathbb{T}} \mathbb{U}(t), \quad \mathbb{Z} \coloneqq \bigcup_{t \in \mathbb{T}} \mathbb{Z}(t)$$

denote the composite sets. For each $t \in \mathbb{T}$, we choose a terminal set $\mathbb{X}_f(t) \subseteq \mathbb{X}(t)$, whose required properties will be discussed later. For a finite input sequence $\mathbf{u} \coloneqq (u(0), u(1), \ldots, u(N-1))$, let $\phi(k, x, \mathbf{u}, t)$ denote the forward

solution to (2.10) starting from state x at time t, i.e., giving $x(t+k) = \phi(k, x, \mathbf{u}, t)$. For a given $N \in \mathbb{T}$, we then define the following three sets:

$$\begin{aligned}
\mathcal{Z}_N(t) &\coloneqq \{(x, \mathbf{u}) : (\phi(k, x, \mathbf{u}, t), u(k)) \in \mathbb{Z}(t+k) \text{ for all } k \in \mathbb{I}_{[0,N)}, \phi(N, x, \mathbf{u}, t) \in \mathbb{X}_f(t+N)\} \\
\mathcal{X}_N(t) &\coloneqq \{x \in \mathbb{X} : \text{there exists } \mathbf{u} \in \mathbb{U}^N \text{ such that } (x, \mathbf{u}) \in \mathbb{Z}_N(t)\} \\
\mathcal{U}_N(x, t) &\coloneqq \{\mathbf{u} \in \mathbb{U}^N : (x, u) \in \mathcal{Z}_N(t)\}
\end{aligned}$$

As an objective function, we define

$$V_N(x, \mathbf{u}, t) \coloneqq \sum_{k=0}^{N-1} \ell(x(k), u(k), t+k) + V_f(x(N), t+N),$$

in which $x(k) \coloneqq \phi(k, x, \mathbf{u}, t)$ for $k \in \mathbb{I}_{[0,N]}$. Here, $\ell(\,\cdot\,)$ is referred to as the "stage cost," and $V_f(\,\cdot\,)$ is the "terminal cost."

At each discrete timestep t, the controller solves the optimal MPC problem

$$\min_{\mathbf{u}} \; V_N(x, \mathbf{u}, t) \text{ such that } \mathbf{u} \in \mathcal{U}_N(x, t) \tag{2.11}$$

given the current state of the system $x \in \mathbb{X}(t)$. We denote the optimal value function as $V_N^*(x, t)$ and the optimal solution set as $\mathcal{U}_N^*(x, t)$. To use as the current input, the optimizer selects any $u \in \kappa_N(x, t) \coloneqq \{u(0) : \mathbf{u} \in \mathcal{U}^*(x, t)\}$, i.e., the *first* input of any optimal sequence. Thus, the closed-loop evolution of the system is given by

$$x(t+1) \in \{f(x, u, t) : u \in \kappa_N(x, t)\}.$$

Note that this is a difference inclusion due to the fact that there may be multiple optimal solutions to (2.11). For convenience, we assume that a selection rule is applied to choose a single unique input $u = \kappa_N(x, t)$, which gives a standard difference equation

$$x(t+1) = f(x, \kappa_N(x, t), t). \tag{2.12}$$

Any possible selection rule can be chosen provided that the *same* choice is made for equal values of x. The goal of the controller is to provide asymptotic stability for this system. Note that the optimal control law $\kappa_N(\,\cdot\,)$ may not be continuous, and so stability theory should be able to address this case.

2.2.3 Assumptions

To prove closed-loop properties of the system, we take the following standard assumptions (see, e.g., Rawlings et al. (2017b)).

Assumption 2.6 (Basic function properties): The function $f(\,\cdot\,)$ is continuous. The functions $\ell(\,\cdot\,)$ and $V_f(\,\cdot\,)$ are lower semi-continuous.

Assumption 2.7 (Basic set properties): For each $t \in \mathbb{T}$, the sets $\mathbb{X}(t)$, $\mathbb{X}_f(t)$, and $\mathbb{Z}(t)$ are closed, and the set $\mathbb{U}(t)$ is compact. The composite set $\mathbb{U}$ is bounded.

Assumption 2.8 (Cost bounds): There exists a $\mathcal{K}_\infty$ function $\alpha(\,\cdot\,)$ such that $\ell(x,u,t) \geq \alpha(\|x\|)$ for all $x \in \mathbb{Z}(t)$ and $V_f(x,t) \geq 0$ for all $x \in \mathbb{X}_f(t)$, both for each $t \in \mathbb{T}$.

Assumption 2.9 (Terminal control law): For each $t \in \mathbb{T}$ and $x \in \mathbb{X}_f(t)$, the set

$$\kappa_f(x,t) = \big\{u \in \mathbb{U}(t) : f(x,u,t) \in \mathbb{X}_f(t+1),\ V_f(f(x,u,t),t+1) \leq V_f(x,t) - \ell(x,u,t)\big\}$$

is nonempty.

Assumption 2.10 (Uniform weak controllability): There exists a $\mathcal{K}_\infty$ function $\alpha_2(\,\cdot\,)$ satisfying $V_N^*(x,t) \leq \alpha_2(\|x\|)$ for all $x \in \mathcal{X}_N(t)$ and $t \in \mathbb{T}$.

Assumptions 2.6 and 2.7 ensure the existence of optimal solutions to (2.11). They imply that the objective function $V_N(x,\mathbf{u},t)$ is lower-semicontinuous in x and $\mathbf{u}$ and that the feasible set $\mathcal{U}_N(x,t)$ is compact. Assumptions 2.8 and 2.9 are used to derive bounds on closed-loop cost and to ensure recursive feasibility. The final Assumption 2.10 is somewhat more technical than the previous assumptions. This bound ensures that the optimal cost function decays to zero as $\|x\|$ approaches zero. It can be shown that Assumption 2.10 holds in a variety of other circumstances as described in the following proposition (adapted from Rawlings et al. (2017b, Proposition 2.38)).

Proposition 2.11 (Conditions for uniform weak controllability): *Suppose the functions $f(\,\cdot\,)$, $\ell(\,\cdot\,)$, and $V_f(\,\cdot\,)$ are uniformly locally bounded for all $t \in \mathbb{T}$, i.e., on any compact set $Z \subset \mathbb{X} \times \mathbb{U}$, the set*

$$\big\{(f(x,u,t),\ell(x,u,t),V_f(x,t)) : (x,u) \in Z, t \in \mathbb{T}\big\}$$

is bounded. Suppose further that $\|\cdot\|$ is the standard Euclidean norm $|\cdot|$. Then, Assumption 2.10 is satisfied if any of the following conditions holds:

(a) Assumptions 2.8 and 2.9 hold, $V_f(x,t)$ is uniformly continuous at the origin, and there exists $\epsilon > 0$ such that $\mathbb{B}_\epsilon \cap \mathcal{X}_N(t) \subseteq \mathbb{X}_f(t)$ for each $t \in \mathbb{T}$.

(b) For $t \in \mathbb{T}$, the optimal value function $V_N^0(x,t)$ is uniformly continuous at $x = 0$.

(c) The functions $f(\,\cdot\,)$ and $\ell(\,\cdot\,)$ are uniformly continuous at the origin $(x,u) = (0,0)$ for all $t \in \mathbb{T}$, and the system is stabilizable with small inputs, i.e., there exists a $\mathcal{K}_\infty$ function $\gamma(\,\cdot\,)$ such that for all $t \in \mathbb{T}$ and $x \in \mathcal{X}_N(t)$, there exists $\mathbf{u} \in \mathcal{U}_N(x,t)$ with $|\mathbf{u}| \leq \gamma(|x|)$.

(d) There exist $\epsilon > 0$ and a $\mathcal{K}_\infty$ function $\alpha(\,\cdot\,)$ such that $V_N^*(x,t) \leq \alpha(|x|)$ for all $t \in \mathbb{T}$ and $x \in \mathbb{B}_\epsilon \cap \mathcal{X}_N(t)$

Proof. See Section 2.6.3. □

REMARK 2.12: Although the general MPC problem is formulated in terms of any generalized norm $\|\cdot\|$, it can be difficult to verify Assumption 2.10 in certain exotic cases, for example a point-to-set distance $\|\cdot\|_X$ when the set X is nonconvex. As discussed in Proposition 2.11, when $\|\cdot\|$ is the Euclidean norm $|\cdot|$ (or any other true norm, due to the equivalence of norms on $\mathbb{R}^n$), Assumption 2.10 boils down to a statement about the behavior of $V_N^*(\cdot)$ at the origin, which is relatively easy to investigate. In certain other "nice" cases, it may also be possible to apply the ideas of Proposition 2.11. For example, if the goal is to bring the system near the origin, one could choose $\|\cdot\|_{\mathbb{B}_\epsilon}$ so that there is no penalty for being within ϵ of the origin. If the terminal set is chosen so that $\mathbb{B}_\delta \subseteq \mathbb{X}_f(t)$ for $\delta > \epsilon$, then the arguments of Proposition 2.11 (a) and (d) can be used to show Assumption 2.10.

2.2.4 Asymptotic Stability

The main result of this section is the following theorem, which provides asymptotic stability for tracking MPC.

THEOREM 2.13 (Asymptotic stability of tracking MPC): *Suppose Assumptions 2.6 to 2.10 are satisfied. Then, the closed-loop system* (2.12) *is asymptotically stable starting from any $x \in \mathcal{X}_N(t)$ for any $t \in \mathbb{T}$.*

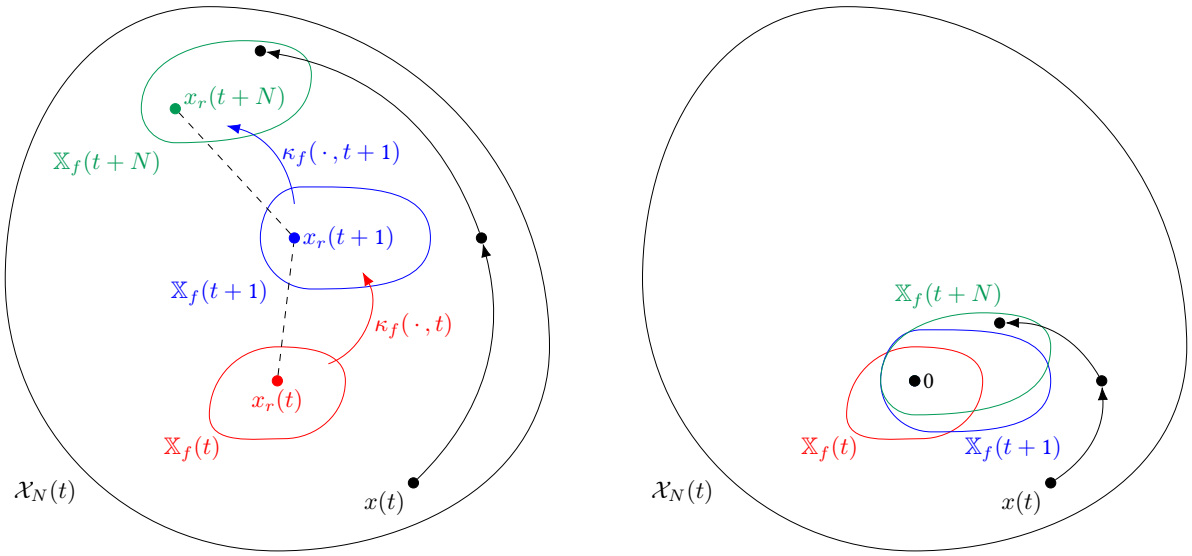
Proof. We show that the optimal cost function $V_N^*(\cdot)$ is a Lyapunov function for the closed-loop system. First, we note that from Assumption 2.6, the objective function $V_N(x, \mathbf{u}, t)$ is lower semi-continuous in x and $\mathbf{u}$, as $\phi(k, x, \mathbf{u}, t)$ is continuous, and thus $V_N(\cdot)$ is the sum of lower semi-continuous functions. Next, because each $\mathbb{U}(t)$ is compact and each $\mathbb{X}_f(t)$ closed by Assumption 2.7, the set $\mathcal{U}_N(x, t)$ is compact for each x and t. Thus, problem (2.11) is the optimization of a lower semi-continuous function over a compact set, which means an optimal solution exists for every $x \in \mathcal{X}_N(t)$ and $t \in \mathbb{T}$, and so $V^*(x, t)$ is well-defined. Next by Assumption 2.8, we have $V_N(x, \mathbf{u}, t) \geq \ell(x, u(0), t) \geq \alpha_1(\|x\|)$, and thus $V^*(x, t) \geq \alpha_1(\|x\|)$. The upper bound $V^*(x, t) \leq \alpha_2(\|x\|)$ comes directly from Assumption 2.10. Therefore, $V^*(\cdot)$ satisfies (2.8), and it remains to show (2.9).

For the decrease condition, let $\mathbf{u}$ be the optimal solution to (2.11) for x at time t, and let u be its first element. Let $x^+ \coloneqq f(x, u, t)$ and let $x_f \coloneqq \phi(N, x, \mathbf{u}, t)$ be the terminal value of x. By Assumption 2.9, there exists an input $u_f \in \mathbb{U}(t+N)$ such that $x_f^+ \coloneqq f(x_f, u_f, t+N) \in \mathbb{X}_f(t+N+1)$. Now, consider the input sequence $\mathbf{u}^+ \coloneqq (u(1), u(2), \ldots, u(N-1), u_f)$ constructed by concatenating the final $N-1$ elements of $\mathbf{u}$ and u_f. This new input sequence satisfies $\mathbf{u}^+ \in \mathcal{U}(x^+, t+1)$. In addition, we have

$$\begin{aligned} V_N(x^+, \mathbf{u}^+, t+1) &= V_N^*(x, t) - \ell(x, u, t) - V_f(x_f, t+N) + \ell(x_f, u_f, t+N) + V_f(x_f^+, t+N+1) \\ &\leq V_N^*(x, t) - \ell(x, u, t) \end{aligned}$$

because the sum of the final three terms is nonpositive by Assumption 2.9. Since this cost bound holds for a feasible solution, we know that the optimal solution must satisfy it as well, and so

$$V_N^*(x^+, t+1) \leq V_N^*(x, t) - \ell(x, u, t) \leq V_N^*(x, t) - \alpha_1(\|x\|)$$



(A) Sets in original space. Reference trajectory $x_r(t)$ is time-varying, and terminal control law $\kappa_f(\cdot, t)$ stabilizes neighborhoods of $x_r(t)$

(B) Sets in transformed deviation-variable space. Terminal regions $\mathbb{X}_f(t)$ are now all neighborhoods of the origin.

FIGURE 2.2: Illustration of time-varying change of variables. By shifting the origin by $-x_r(t)$ at each time t, tracking the reference trajectory is transformed into tracking the origin.

by Assumption 2.8. Since $\alpha_1(\cdot)$ is class $\mathcal{K}_\infty$, (2.9) is satisfied as well. Therefore, $V_N^*(\cdot)$ is a Lyapunov function, and by Theorem 2.4 the closed-loop system is asymptotically stable. □

REMARK 2.14: As formulated, Theorem 2.13 proves asymptotic stability to the origin. However, because the system is allowed to be time-varying, it also applies to stability of any feasible reference trajectory $(x_r(t), u_r(t)) \in \mathbb{Z}(t)$ that satisfies $f(x_r(t), u_r(t), t) = x_r(t+1)$ and $x_r(t) \in \mathbb{X}_f(t)$. For this formulation, Assumption 2.8 requires that $\ell(x, u, t) \geq \alpha_1(\|x - x_r(t)\|)$; by applying a time-varying change of variables, the desired trajectory is shifted to the origin, and Theorem 2.13 can be applied. Figure 2.2 illustrates this change of variables.

2.2.5 SUBOPTIMAL MPC

As formulated, closed-loop MPC requires the controller to find an *optimal* solution to (2.11) at each timestep. While achievable for certain classes of systems and cost functions (e.g., linear systems with convex costs), it is generally very difficult to quickly find optimal solutions (and *prove* the optimality thereof) for nonconvex problems. In particular, because of the equality constraints imposed by the system model (2.7c), any nonlinear $f(\cdot)$ leads to a nonconvex optimization problem. Although solvers such as IPOPT (Wächter and Biegler, 2006) can efficiently find *local* solutions to nonlinear programming problems, such solutions are not sufficient for Theorem 2.13.

To avoid these shortcomings, we propose a modification of the MPC formulation so that only suboptimal solutions are necessary. Various formulations for suboptimal MPC have been proposed in the literature. For example, in Lazar and Heemels (2009) and Picasso et al. (2012), the suboptimal problem is formulated as finding a solution within some absolute tolerance of the true optimal solution. While branch-and-bound-based global optimization methods can provide a conservative estimate of suboptimality (Belotti et al., 2013), it is in general difficult to predict how long it will take to find a solution within that tolerance. Thus, because these methods essentially require knowledge of the optimal objective function value (or at least a quality estimate of it), they do not address the fundamental limitation of optimal MPC. Instead, we follow the developments of Pannocchia et al. (2011) and Allan et al. (2017) which employ the use of a warm-start sequence and express suboptimality with respect to the cost of that solution (which is known), rather than with respect to the true optimal solution.

To formulate the suboptimal problem, we extend the state of the system to $z \coloneqq (x, \mathbf{u})$, in which $\mathbf{u}$ is the predicted future sequence of inputs. We then define the set of successor *warm starts*

$$\zeta(x, \mathbf{u}, t) \coloneqq \big\{(u(1), \dots, u(N-1), u_f) : u_f \in \kappa_f(\phi(N, x, \mathbf{u}, t), t+N)\big\},$$

in which $(u(1), \dots, u(N-1)) = \mathbf{u}_{1:N-1}$ (i.e., the final $N-1$ values of $\mathbf{u}$), and u_f is any element of the terminal control law for the (predicted) terminal value of x. Note that this warm-start is precisely the sequence $\mathbf{u}^+$ used in the proof of Theorem 2.13.

Because we are examining stability in terms of the extended state z, we require the following conditions on the generalized norm $\|z\|$ as in the following assumption:

Assumption 2.15 (Norm compatibility): The generalized norm $\|z\|$ of the extended state can be expressed as

$$\|z\| = \|x\|_x + \|\mathbf{u}\|_u$$

for generalized norms $\|\cdot\|_x$ and $\|\cdot\|_u$ defined on $\mathbb{X}$ and $\mathbb{U}$ respectively. The sublevel sets of $\|\cdot\|_x$ are compact in $\mathbb{X}$. There exists a $\mathcal{K}_\infty$ function $\alpha(\cdot)$ such that the cost function $\ell(\cdot)$ satisfies

$$\ell(x, u, t) \geq \alpha(\|x\|_x + \|u\|_u)$$

for all $(x, u) \in \mathbb{Z}(t)$ and $t \in \mathbb{T}$.

The goal of this suboptimal MPC formulation is that the warm starts are chosen so that they can stabilize the system despite not necessarily being optimal. That is, the system is asymptotically stable even if it follows $\mathbf{u}$ indefinitely (with appropriate extension by applying $\zeta(\cdot)$). While this property eliminates the need for any online optimization whatsoever, it creates the possibility for the system is already at the origin but decides to leave due to a poorly chosen warm start. For example, if the system $x^+ = x + u$ were allowed to follow the warm start $\mathbf{u} = (1, 1, -1, -1)$ starting

from $x = 0$, then asymptotic stability would not hold. To avoid such cases, we require $\mathbf{u}$ to be in the following restricted set:

$$\check{\mathcal{U}}_N(x,t) \coloneqq \{\mathbf{u} \in \mathcal{U}_N(x,t) : V_N(x,\mathbf{u},t) \leq \gamma(\|x\|_x) \text{ if } \|x\|_x \leq \epsilon\}, \tag{2.13}$$

with composite set $\check{\mathcal{Z}}_N(t) \coloneqq \mathcal{X}_N(t) \times \check{\mathcal{U}}(x,t)$. This set requires that, if x is near the origin, then warm start $\mathbf{u}$ must keep the system nearby. We assume the properties of $\check{\mathcal{U}}_N(x,t)$ are as follows:

Assumption 2.16: In the definition of $\check{\mathcal{U}}_N(x,t)$, the constant ϵ is strictly positive, and the function $\gamma(\,\cdot\,)$ is class $\mathcal{K}_\infty$. The set $\check{\mathcal{U}}_N(x,t)$ is nonempty for all $x \in \mathcal{X}_N(t)$ and $t \in \mathbb{T}$.

Thus, Assumption 2.16 ensures that warm starts are sufficiently small when $\|x\|_x$ is small. However, as with Assumption 2.10, Assumption 2.16 can be difficult to verify. Thus, we use the following proposition to provide a more readily verifiable condition:

Proposition 2.17: *Suppose Assumptions 2.8, 2.9 and 2.15 hold. Suppose in addition that* $\mathbb{X}_f(t) \subseteq \{x \in \mathcal{X}_N(t) : \|x\|_x \leq \epsilon\}$. *Then, Assumption 2.16 is satisfied for any* $\mathcal{K}_\infty$ *function* $\gamma(\,\cdot\,)$ *such that* $V_f(x,t) \leq \gamma(|x|)$ *for all* $x \in \mathbb{X}_f(t)$ *and* $t \in \mathbb{T}$.

Proof. We apply Corollary 2.23 in Section 2.6 to show that, for each $x \in \mathbb{X}_f(t)$, we can construct $\mathbf{u} \in \mathcal{U}_N(x,t)$ by repeated application of $\kappa_f(\,\cdot\,)$, and further that $V_N(x,\mathbf{u},t) \leq V_f(x,t)$. Thus, choosing any $\gamma(\,\cdot\,)$ larger than $V_f(\,\cdot\,)$, we have $\gamma(\|x\|_x) \geq V_f(x,t) \geq V_N(x,\mathbf{u},t)$ whenever $\|x\|_x \leq \epsilon$ as required. Therefore, we conclude that $\check{\mathcal{U}}_N(x,t)$ is nonempty, and Assumption 2.16 holds as well. □

With this technical restriction in place, we can now describe the "optimization" procedure for suboptimal MPC. At each timestep, given the value of $z \coloneqq (x,\mathbf{u})$, the optimizer chooses any element $\check{\mathbf{u}} \in \check{\mathcal{U}}_N^*(x,\mathbf{u},t)$ with

$$\begin{aligned}\check{\mathcal{U}}_N^*(x,\mathbf{u},t) \coloneqq \big\{\check{\mathbf{u}} \in \check{\mathcal{U}}_N(x,t) : V_N(x,\check{\mathbf{u}},t) \leq V_N(x,\mathbf{u},t),\\ \zeta(x,\check{\mathbf{u}},t) \in \check{\mathcal{U}}_N(x^+,t+1),\ x^+ = \phi(1,x,\check{\mathbf{u}},t)\big\}.\end{aligned} \tag{2.14}$$

In addition to the technical restriction that the successor warm start satisfies (2.13), we have the additional requirement that the cost of the chosen $\check{\mathbf{u}}$ must be no larger than that of the initial $\mathbf{u}$. If desired, optimization can be performed to improve on $\mathbf{u}$, but no optimization is required, as the bound holds trivially for $\check{\mathbf{u}} = \mathbf{u} \in \check{\mathcal{U}}_N(x,t)$.

With the sequence $\check{\mathbf{u}}$ chosen, the first element is used as the current input, and then $\zeta(\,\cdot\,)$ is used to construct a feasible warm start to use at the next timestep. Letting $\check{\kappa}_N(x,\mathbf{u},t)$ denote the first element of any $\check{\mathbf{u}} \in \check{\mathcal{U}}_N^*(x,\mathbf{u},t)$, the closed-loop evolution of the extended state z is

$$z^+ \coloneqq \begin{pmatrix} x^+ \\ \mathbf{u}^+ \end{pmatrix} = \begin{pmatrix} f(x,\check{\kappa}(x,\mathbf{u},t),t) \\ \zeta(x,\check{\mathbf{u}},t) \end{pmatrix}, \quad \check{\mathbf{u}} \in \check{\mathcal{U}}_N^*(x,\mathbf{u},t). \tag{2.15}$$

As in the case of optimal MPC, we assume that an arbitrary selection rule is applied so that (2.15) is a standard difference equation rather than an inclusion. We then have the following theorem:

THEOREM 2.18 (Asymptotic stability of suboptimal tracking MPC): *Suppose Assumptions 2.6 to 2.9, 2.15 and 2.16 are satisfied. Then, the closed-loop system* (2.15) *is asymptotically stable starting from any* $z \in \check{\mathcal{Z}}_N(t)$ *at time* $t \in \mathbb{T}$.

Proof. We show that $V_N(z,t) \coloneqq V_N(x, \mathbf{u}, t)$ is a Lyapunov function for (2.15). First, by assumption Assumption 2.16, $\check{\mathcal{U}}_N(x,t)$ is nonempty for every $x \in \mathcal{X}_N(t)$. Thus, positive invariance of $\check{\mathcal{Z}}_N(t)$ follows immediately from (2.14).

For the Lyapunov function bounds (2.8), we recall the definition of $\|z\|$ according to assumption Assumption 2.15. For a lower bound, we note that

$$\begin{aligned}
V_N(z,t) &\geq \sum_{k=0}^{N-1} \ell(x(k), u(k), t+k) \\
&\geq \sum_{k=0}^{N-1} \alpha(\|x(k)\|_x + \|u(k)\|_u) \\
&\geq \alpha(2\|x\|_x) + \sum_{k=0}^{N-1} \alpha(2\|u(k)\|_u) \\
&\geq \alpha(2\|x\|_x) + \alpha(2\|\mathbf{u}\|_u) \\
&\geq 2\alpha(2\|x\|_x + 2\|\mathbf{u}\|_u) \\
&\geq \alpha_1(\|z\|), \quad \alpha_1(s) \coloneqq 2\alpha(2s).
\end{aligned}$$

For the upper bound, we have from Assumption 2.16 and the definition of $\check{\mathcal{U}}_N(x,t)$, we have $V_N(z,t) \leq \gamma(\|x\|_x) \leq \gamma(\|z\|)$ for $\|x\|_x \leq \epsilon$. By Assumptions 2.7 and 2.15, we have that sublevel sets of $\|z\|$ are compact, and thus using Corollary 2.29, we can extend this local bound to a global $\mathcal{K}_\infty$ bound $\alpha_2(\,\cdot\,)$ with $V_N(z,t) \leq \alpha_2(\|z\|)$ on $\check{\mathcal{Z}}_N(t)$.

For the decrease condition, we proceed along a similar line to note that $\alpha_1(\|\mathbf{u}\|_u) \leq \alpha_1(\|z\|) \leq V_N(z,t) \leq \gamma(\|x\|_x)$ whenever $\|x\|_x \leq \epsilon$. By Assumption 2.7, we also have that $\check{\mathcal{U}}_N(x,t)$ is compact, and thus $\|\mathbf{u}\|_u \leq M$ for some constant $M > 0$. Thus, defining $\mu_1(s) \coloneqq \alpha_1^{-1}\left(M \min(1, \gamma(s)/\gamma(\epsilon))\right)$, we have that $\|\mathbf{u}\|_u \leq \mu_1(\|x\|_x)$ for all $z \in \check{\mathcal{Z}}_N(t)$. Therefore,

$$\|z\| = \|x\|_x + \|\mathbf{u}\|_u \leq \|x\|_x + \mu_1(\|x\|_x) \leq \mu_2(\|x\|_x), \quad \mu_2(s) \coloneqq s + \mu_1(s),$$

with $\mu_2(\,\cdot\,)$ a $\mathcal{K}_\infty$ function. Thus, $\|x\|_x \geq \mu_2^{-1}(\|z\|)$.

Now, using the same cost decrease argument from Theorem 2.13, we have that

$$V_N(z^+, t+1) - V_N(z,t) \leq -\ell(x, u^*(0), t) \leq -\alpha(\|x\|_x)$$

by applying the lower bound from Assumption 2.8. Substituting the relationship from the previous paragraph, we have

$$V_N(z^+, t+1) - V_N(z,t) \leq \alpha_3(\|z\|), \quad \alpha_3(s) \coloneqq \alpha(\mu_1^{-1}(s)).$$

Therefore, $V_N(\,\cdot\,)$ is a Lyapunov function for (2.15), and asymptotic stability follows from Theorem 2.4. □

The benefit of Theorem 2.18 over Theorem 2.13 is that the optimization requirements of the suboptimal MPC are significantly less than for optimal. Indeed, the requirements of (2.14) is so mild that optimization is only required in two instances: (a) to find an initial warm-start sequence $\mathbf{u}$ for the initial value of x at time $t = 0$; and, (b) to find a low-cost solution near the origin when the restriction of $\check{\mathcal{U}}_N(x,t)$ is active. For solvers such as Gurobi (Gurobi Optimization, LLC, 2018) and BARON (Sahinidis, 2018) that can terminate early and provide an incumbent solution, the entire solution time could be used for optimization, and the best available solution from the solver can be used as $\check{\mathbf{u}}$. By contrast, for local NLP solvers like IPOPT (Wächter and Biegler, 2006), the suboptimal framework ensures that the locally optimal solutions can still be used as long as they beat the warm start.

Remark 2.19: As a generalization of Allan et al. (2017), we have allowed $\|z\| = \|x\|_x + \|\mathbf{u}\|_u$ to be a generalized norm. Aside from the restriction on $\|\cdot\|_x$ in Assumption 2.15, there is a significant amount of flexiliby afforded by these choices. A standard choice would be to choose both generalized norms as the Euclidean norm, which would imply both $x \to 0$ *and* $u \to 0$ as $t \to 0$. However, this choice would impose the additional restriction that the system is stabilizable with small controls (see Proposition 2.11(c)), in particular implying that any discrete actuators are fixed in some neighborhood of the origin. To avoid this property, we can choose instead for $\|\cdot\|_u$ to be the trivial seminorm $\|u\|_u \equiv 0$. Thus, we would have $x \to 0$ as desired, but u could continue to take any value at the origin. This choice could be used to impose soft constraints on x or u that are asymptotically satisfied. Another common case is to reformulate the problem in terms of the increment Δu via the augmented system

$$z^+ = \begin{pmatrix} x^+ \\ u^+ \end{pmatrix} = \begin{pmatrix} f(x, u + \Delta u, t) \\ u + \Delta u \end{pmatrix} \coloneqq f(z, \Delta u, t)$$

and penalize only Δu rather than u directly. Theorem 2.18 can be applied to this system by choosing $\|(\,\cdot\,)\|_x$ as a seminorm $\|z\|_x = |x|$ and $\|\Delta u\|_u = |\Delta u|$. This choice would give $x \to 0$ and $\Delta u \to 0$ asymptotically but not necessarily that $u \to 0$. Finally, we note that $\|\cdot\|_x$ and/or $\|\cdot\|_u$ can be taken as point-to-set distances using any compact sets X and U.

2.3 Discrete Actuators

In the previous section, we made no explicit mention of any actuators being discrete. The reader may be wondering when explicit treatment of discrete will make an appearance. However, the MPC problem as formulated in the previous section (both optimal and suboptimal) is already compatible with discrete valued actuators. In this section, we review some previous methods from the literature for considering discrete actuators, and then we explain how they are accounted for in our formulation.

2.3.1 Literature Review

As mentioned in Section 2.1.1, it is common in the literature to consider any system with discrete-valued inputs or states as "hybrid." However, most of the analysis is carried out in discrete time, and thus, the system is not truly hybrid in the sense of Goebel et al. (2012). Regardless of the chosen vocabulary, stability results have previously appeared for various special classes of systems that consider discrete actuators. For piecewise affine (PWA) systems, Baotić et al. (2006) establishes asymptotic stability for an infinite-horizon problem using a Lyapunov function. Although the actuator space for PWA systems does not explicitly include discreteness restrictions, their effect can be included by adding a separate subdomain for each possible combination of discrete inputs and using a continuous actuator to select from the pieces. For example, the system $x^+ = x + u$ for $u \in \{0, 1\}$ is modeled as

$$x^+ = \begin{cases} x & u \in [0, 0.5], \\ x + 1 & u \in (0.5, 1]. \end{cases}$$

Of course, this approach suffers the curse of dimensionality when multiple discrete actuators are present. To treat discrete variables directly, the mixed-logical-dynamical (MLD) formulation can be used, for which convergence to the origin was demonstrated under positive-definiteness of the objective function. It can be shown that the class of PWA and MLD systems are identical (Camacho et al., 2010), i.e., that an instance of one can be converted to an instance of the other. For nonlinear switched systems, Di Cairano et al. (2014) propose embedding a Lyapunov equation directly into the problem formulation, thereby requiring cost decrease as a hard constraint.

For systems in which the actuators are *quantized*, i.e., finely discretized with a sensible relaxation for intermediate values, discreteness can be regarded as a small disturbance of a continuous-actuator system. In this vein, Quevedo et al. (2004) proposes a rounding procedure for a linear system with a quantized actuator that provides bounds away from the optimal continuous solution. Similar results show asymptotic stability for stable systems and practical stability (i.e., ultimate boundedness) for unstable systems (Aguilera and Quevedo, 2013; Kobayshi et al., 2014; Picasso et al., 2003). For continuous-time problems, Sager et al. (2009) and Sager et al. (2010) proposes a convexification and iterative rounding-based refinement procedure to address discrete actuators, although the application to discrete-time systems is limited. Finally, Thomas (2012) proposes a direct optimization strategies for *finite* $\mathbb{U}$ by enumerating and computing the cost of feasible input sequences (with some reduction techniques to avoid having to check all possible combinations). However, with the increasing capabilities of mixed-integer optimization techniques, it is increasingly possible to optimize discrete variables directly, and thus we which to derive a general theory.

2.3.2 Compatibility of Discrete Actuators

Due to the computational difficulties associated with mixed-integer optimization, much of the early MPC literature considered exclusively the case of continuous-valued actuators (García et al., 1989; Mayne et al., 2000; Rawlings and

Mayne, 2009). This restriction leads to assumptions suited to continuous actuators, e.g., from Mayne et al. (2000), "[U]sually, $\mathbb{U}$ is a convex, compact subset of $\mathbb{R}^m$, and $\mathbb{X}$ a convex, closed subset of $\mathbb{R}^n$, each set containing the origin in its interior." Unfortunately, the inclusion of any discrete-valued components in u is incompatible with these requirements, most notably the fact that the setpoint must be on the interior of $\mathbb{U}$. Indeed, this very assumption is pervasive throughout early MPC literature (Chmielewski and Manousiouthakis, 1996; Keerthi and Gilbert, 1988; Rawlings and Muske, 1993; Scokaert and Rawlings, 1996; Sznaier and Damborg, 1987) to ensure local stabilizability of the setpoint. However, as MPC allows direct handling of process constraints, it is natural to desire that the system operate not just close to but precisely at one or more constraints.

As first recognized in (Rao and Rawlings, 1999), active constraints at the setpoint can be addressed by defining a terminal control law that moves in the null space of active constraints, thus not requiring use of saturated actuators. As a result, the assumption of the setpoint being on the interior of $\mathbb{X}$ or $\mathbb{U}$ is no longer necessary. The inclusion of discrete actuators is essentially a logical extension of this idea: it is perfectly acceptable to have active constraints throughout the feasible space, and discrete actuators are simply constraints that are always active. This idea leads to the development of the following result:

THEOREM 2.20 (Folk theorem): *Any result that holds for standard MPC holds also for MPC with discrete actuators. (Rawlings and Risbeck, 2017)*

While referring to this result as a theorem is somewhat tongue-in-cheek, it holds for all of the results in this and the following chapter. We give a justification for the folk theorem in Figure 2.3. Historically, $\mathbb{U}$ was required to have the geometry of (a), so that both u_1 and u_2 could be used for local stabilization. These requirements were later relaxed to (b), in which u_2 saturates, but u_1 is still available. Allowing discrete-valued actuators then results in the geometry of (c), which is locally not much different from the previous case. Thus, provided that restrictive assumptions on $\mathbb{U}$ are not made, modern MPC results are agnostic to whether $\mathbb{U}$ is convex, disconnected, discrete, etc. Consider, for example Assumption 2.7 from this chapter, which requires only that, "the set $\mathbb{U}(t)$ is compact." We are therefore free to include discreteness restrictions in the set $\mathbb{U}$, and the system can still be rendered asymptotically stable by MPC. Note, however, that this result does not mean that every system can be controlled with discrete actuators. Theorem 2.13 states that MPC is stabilizing on the set $\mathcal{X}_N(t)$, i.e., the set of states where the MPC problem is feasible. If too many discreteness restrictions are imposed on the problem, then the geometry of $\mathcal{X}_N(t)$, perhaps consisting of disconected, isolated, or otherwise nonconvex regions. Thus, it is important to recognize that there may be inherent limitations in what can be achieved with discrete actuators, and before applying tracking MPC, it is important to decide whether setpoint stabilization is a worthwhile goal.

REMARK 2.21: Throughout this discussion, we have made reference to only the actuators, i.e., inputs u, being discrete. This may raise the question of whether it is possible to consider discreteness restrictions on system states. The short

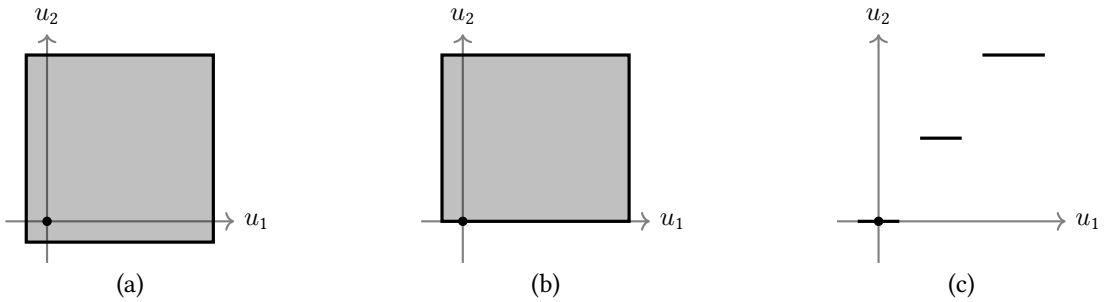


FIGURE 2.3: Illustration of discrete actuators as permanently active consraints. Figures show the geometry of $\mathbb{U}$ for continuous actuators with no active constraints at the setpoint (a), continuous actuators with constraints active at the setpoint (b), and mixed-integer actuators (c). The setpoint is indicated by a circular marker.

answer is yes; discreteness restrictions can be embedded in $\mathbb{X}$ just as they can be in $\mathbb{U}$. In fact, in some instances, these restrictions can make certain mathematical developments easier. For example, if the set $\mathbb{X}$ is *finite*, then the rather technical Assumption 2.10 is satisfied automatically, as all points are isolated, and thus $V^*(\,\cdot\,)$ is trivially continuous.

However, we focus more directly on discrete-valued inputs for two main reasons. First, when establishing closed-loop properties, we are generally concerned with limiting behavior of x, which has a natural connection to vanishingly small quantities. In the case of discreteness restrictions in $\mathbb{X}$, certain statements become vacuous, and other considerations (e.g., small disturbances affecting the system) become meaningless. Second, it is almost always possible to render states *implicitly discrete* by appropriate discreteness restrictions on u. For example, to consider the system $x^+ = x + u$ with the constraints $\mathbb{X} = \mathbb{I}$ (i.e., x is constrained to be integer-valued) and $\mathbb{U} = [-1, 1]$, one can instead swap the discreteness restrictions to $\mathbb{U}$, giving $\mathbb{U} = \{-1, 0, 1\}$ and $\mathbb{X} = \mathbb{R}$. With a terminal constraint $\mathbb{X}_f \subseteq \mathbb{I}$, the feasible space $\mathcal{X}_N$ contains only discrete-valued x (and of course is positive invariant). Thus, considering only discrete actuators is sufficient for our purposes, but it is not a hard requirement.

2.4 Illustrative Examples

In this section, we present two examples to demonstrate some of the behavior of mixed-integer tracking MPC. We will see, in some cases, the optimizer take unexpected control action in order to achieve its goals, which illustrates the benefits that can be achieved via optimizing discrete actuators directly, rather than using heuristics or decompositions.

2.4.1 Feasible Sets

As a first example, we consider a two-state system and compare the feasible sets $\mathcal{X}_N$ when there are discrete variables and when there are not. The system consists of two coupled zones that must be cooled by two chillers, each of which can be either on or off and are subject to minimum capacities when on. The system model is linear as follows:

$$\begin{pmatrix} x_1^+ \\ x_2^+ \end{pmatrix} = \begin{pmatrix} 0.9197 & 0.0549 \\ 0.1098 & 0.8428 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -0.4793 & -0.0287 \\ -0.0287 & -0.9186 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix},$$

in which x_1 and x_2 are the zone temperatures, and u_1 and u_2 are the cooling sent to each zone, the sum of which is bounded by the minimum and maximum capacities of active chillers. The system is in deviation variables, and the goal is to stabilize the origin. When the minimum capacity is nonzero, the input set $\mathbb{U}$ becomes disconnected.

To examine the feasible sets, we note that in this case A is invertible, and thus we have the relationship

$$\mathcal{X}_N = A^{-1} \left(\mathcal{X}_{N-1} \oplus (-B\mathbb{U}) \right).$$

Because $\mathbb{U}$ is a union of (convex) polyhedral sets, we can use their extreme point representations together with this update formula, proceeding backwards from $\mathbb{X}_f$. For illustration purposes, we choose $\mathbb{X}_f$ as a polyhedral approximation of the set $\{x \in \mathbb{R}^2 : x^\mathsf{T} P x \leq 0.1\}$ with P satisfying the Lyapunov equation $P = A^\mathsf{T} P A + I$. Because A is stable, we could choose an arbitrarily large terminal region, but we choose this small region for the purpose of visualization.

Figure 2.4 shows the input and feasible sets for this system for varying minimum capacities. For case of zero minimum capacity (first row), $\mathbb{U}$ is convex, and so convexity of $\mathbb{X}_f$ leads to convexity of each $\mathcal{X}_N$. By contrast, when there are nonzero minimum capacities (which requires discrete variables for optimization), the set $\mathbb{U}$ is nonconvex and disconnected, which leads to similar properties for $\mathcal{X}_N$. In the second row, the minimum capacity is 90% of the maximum capacity, and thus there is still some degree of continuous actuation. Therefore, the feasible sets are smaller, but not significantly so. In the bottom row, the minimum capacity is exactly equal to the maximum capacity, and so total cooling is quantized. For this case, we see even narrower feasible sets for small N, but as N increases, the feasible region loses most of its gaps. However, we note that in all cases, the sets $\mathcal{X}_N$ are nested (i.e, $\mathcal{X}_{N-1} \subseteq \mathcal{X}_N$), and as the horizon increases, the difference between the continuous and discrete cases becomes smaller. Thus, when the nonzero minimum capacities are added, short horizons may no longer be adequate near the origin, but for sufficiently long horizons, the feasible sets are almost as large as in the zero-minimum-capacity case.

2.4.2 Switched System

As the next example, we consider a simple linear switched system. The system model is

$$x^+ = A(u_1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + B(u_1) \begin{pmatrix} u_2 \\ u_3 \end{pmatrix},$$

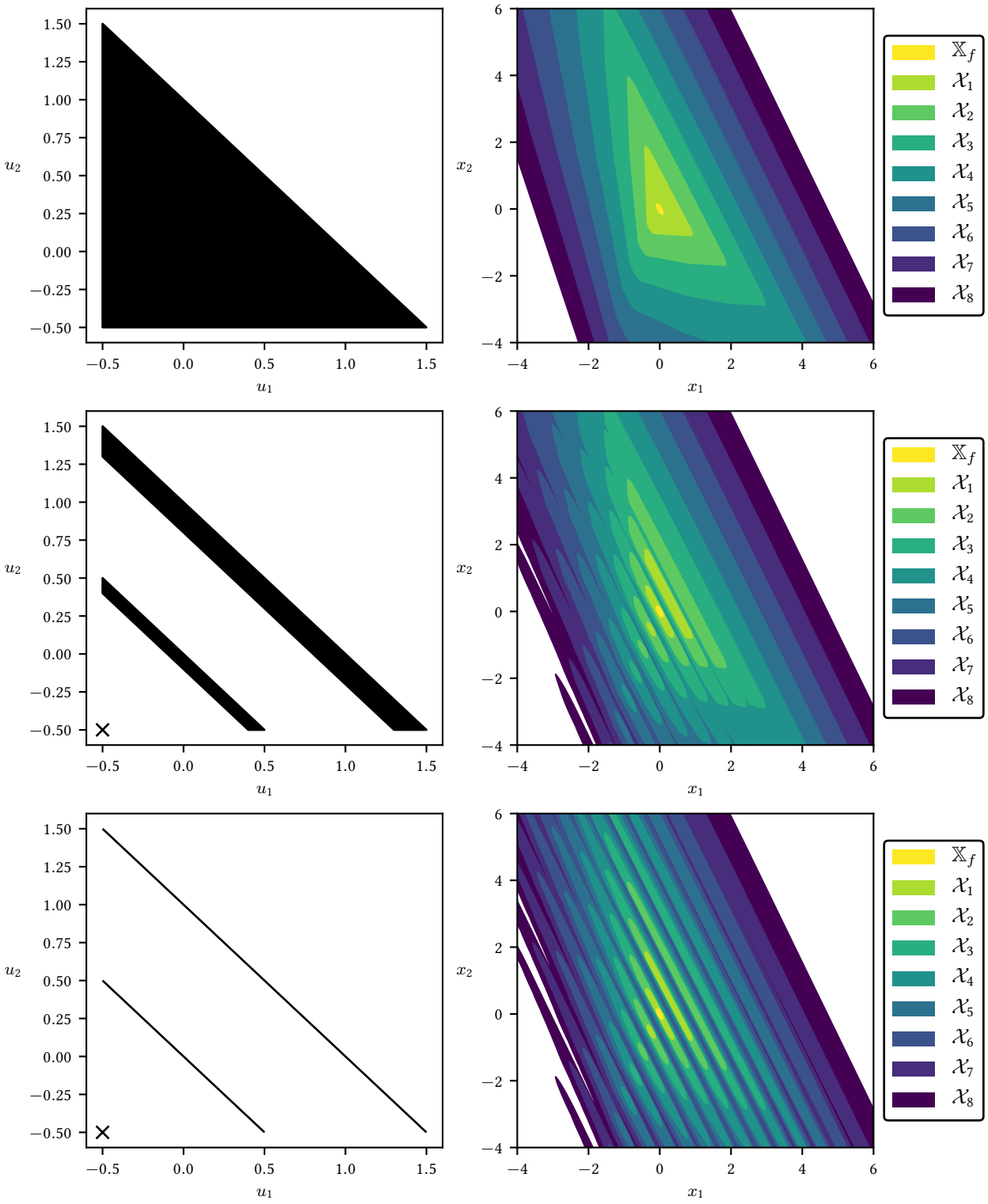


FIGURE 2.4: Feasible sets for two-state example with varying discreteness restrictions. Sets $\mathbb{U}$ are shown in the first column, with isolated points indicated by $\times$ markers. Sets $\mathcal{X}_N$ for varying N are shown in the second column.

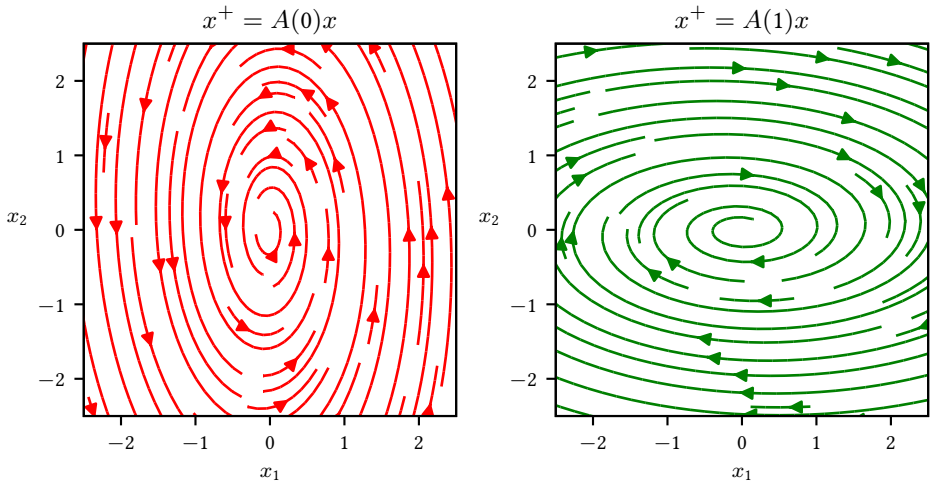


FIGURE 2.5: Unforced dynamics for switched system example. When switching from $u_1 = 0$ to $u_1 = 1$, the direction of rotation changes, as does the stretching along each axis.

TABLE 2.1: Values of matrices for switched system example.

i	$A(i)$	$B(1)$
0	$\begin{pmatrix} 0.9799 & -0.0616 \\ 0.2463 & 0.9799 \end{pmatrix}$	$\begin{pmatrix} 0.0248 & -0.0008 \\ 0.0031 & 0.0248 \end{pmatrix}$
1	$\begin{pmatrix} 0.9799 & 0.2463 \\ -0.0616 & 0.9799 \end{pmatrix}$	$\begin{pmatrix} 0.0248 & 0.0031 \\ -0.0008 & 0.0248 \end{pmatrix}$

with $u_1 \in \{0, 1\}$ to choose between to possible sets of linear dynamics. Figure 2.5 shows the (unforced) evolution of x for $u_1 = 0$ and $u_1 = 1$. The remaining inputs $(u_1, u_2) \in [-1, 1]^2$ are provided for to give additional degrees of freedom, although the gain is small (see values in Table 2.1). The system states are restricted to the box $(x_1, x_2) \in [-2.5, 2.5]^2$. Choosing a feasible setpoint $x^{\mathrm{sp}}(t)$, $u^{\mathrm{sp}}(t)$, the stage cost is taken as

$$\ell(x, u, t) = |x - x^{\mathrm{sp}}(t)|_Q^2 + |u - u^{\mathrm{sp}}(t)|_R^2$$

with $Q = I$ and $R = \mathrm{diag}(0, 1, 1)$ so that the discrete actuator is not penalized, but the two continuous actuators are. We note that, following Remark 2.19, this cost function ensures that $x \to 0$ and $u_2, u_3 \to 0$ as $t \to \infty$, but not necessarily that $u_1 \to 0$. For a terminal control law, we note that the matrices $A(0)$ and $A(1)$ are both stable, and thus a feasible (though highly suboptimal) control law is $u = (1, 0, 0)$ with terminal cost $V_f(x, t) = |x - x^{\mathrm{sp}}(t)|_P^2$ with P satisfying the Lyapunov equation $A(1)^\top P A(1) + Q = P$. As a terminal regions $\mathbb{X}_f(t)$, we can take any sublevel set of $V_f(\,\cdot\,)$ that fits within $\mathbb{X}(t)$, although in this example, the terminal constraint is not binding for a suitably large $\mathbb{X}_f$ and horizon N.

To start, we consider a steady-state setpoint $x^{\mathrm{sp}}(t) \equiv 0$, $u^{\mathrm{sp}}(t) \equiv 0$. Note that at $x = 0$, the value of u_1 does not matter. For three initial conditions, we simulate closed-loop MPC. Note that because the optimization is difficult to

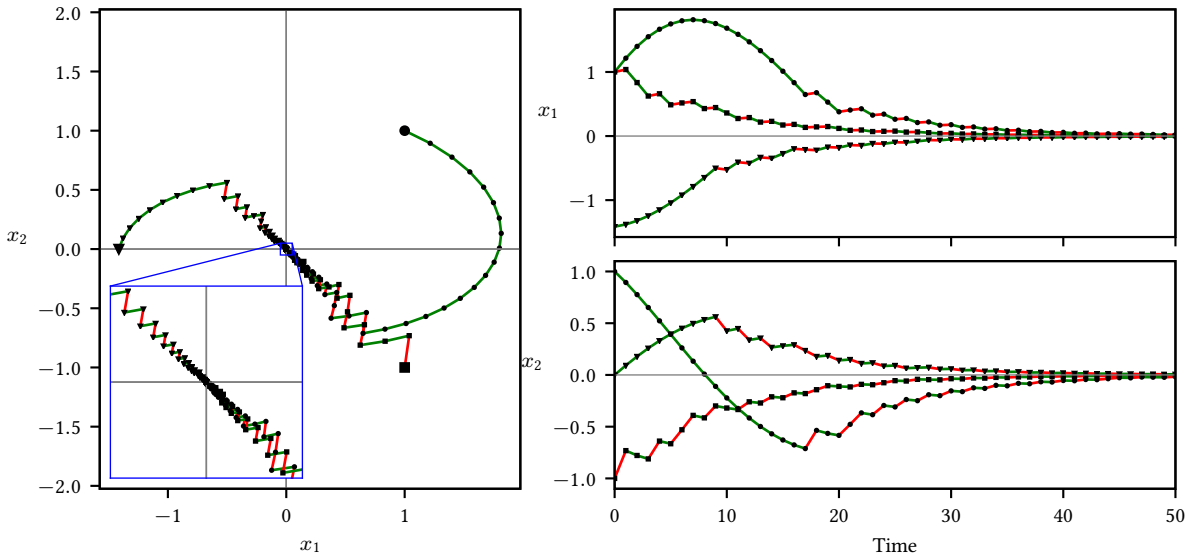


FIGURE 2.6: Closed-loop trajectory for switched system with $x^{\text{sp}}(t) \equiv 0$. Red segments indicate $u_1 = 0$, while green segments indicate $u_1 = 1$.

solve to optimality, we make use of suboptimal MPC and give the solver a time limit of 10 s per step, and we only optimize every tenth time point. A plot of closed-loop behavior is shown in Figure 2.6 These trajectories illustrate the fact that the system can approach the origin along the line $x_1 = -x_2$ by switching back and forth from $u_1 = 0$ to $u_1 = 1$. Thus, the optimal trajectory is to choose one value of u_1 to get to the critical line as quickly as possible and then cycle between the two. However, u_1 does not necessarily switch every period, even close to the origin. Indeed, when the constraints on x and u are inactive, the problem is scale-invariant (that is, the optimal switching sequence for an initial condition x is the same as for initial condition ρx for scalar ρ). Thus, the optimal solution illustrates fractal behavior near the origin, and switching continues indefinitely as the system approaches the origin. Thus, even though the discrete actuator is asymptotically constant, there is still significant benefit optimizing it directly.

To illustrate the time-varying capabilities of MIMPC, we use a time-varying setpoint equal to the non-decaying orbit of the system under $A(1)$. Note that we wish to stabilize the system *in phase* with the setpoint, and that the setpoint is *not* periodic (the trajectory is homeomorphic to irrational rotation). As before, since $A(1)$ is stable, so we use $V_f(x,t) = |x - x^{\text{sp}}(t)|_P^2$ and $\kappa_f(x,t) = \{u^{\text{sp}}(t)\}$ once more. $\mathbb{X}_f(t)$ can still be taken as sublevel sets of $V_f(\,\cdot\,)$ except that they must be shrunk due to the setpoint being closer to the state constraints. We illustrate one possible choice for $\mathbb{X}_f(t)$ in Figure 2.7. Starting from an initial condition $x(0) = (-2, 0)$, which is well away from the setpoint $x^{\text{sp}}(0) = (2, 0)$, closed-loop MPC is simulated using the same strategy as before. The resulting closed-loop trajectory is shown in Figure 2.8. From this trajectory, we see that MPC does stabilize the time-varying trajectory. Interestingly, rather than try to move directly toward the setpoint, the optimal solution is actually to wait for the setpoint to move

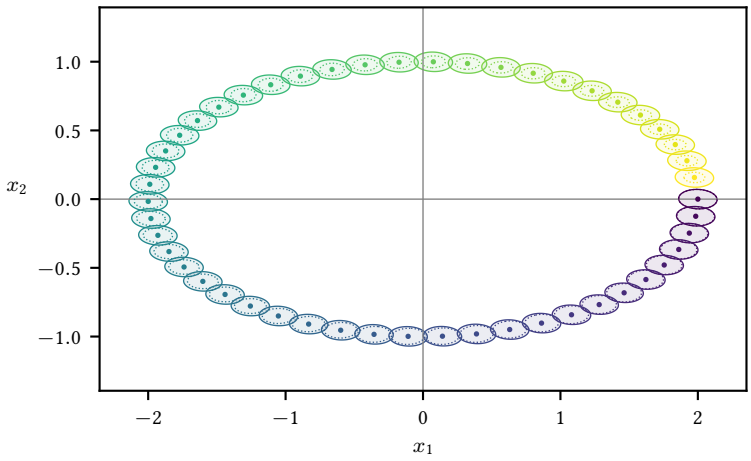


FIGURE 2.7: Time-varying terminal sets for switched system example. Shaded regions show $\mathbb{X}_f(t)$, points show $x^{\text{sp}}(t)$, while dashed lines show evolution of $\mathbb{X}_f(0)$ under the terminal control law $\kappa_f(\,\cdot\,)$. Note that the terminal regions can be taken larger than shown.

towards the initial condition. The optimizer exploits this capability by using the discrete actuator to remain near the left edge of $\mathbb{X}$, after which it meets up with and follows $x^{\text{sp}}(t)$. After 100 steps, $|x - x^{\text{sp}}|$ is on the order of 0.004; were the system forced to use $u_1 = 1$ constantly, it would require nearly 500 steps to achieve the same level of convergence. Therefore, direct optimization of the discrete actuator u_1 leads to significantly improved performance even though it is asymptotically constant.

Note that for simulation, this problem is formulated using CasADi (Andersson et al., 2018) via MPCTools (Risbeck and Rawlings, 2018a). Optimizations are performed using Gurobi (Gurobi Optimization, LLC, 2018).

2.4.3 Nonlinear Batch Production

In this example, we consider an example of nonlinear batch production first presented in Rawlings and Risbeck (2015). The goal of the system is to deliver a liquid product with concentration limits on a specific component. To produce this product, two parallel batch reactors are operated. In each time interval, the batch reactors can be used or not used, and if they are used, the total volume and component concentration can vary between (nonzero) minimum and maximum levels. The output of each batch reactor is sent into a large mixing tank, and then a fixed amount of product is removed; while the individual batches do not obey the concentration limits, the final product withdrawn from the tank must satisfy constraints. The system is diagrammed in Figure 2.9. Note that the problem is time-invariant, so we omit time indices throughout. We wish to apply mixed-integer tracking MPC to this system.

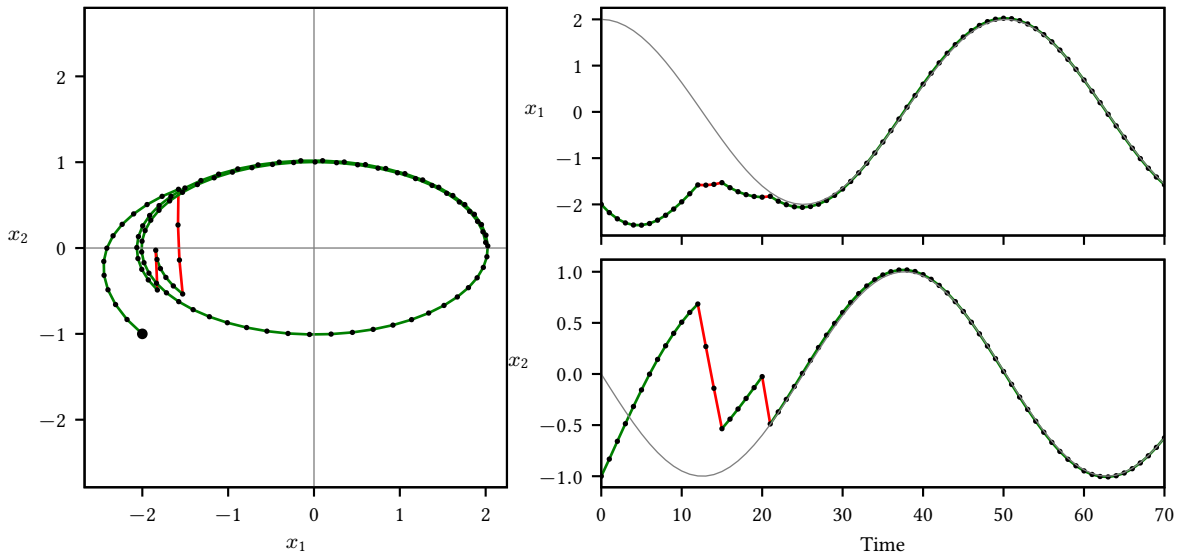


FIGURE 2.8: Closed-loop trajectory for switched system with time-varying $x^{\text{sp}}(t)$. Gray curve in the timeseries plots show setpoint.

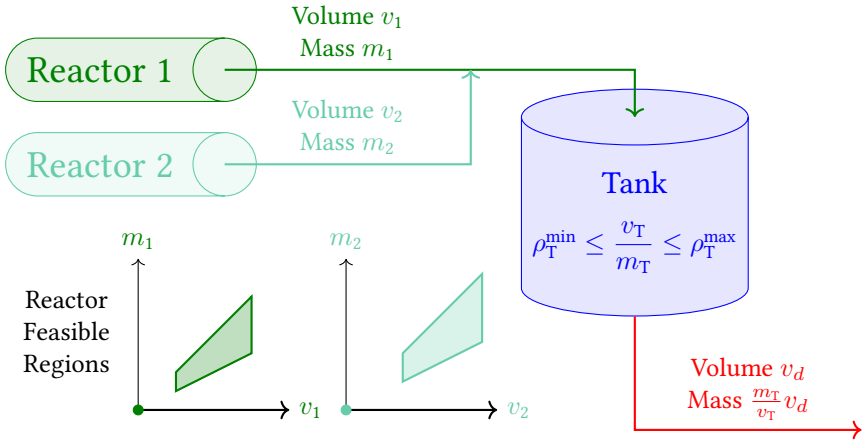


FIGURE 2.9: Diagram of nonlinear batch production example. Discrete actuators decide whether or not to operate each reactor, and continuous actuators choose batch sizes and concentrations.

To model this system, we track the tank concentration via its total volume v_T and the mass m_T of specific component. Similar variables v_i and m_i are used for reactor batches, with additional discrete variables $z_i \in \{0,1\}$ to determine utilization (all for $i \in \{1,2\}$). The system model is as follows:

$$\begin{aligned} v_\mathrm{T}^+ &= v_\mathrm{T} + v_1 + v_2 - v_d \\ m_\mathrm{T}^+ &= (m_\mathrm{T} + m_1 + m_2)\left(1 - \frac{v_d}{v_\mathrm{T} + v_1 + v_2}\right) \end{aligned}$$

Note that v_d is a fixed parameter that represents the volume withdrawn from the tank at the end of each period.. The system states $x \coloneqq (v_\mathrm{T}, m_\mathrm{T})$ are required to satisfy

$$v_\mathrm{T}^{\min} \le v_\mathrm{T} \le v_\mathrm{T}^{\max}, \qquad \rho_\mathrm{T}^{\min} v_\mathrm{T} \le m_\mathrm{T} \le \rho_\mathrm{T}^{\max} v_\mathrm{T},$$

which enforces both volume and concentration limits for the tank. Similarly, the system inputs $u \coloneqq (z_1, v_1, m_1, z_2, v_2, m_2)$ must satisfy

$$z_i \in \{0,1\}, \quad v_i^{\min} z_i \le v_i \le v_i^{\max} z_i, \quad \rho_i^{\min} v_i \le m_i \le \rho_i^{\max} v_i \tag{2.16}$$

for $i \in \{1,2\}$. Here, $v_i^{\min}$ and $v_i^{\max}$ give the minimum and maximum batch sizes, while $\rho_i^{\min}$ and $\rho_2^{\max}$ are the minimum and maximum concentrations that can be achieved by each reactor. Note that the second constraint is needed so that volume satisfies $v_i = 0$ when the reactor is off ($z_i = 0$) and $v_i^{\min} \le v_i \le v_i^{\max}$ when the reactor is on ($v_i = 1$).

Assume that the desired steady state

$$x^{\mathrm{ss}} \coloneqq (v_\mathrm{T}^{\mathrm{ss}}, m_\mathrm{T}^{\mathrm{ss}}), \qquad u^{\mathrm{ss}} \coloneqq (z_1^{\mathrm{ss}}, v_1^{\mathrm{ss}}, m_1^{\mathrm{ss}}, z_2^{\mathrm{ss}}, v_2^{\mathrm{ss}}, m_2^{\mathrm{ss}})$$

is given. As a tracking cost, we take the standard quadratic $\ell(x,u) = |x - x^{\mathrm{ss}}|_Q^2 + |u - u^{\mathrm{ss}}|_R^2$ with $Q = \mathrm{diag}(1,1)$ and $R = \mathrm{diag}(0, 0.25, 0.5, 0, 0.25, 0.5)$. Numerical values of constraints are shown in Figure 2.10.

As suggested in Rawlings and Risbeck (2015), we choose a terminal control law by linearizing the model and determining a local linear quadratic regulator (LQR) near the steady-state operating point. Although the system has six inputs, only two are available at steady state: the discrete inputs z_1 and z_2 are fixed, v_1 is at its lower bound, and m_2 is at its upper bound (which depends on v_2). Thus, we follow the suggestion of Rao and Rawlings (1999) and use only the reduced system of unsaturated inputs: take m_1 and v_2 as free inputs, assume $m_2 = \rho_2^{\max} v_2$ to stay in the null space of the active constraint, and fix all other inputs to their steady-state values, giving a reduced linear system with two states and two inputs. This procedure yields a linear control law $u = \kappa_f(x) \coloneqq K(x - x^{\mathrm{ss}}) + u^{\mathrm{ss}}$. A candidate terminal set is then

$$\mathbb{X}_f = \{x \in \mathbb{X} \mid \kappa_f(x) \in \mathbb{U}, f(x, \kappa_f(x)) \in \mathbb{X}\}$$

We show $\mathbb{X}_f$ and the active constraints along each boundary segment in Figure 2.11. After verifying that $\mathbb{X}_f$ is invariant under $\kappa_f(\,\cdot\,)$ (as illustrated in Figure 2.11), the terminal set $\mathbb{X}_f$ is defined.

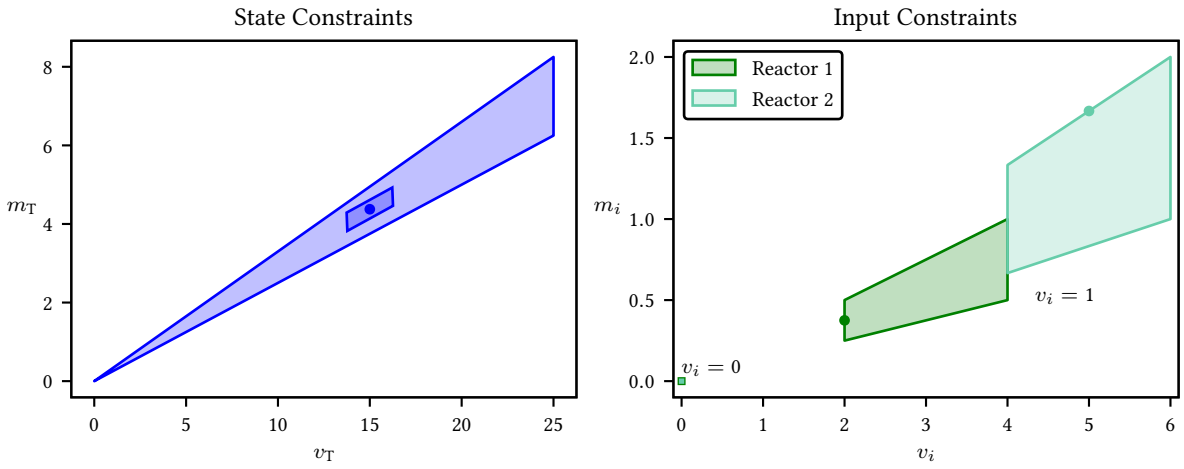


FIGURE 2.10: Feasible state and input space for the nonlinear batch production example. Circular markers indicate steady-state values. The terminal region $\mathbb{X}_f$ is also shown in the feasible state diagram.

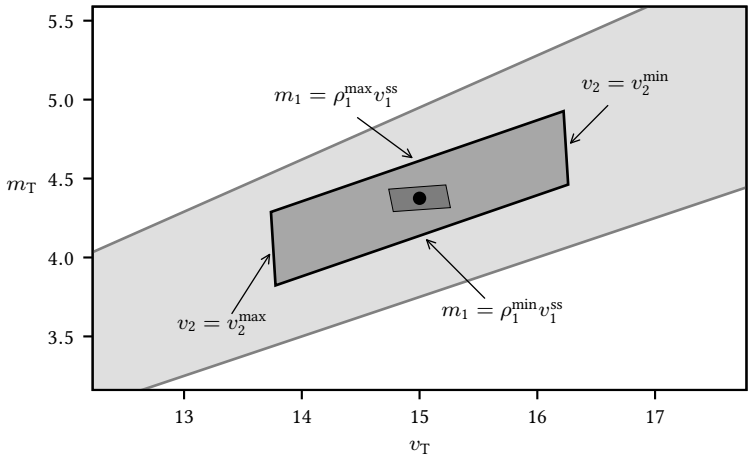


FIGURE 2.11: Terminal set $\mathbb{X}_f$ for the LQR local controller for the batch production example. Along each boundary segment, the active constraint is indicated. x^{ss} is indicated by the circular marker. The innermost black region is the image of $\mathbb{X}_f$ under one step of LQR control. The feasible set is also shown in gray.

TABLE 2.2: Relaxation and restoration methods for suboptimal solution. All methods use horizon $N = 10$.

Method	Relaxation and Restoration Steps
Warm-Start	Use warm-start without any further optimization.
All On	Fix all binary variables to 1.
Sum Error	Relax binary variables to continuous; then, use a sum-up rounding procedure (Sager et al., 2010) to fix binary variables and re-optimize continuous variables.
Round	Relax binary variables to continuous; then, round $v_i \in (0, v_i^{\min})$ to 0 or $v_i^{\min}$ and adjust m_i appropriately

To illustrate Theorems 2.13 and 2.18, we simulate the closed-loop evolution of the system under both optimal and suboptimal MPC. For optimal MPC, each MINLP problem is solved to optimality using BARON (Sahinidis, 2018). For suboptimal MPC, we start from a feasible (but highly suboptimal solution) $\mathbf{u}$ also obtained from BARON. The "optimization" step of choosing $\check{\mathbf{u}} \in \check{\mathcal{U}}_N^*(x, \mathbf{u})$ is then as follows:

- Solve a continuous relaxation of the control problem using a local NLP solver
- Restore feasibility by adjusting the solution, in particular ensuring the z_i are binary
- If this rounded solution is better, use it; otherwise, use the warm-start

The successor warm-start $\mathbf{u}^+$ is then obtained by applying the the linear terminal control law $\kappa_f(\,\cdot\,)$. For the relaxation and restoration steps, multiple methods are employed as summarized in Table 2.2. Closed-loop solutions for optimal and suboptimal MPC are shown in Figure 2.12. From this example, we see that optimal MPC proceeds directly to the setpoint, while suboptimal MPC takes a roundabout path but eventually stabilizes.

To compare optimal MPC to the other suboptimal methods, we show a phase portrait of closed-loop evolution in Figure 2.13. As expected, the system is stable regardless of which suboptimal strategy is used, although the closed-loop trajectories can be quite different. For example, while the optimal controller proceeds directly to the setpoint, the Warm-Start and Sum Error methods take longer transients before finally converging; by contrast, the Round method is quite close to optimal, indicating that rounding is a suitable heuristic method for this particular system. Thus, while any suboptimal algorithm (satisfying (2.14)) will stabilize the system, performing some form of direct optimization often yields better performance.

2.5 SUMMARY

In this chapter, we have presented a mathematical formulation for time-varying tracking MPC. Under generalized but otherwise standard assumptions, we have shown that optimal and suboptimal MPC are both stabilizing on the feasible set $\mathcal{X}_N(t)$. In this formulation, discreteness restrictions on u do not need to be explicitly handled, and thus these

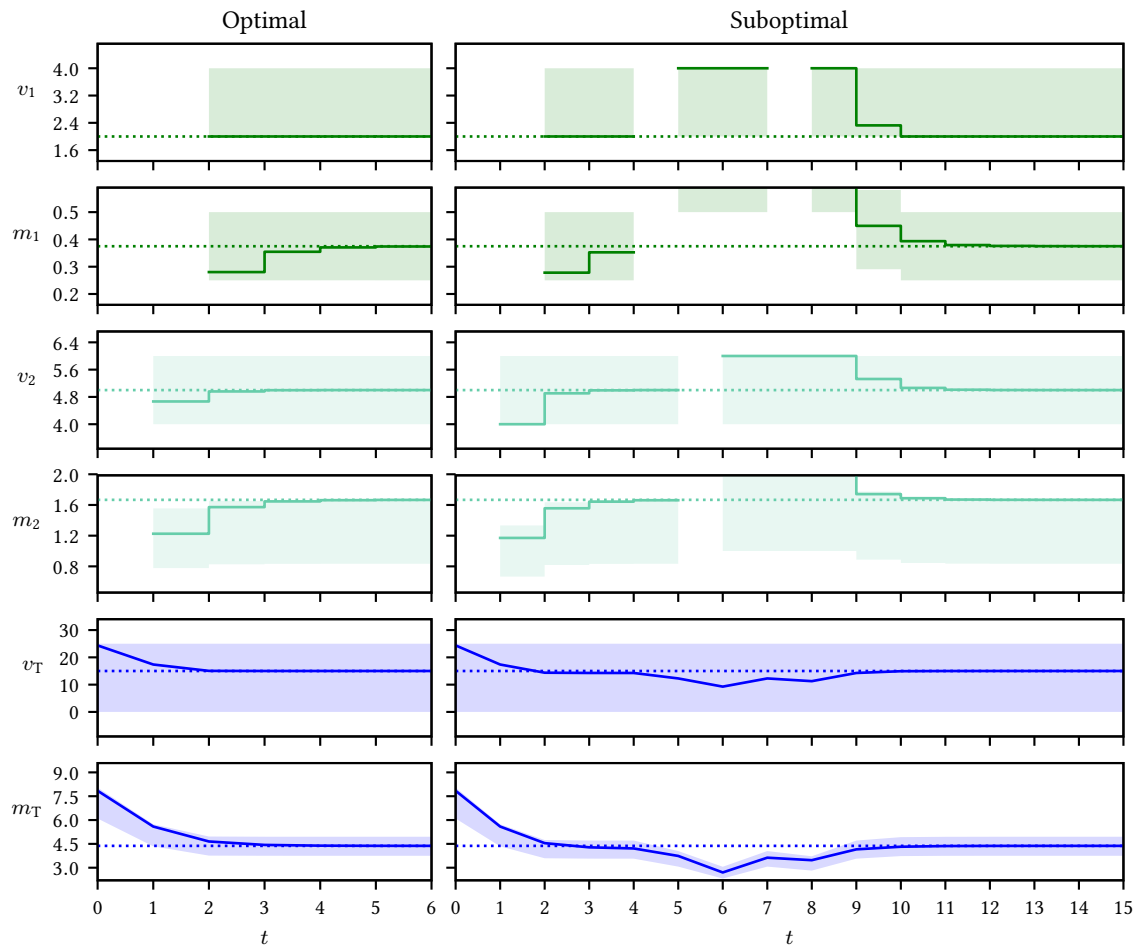


FIGURE 2.12: Closed-loop trajectories for the batch scheduling example using optimal and suboptimal MPC. Shaded regions show feasible space, while dashed lines show the setpoint. Note that the suboptimal trajectory is for the Sum Error method.

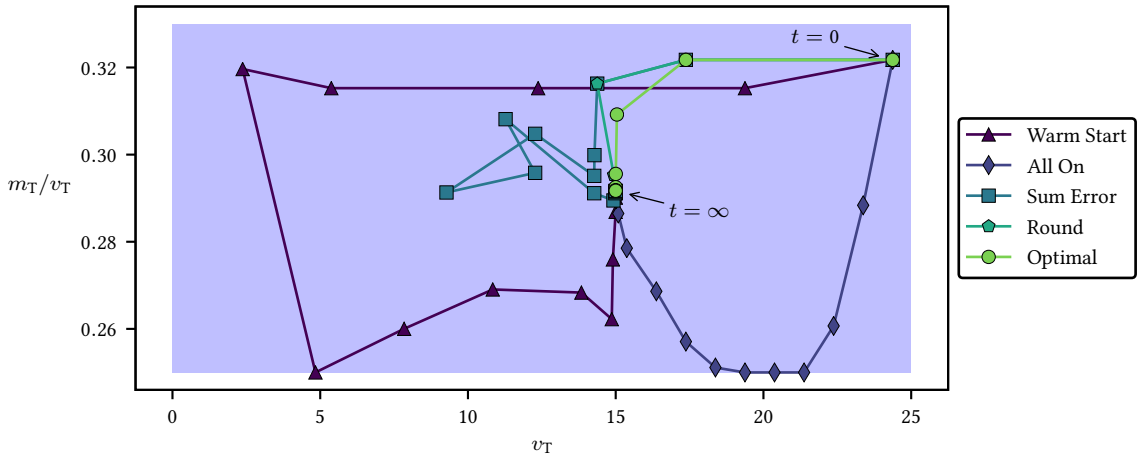


FIGURE 2.13: Closed-loop phase plot for the batch production example. Shaded region shows feasible space. Note that the y-axis is tank concentration m_T/v_T for easier visualization.

results are a natural extension of modern MPC theory. Through examples, we have also illustrated the benefits of allowing discrete inputs to be optimized directly, rather than trying to apply heuristic methods. Via liberal application of suboptimal MPC, we need not find globally optimal solutions to any MPC problem, and thus computational requirements are readily satisfiable using standard hardware.

Extensions of the ideas in this chapter to other settings are presented in Chapter 3. Specific applications of these ideas are shown in the remaining chapters. For additional developments for tracking MPC with discrete actuators, see Rawlings and Risbeck (2017) and Allan et al. (2017).

2.6 Appendix: Lemmas and Proofs

2.6.1 Cost to Go

Lemma 2.22 (Cost to go bound): *Suppose Assumption 2.9 is satisfied. Then, for any $t, T \in \mathbb{T}$ and $x \in \mathbb{X}_f(t)$, there exists a sequence $\mathbf{u} \coloneqq (u(t), u(t+1), \dots, u(t+T-1)) \in \mathcal{U}_T(x,t)$ such that $V_T(x, \mathbf{u}, t) \leq V_f(x,t)$.*

Proof. We prove by induction. For the base case $T = 0$, we have $V_T(x, \mathbf{u}, t) = V_f(x,t)$, and so the bound holds trivially. Now suppose the bound holds for $T \in \mathbb{T}$; we show that it also holds for $T+1$. Choose any $u \in \kappa_f(x,t)$ and define $x^+ \coloneqq f(x,u,t)$. We have $x^+ \in \mathbb{X}_f(t+1)$, and thus by the inductive hypothesis, there exists a sequence $\mathbf{u}^+ \in \mathcal{U}_T(x^+, t+1)$ satisfying

$$V_T(x^+, \mathbf{u}^+, t+1) \leq V_f(x^+, t+1).$$

Now, let $\mathbf{u}$ be the sequence formed by prepending u to $\mathbf{u}^+$. We have $\mathbf{u} \in \mathcal{U}_{T+1}(x,t)$, and

$$\begin{aligned} V_{T+1}(x, \mathbf{u}, t) &= \ell(x,u,t) + V_T(x^+, \mathbf{u}^+, t) \\ &\leq \ell(x,u,t) + V_f(x^+, t+1) \\ &\leq V_f(x,t) \end{aligned}$$

in which we have applied the cost decrease condition from Assumption 2.9. Thus, the inductive hypothesis holds for $T+1$, and the lemma is proved. □

Corollary 2.23 (Optimal cost bound): *Suppose Assumption 2.9 is satisfied. For all $t, T \in \mathbb{T}$ and $x \in \mathbb{X}_f(t)$, the optimal cost $V^*(\,\cdot\,)$ satisfies $V^*(x,t) \leq V_f(x,t)$.*

Proof. From Lemma 2.22, there exists a feasible sequence $\mathbf{u} \in \mathcal{U}_T(x,t)$ whose cost is less than $V_f(x,t)$; thus, the optimal cost is less than $V_f(x,t)$. □

2.6.2 $\mathcal{K}$ Functions and Continuity

The following definitions and theorems are adapted from Rawlings and Risbeck (2015).

Definition 2.24 (Property $\mathcal{P}_\delta$): A system with testable condition $C : X \to \mathbb{R}_{\geq 0}$ with $C(0) = 0$ is said to have property $\mathcal{P}_\delta$ if for every $\epsilon > 0$ there exists $\delta(\epsilon) > 0$ such that $C(x) \leq \epsilon$ for every $x \in X$ satisfying $|x| \leq \delta(\epsilon)$.

Definition 2.25 (Property $\mathcal{P}_\mathcal{K}$): A system with testable condition $C : X \to \mathbb{R}_{\geq 0}$ with $C(0) = 0$ said to have property $\mathcal{P}_\mathcal{K}$ if there exists finite $b > 0$ and $\mathcal{K}$-function $\gamma(\,\cdot\,)$ defined on $[0, b]$, such that $C(x) \leq \gamma(|x|)$ for every $x \in X$ satisfying $|x| \leq b$.

Proposition 2.26 (Equivalence of $\mathcal{P}_\delta$ and $\mathcal{P}_\mathcal{K}$): *A system has property $\mathcal{P}_\delta$ if and only if it has property $\mathcal{P}_\mathcal{K}$.*

Proof. We show first that property $\mathcal{P}_\mathcal{K}$ implies property $\mathcal{P}_\delta$. Suppose $\mathcal{P}_\mathcal{K}$ holds and let $\epsilon > 0$ be arbitrary. Define

$$\delta(\epsilon) \coloneqq \begin{cases} \gamma^{-1}(\epsilon) & \epsilon \leq \gamma(b), \\ b & \text{else}, \end{cases}$$

which means $\gamma(\delta(\epsilon)) = \min(\epsilon, \gamma(b))$. Now, suppose $|x| \leq \delta(\epsilon)$. By Property $\mathcal{P}_\mathcal{K}$, we have

$$C(x) \leq \gamma(|x|) \leq \gamma(\delta(\epsilon)) \leq \min(\epsilon, \gamma(b)) \leq \epsilon,$$

which implies Property $\mathcal{P}_\delta$.

We now show that property $\mathcal{P}_\delta$ implies property $\mathcal{P}_\mathcal{K}$. Without loss of generality, we assume that the function $\delta(\epsilon)$ is strictly increasing. As proof, suppose property $\mathcal{P}_\delta$ holds for some $\hat{\delta}(\epsilon)$ which is possibly *not* increasing. Let $\bar{\delta}(\epsilon) \coloneqq \min(\hat{\delta}(\epsilon), 1)$. We note that property $\mathcal{P}_\delta$ holds also for $\bar{\delta}(\epsilon)$ because $\bar{\delta}(\epsilon) \leq \hat{\delta}(\epsilon)$. Next, let

$$\delta(\epsilon) \coloneqq (1 - e^{-\epsilon}) \sup_{s \in (0, \epsilon]} \bar{\delta}(s),$$

which is well-defined because $\bar{\delta}(\,\cdot\,)$ is bounded. Furthermore, $\delta(\epsilon)$ is increasing, as the first term is strictly increasing and the second is nondecreasing. To show that $\mathcal{P}_\delta$ holds for $\delta(\epsilon)$, choose $\epsilon_1 > 0$ arbitrarily. We claim there exists $\epsilon_0 \in (0, \epsilon_1]$ such that $\bar{\delta}(\epsilon_0) \geq \delta(\epsilon_1)$. Suppose not; then, for each $s \in (0, \epsilon_1]$, it holds that

$$\bar{\delta}(s) < \delta(\epsilon_1) < \sup_{s \in (0, \epsilon_1]} \bar{\delta}(s),$$

which is a contradiction because the strict upper bound $\delta(s)$ upper bound strictly less than the supremum. With this value, for all $x \in X$, we have

$$|x| \leq \delta(\epsilon_1) \implies |x| \leq \bar{\delta}(\epsilon_0) \implies C(x) \leq \epsilon_0 \implies C(x) \leq \epsilon_1$$

which means $\mathcal{P}_\delta$ holds for $\delta(\epsilon)$.

Now, define the doubly-infinite sequence $a_k = e^k$ for integers k and let $b_k \coloneqq \delta(a_{k-1})$. Note the shift in the definition of b_k so that $b_k < \delta(a_k)$. Define

$$\beta(s) \coloneqq b_k + (b_{k+1} - b_k) \frac{s - a_k}{a_{k+1} - a_k}, \qquad s \in [a_k, a_{k+1}], \quad k = 0, \pm 1, \pm 2, \ldots,$$

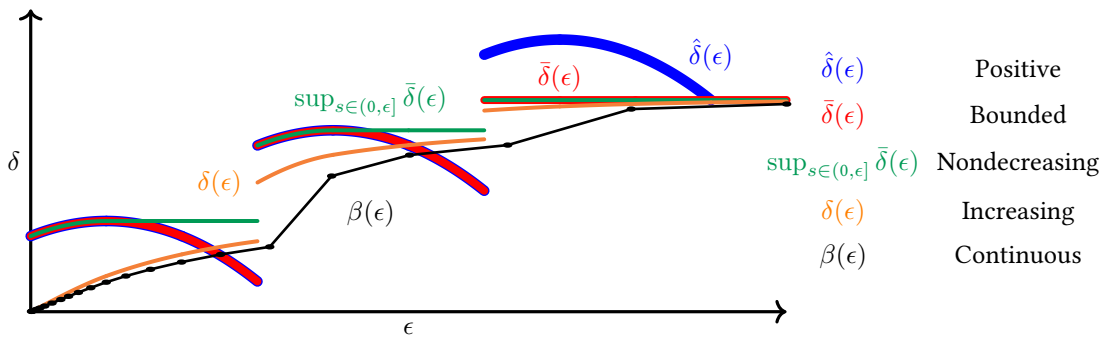


FIGURE 2.14: Construction of $\beta(\,\cdot\,)$ as used in the proof of Proposition 2.26. The listing on the right shows the additional property gained by each successive function.

with $\beta(0) \coloneqq 0$. The function $\beta(\,\cdot\,)$ is piecewise-linear, and thus continuous; in addition, because b_k is a strictly increasing sequence, $\beta(\,\cdot\,)$ is also increasing. Therefore, $\beta(\,\cdot\,)$ is a $\mathcal{K}$ function defined on $[0, \infty)$, and furthermore, $\beta(\epsilon) < \delta(\epsilon)$ for all $\epsilon > 0$.

Finally, choose any b such that $0 < b < \lim_{s\to\infty} \beta(s) \leq 1$, and let $\gamma(s) \coloneqq \beta^{-1}(s)$ for $s \in [0, b]$. Now, suppose $|x| \leq b$ and let $\epsilon = \gamma(|x|)$. Thus, because

$$|x| \leq \gamma^{-1}(\epsilon) = \beta(\epsilon) \leq \delta(\epsilon),$$

we have by property $\mathcal{P}_\delta$ that $C(x) \leq \epsilon = \gamma(|x|)$, Therefore, property $\mathcal{P}_\mathcal{K}$ holds for bound $\gamma(\,\cdot\,)$. Note that an example of the various functions constructed in this proof is shown in Figure 2.14 □

COROLLARY 2.27 ($\mathcal{K}$-function continuity): *A function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}^m$ is continuous at $x \in X$ if an only if there exists a scalar $b > 0$ and $\mathcal{K}$ function $\gamma(\,\cdot\,)$ such that*

$$|f(x) - f(y)| \leq \gamma(|x - y|)$$

for all $y \in X$ such that $|x - y| \leq b$.

Proof. Without loss of generality, assume $x = 0$. Define $C(y) \coloneqq |f(x) - f(y)|$ and apply Proposition 2.26 to standard ϵ/δ continuity. □

PROPOSITION 2.28 (Global $\mathcal{K}$-function overbound.): *Let $X \subseteq \mathbb{R}^n$ be closed and suppose that a function $V : X \to \mathbb{R}^m$ is continuous at $x_0 \in X$ and locally bounded on X (i.e., bounded on every compact subset of X). Then, there exists a $\mathcal{K}_\infty$-function $\alpha(\,\cdot\,)$ such that $|V(x) - V(x_0)| \leq \alpha(|x - x_0|)$ for all $x \in X$.*

Proof. First, by Corollary 2.27, we know that there exists a $\mathcal{K}$-function $\gamma(\,\cdot\,)$ and a constant $b_0 > 0$ such that $|V(x) - V(x_0)| \leq \gamma(|x - x_0|)$ whenever $|x - x_0| \leq b_0$ Starting from b_0, choose any strictly increasing and un-

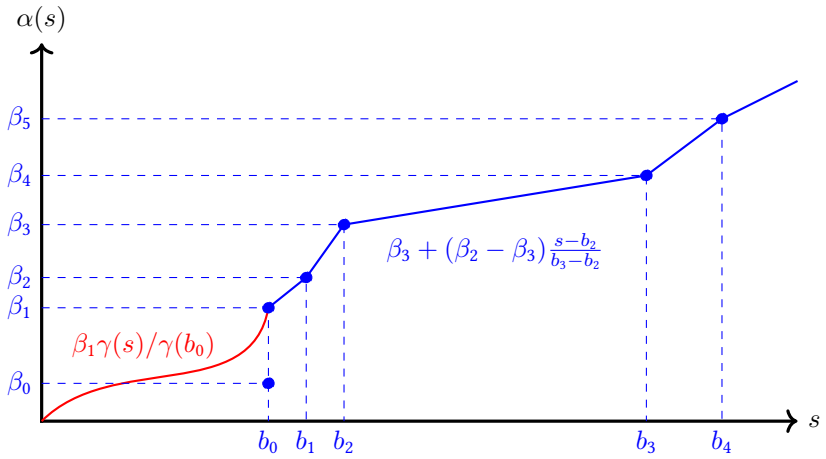


FIGURE 2.15: Construction of global $\mathcal{K}$ function overbound $\alpha(\,\cdot\,)$. On $[0, b_0]$, $\alpha(s)$ is constructed by rescaling $\gamma(s)$. For larger values of s, $\alpha(s)$ interpolates the points (b_k, β_{k+1})

bounded sequence $(b_k)_{k=0}^{\infty}$. For each $k \in \mathbb{I}_{\geq 1}$, define the set $B_k := \{x \in X : |x - x_0| \leq b_k\}$. We note that each B_k is a compact subset of X and further that $X = \bigcup_{k=0}^{\infty} B_k$. Next, define a sequence $(\beta_k)_{k=0}^{\infty}$ as

$$\beta_k := \sup_{x \in B_k} |V(x) - V(x_0)| + k$$

which is well-defined by compactness of the B_k and strictly increasing due to the extra $+k$ term. Finally, define

$$\alpha(s) := \begin{cases} \dfrac{\beta_1}{\gamma(b_0)}\gamma(s) & s \in [0, b_0) \\ \beta_{k+1} + (\beta_{k+2} - \beta_{k+1})\dfrac{s - b_k}{b_{k+1} - b_k} & s \in [b_k, b_{k+1}) \quad \forall k \in \mathbb{I}_{\geq 0} \end{cases}$$

We illustrate this construction in Figure 2.15. Clearly, $\alpha(0) = 0$ and α is continuous and increasing. Furthermore, because we have shifted the β_k as before, we see that $|V(x) - V(x_0)| \leq \alpha(|x - x_0|)$. □

COROLLARY 2.29 (Global $\mathcal{K}$-function overbound, time-varying case.): *Let $X \subseteq \mathbb{R}^n$ be closed and suppose that $V : X \times \mathbb{T} \to \mathbb{R}^m$ is uniformly continuous at $x_0 \in X$ and uniformly locally bounded on X (i.e., bounded on every compact subset of X). Then, there exists a $\mathcal{K}$-function $\alpha(\,\cdot\,)$ such that $|V(x,t) - V(x_0,t)| \leq \alpha(|x - x_0|)$ for all $x \in X$ and $t \in \mathbb{T}$.*

Proof. By applying Proposition 2.26 to uniform continuity, a uniform local $\mathcal{K}$-function bound on $V(\,\cdot\,)$ can be obtained. The corollary then follows from the the logic of Proposition 2.28 using

$$\beta_k := \sup_{t \in \mathbb{T}} \sup_{x \in B_k} |V(x,t) - V(x_0,t)| + k,$$

which is well-defined by the uniform locally boundedness property. □

2.6.3 Proof of Proposition 2.11

Proof. We will show that statements (a), (b), and (c) of Proposition 2.11 all imply (d), which itself implies Assumption 2.10.

(a) By Corollary 2.23, we have that $V_N^*(x,t) \leq V_f(x,t)$ for all $x \in \mathbb{X}_f(t)$. From continuity and Proposition 2.28, there exists a $\mathcal{K}_\infty$ function $\alpha(\,\cdot\,)$ such that $V_f(x,t) \leq \alpha(|x|)$ on $\mathbb{X}_f(t)$. Thus, condition (d) is implied.

(b) From uniform continuity of $V_N^*(\,\cdot\,)$, we know that for each $\epsilon > 0$, there exists $\delta > 0$ such that $|x| \leq \delta$ implies $V_N^*(x,i) \leq \epsilon$ for all $t \in \mathbb{T}$, recalling that $V_N^*(\,\cdot\,)$ is nonnegative and zero at the origin. By Corollary 2.27 in Section 2.6, this is equivalent to the existence of a $\mathcal{K}$ function $\gamma(\,\cdot\,)$ defined on $[0,b]$, for some $b > 0$, such that $V_N^*(x,i) \leq \gamma(|x|)$ for all $x \in X$, with $X \coloneqq \{x \in \mathbb{R}^n : |x| \leq b\}$ a neighborhood of the origin. Thus, condition (d) is also implied.

(c) The uniform continuity of $f(\,\cdot\,)$ and $\ell(\,\cdot\,)$ implies the existence of $\mathcal{K}_\infty$ function bounds of the form

$$\begin{aligned} |f(x,u,t)| &\leq \alpha_{fx}(|x|) + \alpha_{fu}(|u|) \\ \ell(x,u,t) &\leq \alpha_{\ell x}(|x|) + \alpha_{\ell u}(|u|) \end{aligned}$$

for all $(x,u) \in \mathbb{Z}(t)$ and $t \in \mathbb{T}$. Note that these bounds are *global* via application of Corollary 2.29.

Let $x \in \mathcal{X}_N(t)$ be arbitrary, and let $\mathbf{u} \in \mathcal{U}_N(x,t)$ satisfy $|\mathbf{u}| \leq \gamma(|x|)$ as per the hypothesis. Define $\mathbf{x}$ as the corresponding sequence of states. For the initial time point, we have

$$|x(1)| = f(x(0),u(0),t) \leq \alpha_{fx}(|x(0)|) + \alpha_{fu}(|u(0)|)$$

Proceeding recursively, we have

$$\begin{aligned} |x(k+1)| &\leq \alpha_{fx}(|x(k)|) + \alpha_{fu}(|u(k)|) \\ &\leq \alpha_{fx}\left(\alpha_{fx}(|x(k-1)|) + \alpha_{fu}(|u(k-1)|)\right) + \alpha_{fu}(|u(k)|) \\ &\leq \alpha_{fx}\left(2\,|x(k-1)|\right) + \alpha_{fx}\left(2\,|u(k-1)|\right) + \alpha_{fu}(|u(k)|) \\ &\leq \alpha_{fx}(2^k\,|x(0)|) + \sum_{j=0}^{k} \alpha_{fu}(2^{k-j}\,|u(j)|) \end{aligned}$$

in which we have used the fact that for any $\mathcal{K}_\infty$ function $\alpha(\,\cdot\,)$, it holds that $\alpha(s_1+s_2) \leq \alpha(2s_1) + \alpha(2s_2)$. Recalling that $|u(k)| \leq |\mathbf{u}| \leq \gamma(|x(0)|)$, we have

$$|x(k)| \leq \rho_k(|x(0)|), \quad \rho_k(s) \coloneqq \alpha_{fx}(2^{k-1}s) + \sum_{j=0}^{k-1} \alpha_{fu}(2^{k-1-j}\gamma(s))$$

for $k \geq 1$. Note also that each $\rho_k(\,\cdot\,)$ is a $\mathcal{K}_\infty$ function, as it is a finite sum and composition of $\mathcal{K}_\infty$ functions.

For the stage cost, we have

$$\begin{aligned}\ell(x(k), u(k), t+k) &\leq \alpha_{\ell x}\left(\rho_k(|x(0)|)\right) + \alpha_{\ell u}(|u(k)|) \\ &\leq \alpha_{\ell x}\left(\rho_k(|x(0)|)\right) + \alpha_{\ell u}(\gamma(|x(0)|))\end{aligned}$$

Similarly, for the terminal cost, we have

$$\begin{aligned}V_f(x(N), i+N) &\leq \alpha_2(|x(N)|) \\ &\leq \alpha_2\left(\rho_N(|x(0)|)\right)\end{aligned}$$

Applying these bounds to the objective function, we have

$$V_N(x, \mathbf{u}, t) \leq \alpha(|x(0)|), \quad \alpha(s) \coloneqq \sum_{k=0}^{N-1} \left(\alpha_{\ell x}(\rho_k(s)) + \alpha_{\ell u}(\gamma(s))\right) + \alpha_2\left(\rho_N(s)\right)$$

Because $\mathbf{u}$ is feasible, the optimal value of $V_N(\,\cdot\,)$ also satisfies this bound. Taking any $\epsilon > 0$, we have $V_N^*(x,t) \leq \alpha(|x|)$ for all $t \in \mathbb{T}$ and $x \in \mathbb{B}_\epsilon \cap \mathcal{X}_N(t)$, which implies condition (d).

(d) By Proposition 2.26, we use the bound $V^*(x,t) \leq \alpha(|x|)$ (which is valid for $|x| \leq \epsilon$ to conclude that $V^*(\,\cdot\,)$ is uniformly continuous at zero. In addition, because $V_N(\,\cdot\,)$ is uniformly locally bounded, $V_N^*(\,\cdot\,)$ is also uniformly locally bounded. Therefore, by Corollary 2.29, we can construct a global $\mathcal{K}$ overbound that is valid for $t \in \mathbb{T}$ and $\mathcal{X}_N(t)$, and Assumption 2.10 holds. □

How many people ruin themselves by laying out money on [setpoints] of frivolous utility?

— Adam Smith
The Theory of Moral Sentiments

Chapter 3

Extensions of MIMPC

3.1 Introduction

In the previous chapter, a problem formulation for mixed-integer MPC has been presented, and nominal stability properties are shown. While these developments form a solid basis for theory and application of MIMPC, they are not sufficient for the systems we would like to consider as major applications. Thus, in this chapter, we wish to develop extensions of nominal tracking MIMPC to cover a much wider class of systems. To this end, we present two main extensions: first, we consider the inherent robustness of MIMPC to small system disturbances to show that certain stability properties sill hold; second, we discuss performance a stability theorems to cover cases where MPC is not tracking a fixed setpoint or trajectory but rather directly optimizing a more tangible cost function. These developments are then used throughout the remainder of this thesis.

3.2 Inherent Robustness of MIMPC

The stability theory in the previous chapter was developed assuming nominal operation; that is, the system model $f(\,\cdot\,)$ is exact and the system evolves exactly as $x^+ = f(x, u, t)$. However, this assumption is unlikely to be satisfied in real systems, for example due to unmodeled disturbances and the intrinsic fact that the model $f(\,\cdot\,)$ does not exactly describe the system. Therefore, it is desirable to ensure that nominal system properties do not catastrophically deteriorate when small disturbances are present.

Within the literature, there are two main classes of techniques to address disturbances at the design phase (Mayne, 2016). In robust MPC, the system is designed to maintain constraint satisfaction subject to a set of disturbances that is fixed a priori (Mayne et al., 2005), e.g, using min/max approaches (Lazar et al., 2008) or tube-based methods (Falugi and Mayne, 2011). Alternatively, in stochastic MPC, cost and constraint satisfaction are considered in a probabilistic sense based on distributional knowledge of disturbances (Mesbah, 2016). In general, the goal is to optimize over feedback *policies* so that the system can hedge against and respond to disturbances as they are realized (Goulart et al., 2006). For more information about these methods, see Rawlings et al. (2017b, Chapter 3). Although robust-by-design approaches are valuable when disturbances are prevalent, we instead focus on demonstrating that nominal MPC is *inherently* robust, i.e., robust to small disturbances without any modification to the nominal controller.

For inherent robustness, Grimm et al. (2004) demonstrates instances where tight state or terminal constraints can lead to destabilization from arbitrarily small disturbances. When there are no state constraints, robustness can be shown via particular choice of the terminal region and control law (Yu et al., 2014). It can also be shown that robustness for optimal MPC follows from uniform continuity of the optimal cost function (Grimm et al., 2007). However, the converse is not true, and robustness *can* follow despite a discontinuous optimal cost and/or control law (Allan et al., 2017). Inherent robustness has also been shown for various formulations of suboptimal MPC (Allan et al., 2017; Lazar et al., 2008; Pannocchia et al., 2011). In Allan et al. (2016), it was also shown that these results extend to systems with discrete actuators. We will follow these developments with suitable extensions to consider generalized norms rather than true norms.

3.2.1 Robustness Formulation

For simplicity, we consider the class of additive state disturbances. Thus, the (autonomous) system evolves as

$$x^+ = f(x,t) + w \tag{3.1}$$

for the disturbance variable w. Under suboptimal MPC as defined in Section 2.2.5, we analyze stability using the extended state $z \coloneqq (x, \mathbf{u})$ in which $\mathbf{u}$ is the incumbent warm-start sequence of inputs. By analogy with (2.15), we define the disturbed closed-loop evolution of x as

$$z^+ \coloneqq \begin{pmatrix} x^+ \\ \mathbf{u}^+ \end{pmatrix} = \begin{pmatrix} f(x, \check{\kappa}(x,\mathbf{u},t),t) + w \\ \zeta(x,\check{\mathbf{u}},t) \end{pmatrix}, \quad \check{\mathbf{u}} \in \check{\mathcal{U}}_N^*(x,\mathbf{u},t),\ w \in \mathbb{W}. \tag{3.2}$$

for the disturbance taking values in $\mathbb{W}$. Recall that

$$\begin{aligned}
\zeta(x,\mathbf{u},t) &\coloneqq \big\{(u(1),\dots,u(N-1),u_f) : u_f \in \kappa_f(\phi(N,x,\mathbf{u},t),t+N)\big\},\\
\check{\mathcal{U}}_N(x,t) &\coloneqq \{\mathbf{u} \in \mathcal{U}_N(x,t) : V_N(x,\mathbf{u},t) \le \gamma(\|x\|_x) \text{ if } \|x\|_x \le \epsilon\},\\
\check{\mathcal{U}}_N^*(x,\mathbf{u},t) &\coloneqq \big\{\check{\mathbf{u}} \in \check{\mathcal{U}}_N(x,t) : V_N(x,\check{\mathbf{u}},t) \le V_N(x,\mathbf{u},t),\\
&\qquad\qquad \zeta(x,\check{\mathbf{u}},t) \in \check{\mathcal{U}}_N(x^+,t+1),\ x^+ = \phi(1,x,\check{\mathbf{u}},t)\big\},
\end{aligned}$$

as defined in the previous chapter.

As a definition of robustness, we consider input-to-state stability (ISS) of the extended state (Jiang and Wang, 2001). As general definitions, we take the following, which are time-varying extensions of the definitions in Allan et al. (2017). Note that in these definitions, we allow a generalized norm $\|\cdot\|$ on x, but we require a true norm (in this case, the Euclidean norm $|\cdot|$) on the disturbance sequence $\mathbf{w}$. Thus, we can choose to exclude components of x from this stability requirement, but the system must be robust to all components of w.

Definition 3.1 (Robust positive invariance): Sets $\mathcal{X}(t)$ are said to be robustly positive invariant under the disturbed system $x^+ = f(x,t) + w$ if there exists $\omega > 0$ such that, if $|w| \leq \omega$, then $f(x,t) + w \in \mathcal{X}(t+1)$ for all $x \in \mathcal{X}(t)$ and $t \in \mathbb{T}$.

Definition 3.2 (Robust asymptotic stability): The disturbed system $x^+ = f(x,t) + w$ is said to be robustly asymptotically stable for the generalized norm $\|\cdot\|$ on $\mathcal{X}(t)$ if the sets $\mathcal{X}(t)$ are robustly positive invariant and there exist $\omega > 0$, a $\mathcal{KL}$ function $\beta(\,\cdot\,)$, and a $\mathcal{K}_\infty$ function $\sigma(\,\cdot\,)$ such that, for all $t \in \mathbb{T}$,

$$\|x(t+k)\| \leq \beta(\|x(t)\|, k) + \sigma(|\mathbf{w}_{0:k-1}|)$$

with and $k \in \mathbb{I}_{\geq 0}$ for all trajectories starting from $x(t) \in \mathcal{X}(t)$ subject to disturbance sequences $\mathbf{w}$ with $|\mathbf{w}| \leq \omega$.

Analogous to the undisturbed case, we prove that the system is robustly asymptotically stable indirectly by means of a modified Lyapunov equation as follows:

Definition 3.3 (ISS Lyapunov function): A function $V : \mathcal{X} \times \mathbb{T} \to \mathbb{R}_{\geq 0}$ is said to be an ISS Lyapunov function for the disturbed system $x^+ = f(x,t) + w$ and generalized norm $\|\cdot\|$ on $\mathcal{X}(t)$ if the sets $\mathcal{X}(t)$ are robustly positive invariant and there exist $\omega > 0$, $\mathcal{K}_\infty$ functions $\alpha_1(\,\cdot\,)$, $\alpha_2(\,\cdot\,)$, and $\alpha_3(\,\cdot\,)$ as well as $\mathcal{K}$ function $\sigma(\,\cdot\,)$ such that

$$\alpha_1(\|x\|) \leq V(x,t) \leq \alpha_2(\|x\|) \tag{3.3}$$
$$V(f(x,t) + w, t+1) \leq V(x,t) - \alpha_3(\|x\|) + \sigma(|w|) \tag{3.4}$$

for all $x \in \mathcal{X}(t)$, w satisfying $|w| \leq \omega$, and $t \in \mathbb{T}$.

Theorem 3.4 (ISS Lyapunov stability theorem): *Suppose the sets $\mathcal{X}(t)$ are robustly positive invariant for the disturbed system $x^+ = f(x,t) + w$ and that a function $V(\,\cdot\,)$ is an ISS Lyapunov function for $f(\,\cdot\,)$. Then, the system is robustly asymptotically stable.*

Proof. A proof of the time-invariant case can be found in Allan et al. (2017). The time-varying case follows by the same logic, as the bounds $\alpha_i(\,\cdot\,)$ and $\sigma(\,\cdot\,)$ are time-invariant. □

3.2.2 Robustness of Suboptimal MPC

To prove robustness for suboptimal MPC, we use slightly more restrictive assumptions than in Chapter 2. These assumptions ensure favorable continuity properties for the system model and cost functions. Note, however, that as in Allan et al. (2017), these assumptions do not necessarily imply that the resulting optimal control law or optimal cost function are continuous.

Assumption 3.5 (Basic function properties): The functions $f(\,\cdot\,)$, $\ell(\,\cdot\,)$, and $V_f(\,\cdot\,)$ are uniformly continuous in t *and* x on $\mathbb{T}$ and $\mathbb{X}(t)$.

Assumption 3.6 (Basic set properties): For each $t \in \mathbb{T}$, $\mathbb{X}(t) = \mathbb{R}^n$ and $\mathbb{U}(t)$ is compact. The set $\mathbb{Z}(t) = \mathbb{X}(t) \times \mathbb{U}(t)$. The composite set $\mathbb{U}$ is bounded.

Assumption 3.7 (Norm and cost bounds): The generalized norm of the extended state $\|z\|$ can be expressed as $\|z\| = \|x\|_x + \|u\|_u$ for generalized norms $\|\cdot\|_x$ and $\|\cdot\|_u$. The sublevel sets of $\|\cdot\|_x$ are compact. There exists a $\mathcal{K}_\infty$ function $\alpha(\,\cdot\,)$ such that $\ell(x,u,t) \geq \alpha_1(\|x\|_x + \|u\|_u)$ for all $x \in \mathbb{Z}(t)$ and $t \in \mathbb{T}$. The function $\gamma(\,\cdot\,)$ is $\mathcal{K}_\infty$, and the terminal cost satisfies $0 \leq V_f(x,t) \leq \gamma(\|x\|_x)$ for all $x \in \mathbb{X}_f(t)$ and $t \in \mathbb{T}$.

Assumption 3.8 (Terminal control law): For each $t \in \mathbb{T}$ and $x \in \mathbb{X}_f(t)$, the set

$$\kappa_f(x,t) = \big\{u \in \mathbb{U}(t) : f(x,u,t) \in \mathbb{X}_f(t+1),\ V_f(f(x,u,t),t+1) \leq V_f(x,t) - \ell(x,u,t)\big\}$$

is nonempty. For $t \in \mathbb{T}$, the terminal sets satisfy $\mathbb{X}_f(t) = \operatorname{lev}_\tau V_f(\,\cdot\,,t)$ for some $\tau > 0$.

In comparison to Section 2.2.3, the assumptions here are stronger. We highlight the main differences as follows:

- The model and cost functions must be uniformly continuous in x rather than just continuous or lower semi-continuous.
- The set $\mathbb{X}(t)$ is required to be the entire space $\mathbb{R}^n$.
- No shared state/input constraints are allowed.
- The terminal cost $V_f(\,\cdot\,)$ must be overbounded by the $\mathcal{K}_\infty$ function $\gamma(\,\cdot\,)$.
- The terminal regions $\mathbb{X}_f(t)$ are required to be nontrivial in that they contain a neighborhood of the origin and are defined as a sublevel set of $V_f(\,\cdot\,)$.

In particular, the less general assumptions about the constraint sets is to cope with the fact that nonzero disturbances w could force the system to violate given constraints.

The main result of this section is the following stability theorem:

THEOREM 3.9 (Robust asymptotic stability of suboptimal tracking MPC): *Suppose Assumptions 3.5 to 3.8 are satisfied. Then, the closed-loop system* (3.2) *is robustly asymptotically stable on* $\check{\mathcal{Z}}_N(t)$.

Proof. The proof here is based on Allan et al. (2017), although notation has been changed. The proof consists of two main parts: we first demonstrate that $\mathcal{Z}_N(t)$ is robustly positive invariant, and we then show that the cost function $V_N(\,\cdot\,)$ is an ISS Lyapunov function.

For robust positive invariance, we start by noting that because $\phi(\,\cdot\,)$ is the finite composition of $f(\,\cdot\,)$, and both $f(\,\cdot\,)$ and $V_f(\,\cdot\,)$ are uniformly continuous in x, we can apply Proposition 2.26 to find a constant $\omega_1 > 0$ and a $\mathcal{K}_\infty$ function $\alpha_f(\,\cdot\,)$ such that

$$|V_f(\phi(N, x_1, t, \mathbf{u}), t+N) - V_f(\phi(N, x_2, \mathbf{u}, t), t+N)| \leq \alpha_f(|x_1 - x_2|),$$

for any $|x_1 - x_2| \leq \omega_1$. Now, suppose the system is currently at $z \coloneqq (x, \mathbf{u})$. Let $z^+ \coloneqq (x^+, \mathbf{u}^+)$ denote the (disturbed) successor state of (3.2), and let $\hat{x}^+ = x^+ - w$, i.e., the nominal evolution of the state x. Let $x_f \coloneqq \phi(N-1, x^+, t+1, \mathbf{u}^+)$, $x_f^+ \coloneqq \phi(N, x^+, t+1, \mathbf{u}^+)$, and $\hat{x}_f^+ \coloneqq \phi(N, \hat{x}^+, t+1, \mathbf{u}^+)$. We thus have

$$\left|V_f(x_f^+, t+N+1) - V_f(\hat{x}_f^+, t+N+1)\right| \leq \alpha_f\left(\left|x^+ - \hat{x}^+\right|\right) = \alpha_f(|w|).$$

Since $V_f(\,\cdot\,)$ is a scalar, we can infer that

$$V_f(x_f^+, t+N+1) \leq V_f(\hat{x}_f^+, t+N+1) + \alpha_f(|w|). \tag{3.5}$$

By Assumptions 3.7 and 3.8, we also have

$$V_f(\hat{x}_f^+, t+N+1) - V_f(x_f, t+N) \leq -\ell(x_f, u_f, t+N) \leq -\alpha_1(\|x_f\|_x + \|u_f\|_u),$$

in which u_f^+ is the final element of $\mathbf{u}^+$. Now, we claim that there exists a constant $\tau^+ \in (0, \tau)$ such that $V_f(\hat{x}_f^+, t+N+1) \leq \tau^+$:

- Suppose $V_f(x_f, t+N) \geq \tau/2$. From Assumption 3.7, we have $\|x_f\|_x \geq \alpha_2^{-1}(\tau/2)$. Therefore, $V_f(\hat{x}_f^+, t+N+1) - V_f(x_f, t+N) \leq -\alpha_1(\alpha_2^{-1}(\tau/2))$. Thus, $V_f(\hat{x}_f^+, t+N+1) \leq \tau - \alpha_1(\alpha_2^{-1}(\tau/2))$.
- Alternatively, suppose $V_f(x_f, t+N) < \tau/2$. From the cost decrease condition in Assumption 3.8, we have immediately that $V_f(\hat{x}_f^+, t+N+1) \leq \tau/2$.

Therefore, taking $\tau^+ = \min(\tau/2, \tau - \alpha_1(\alpha_2^{-1}(\tau/2)))$, the statement holds. Combining this bound with (3.5), we have that

$$V_f(x_f^+, t+N+1) \leq \tau^+ + \alpha_f(|w|).$$

Thus, if we restrict $|w| \leq \omega_2 \coloneqq \alpha_f^{-1}(\tau - \tau^+)$, we have that $V_f(x_f^+, t+N+1) \leq \tau$, and so $x_f^+ \in \mathbb{X}_f(t+N+1)$. Therefore, $\mathbf{u}^+ \in \check{\mathcal{U}}_N(x^+, t+1)$, and $z^+ \in \check{\mathcal{Z}}_N(t+1)$. We conclude that for $|w| \leq \omega \coloneqq \min(\omega_1, \omega_2)$, we have that $\check{\mathcal{Z}}_N(t)$ is robustly positive invariant.

For the Lyapunov function, we note first that the stronger assumptions imply Theorem 2.18 from the previous chapter. Therefore, we immediately have the existence of $\mathcal{K}_\infty$ functions $\alpha_1(\,\cdot\,)$, $\alpha_2(\,\cdot\,)$, and $\alpha_3(\,\cdot\,)$ such that $\alpha_1(\|z\|) \leq V_N(z,t) \leq \alpha_2(\|z\|)$ and

$$V_N(\hat{z}^+, t+1) \leq V_N(z,t) - \alpha_1(\|x\|),$$

for $\hat{z}^+ \coloneqq (\hat{x}^+, \mathbf{u}^+)$ giving the nominal evolution of the system. We then appeal to the fact that $V_N(\,\cdot\,)$ is uniformly continuous in x and t (by composition of $f(\,\cdot\,)$, $\ell(\,\cdot\,)$, and $V_f(\,\cdot\,)$, all of which are uniformly continuous). Therefore, by application of Proposition 2.26, we have a constant $\omega_3 > 0$ and a $\mathcal{K}_\infty$ function $\alpha_V(\,\cdot\,)$ such that

$$\left|V_N(x^+, \mathbf{u}^+, t+1) - V_N(\hat{x}^+, \mathbf{u}^+, t+1)\right| \leq \sigma_V(\left|x^+ - \hat{x}^+\right|) = \sigma_V(|w|).$$

As before, we can remove the absolute value to find

$$V_N(x^+, \mathbf{u}^+, t+1) \leq V_N(\hat{x}^+, \mathbf{u}^+, t+1) + \sigma_V(|w|).$$

Combining this relationship with the nominal cost decrease, we have

$$V_N(z^+, t+1) \leq V_N(z,t) - \alpha_3(\|z\|) + \sigma_V(|w|)$$

as the necessary decrease condition. Therefore, $V_N(\,\cdot\,)$ is an ISS Lyapunov function, and by Theorem 3.4, the system is robustly asymptotically stable. □

The proof is based on the fact that when the warm start adds the additional terminal input, the *predicted* trajectory moves into the interior of $\mathbb{X}_f(t)$. Thus, when the disturbance w affects the system, the resulting x_f may not be in $\mathbb{X}_f(t)$; however, when the additional control input is applied, the successor state x_f^+ *is* in $\mathbb{X}_f(t)$. We illustrate a time-invariant case in Figure 3.1. For $w = 0$, the system would move to $\hat{x}^+ \in \mathcal{X}_{N-1}$. Because of the disturbance, the system instead moves to x^+, but for sufficiently small w, it is guaranteed that $x^+ \in \mathcal{X}_N$. Therefore, although the perturbed value of $x_f \notin \mathbb{X}_f$ when the additional control input added by $\zeta(\,\cdot\,)$, the successor state x_f^+ does reach $\mathbb{X}_f$. Thus, $\check{\mathcal{Z}}_N$ is robustly invariant.

The main consequence of Theorem 3.9 is essentially that a tracking MPC controller that is nominally stable cannot be destabilized by arbitrarily small disturbances. This property is important because even the most accurate model is not going to exactly match the true system (e.g., due to finite numerical precision of the optimizer). A potential downside of the formulation of robustness as (local) ISS is that robust stability only holds for disturbance sequences $\mathbf{w}$ satisfying $|\mathbf{w}| \leq \omega$ for some constant $\omega > 0$. One can of course construct systems where this ω is very small, for example by creating a model $f(\,\cdot\,)$ that is nearly discontinuous. In such cases, very small (but not arbitrarily small) w could push the system out of $\mathcal{X}_N(t)$, and thus at the next timestep, the MPC problem is infeasible. In specific applications where more explicit guarantees are necessary, robustness by design (e.g., by applying robust or stochastic

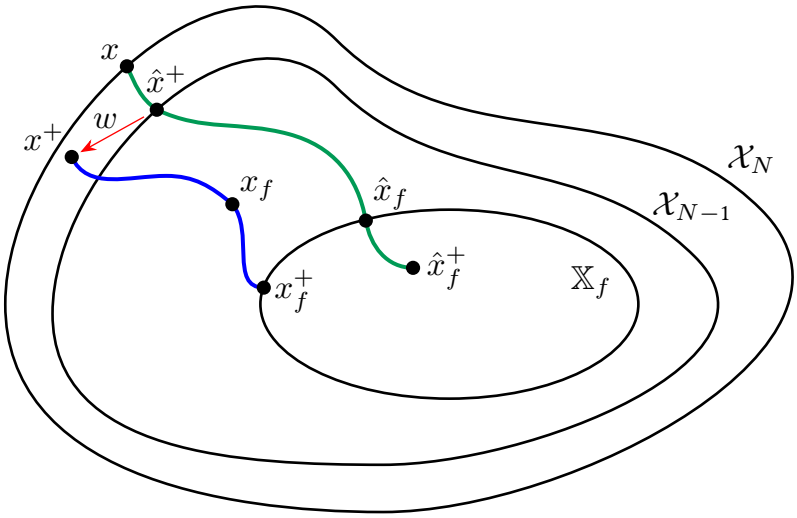


FIGURE 3.1: Illustration of the proof of Theorem 3.9. The system is time-invariant.

MPC techniques) may be necessary. However, experience has shown that in most practical cases, ω can be made quite large with the system still remaining relatively close to the setpoint. Thus, the inherent robustness of tracking MIMPC is often sufficient.

3.2.3 REMARKS

REMARK 3.10: In Assumption 3.6, is was assumed that the sets $\mathbb{X}(t)$ are all of $\mathbb{R}^n$. This restriction ensures that the MPC problem could not be rendered infeasible by a disturbance pushing x out of $\mathbb{X}(t)$ by an arbitrarily small amount. For cases where the sets $\mathbb{X}(t)$ represent true physical limitations of the system, the assumption can be relaxed, as no physical disturbance could lead to violation of this constraint. For example, if a state represents the mass of water in a tank, then that state is intrinsically constrained to be nonnegative; thus, no realization of w could cause $x < 0$. As long as one assumes that the disturbances w cannot cause the system to leave $\mathbb{X}(t)$, then the preceding results still hold. However, it is common to employ state constraints to enforce bounds that are desirable but can clearly be violated by the system. Fortunately, these considerations can be included in the optimization problem via soft constraints. If the slack variables on these constraints are subject to sufficiently large penalties, then the optimizer will typically not choose to violate them if possible. If a disturbance happens to cause these constraints to be violated, then the optimizer will try to choose u such that the violation is reduced and ultimately removed. Indeed, by augmenting the state to include these constraint violations, clever choice of $\|\cdot\|_x$ can ensure that the slacks are asymptotically zero in the nominal case and bounded when there are disturbances.

Remark 3.11: As formulated in this section, robustness is with respect to additive state disturbances that leads to $x^+ \neq f(x, u, t)$ as predicted. However, this formulation still assumes that the optimizer gets to know the true value of the state x at every timestep. A more realistic treatment considers the possibility that the value of x known to the controller is not exactly the true system value of x, for example due to measurement noise or state estimation error. However, it can be shown Roset et al. (2008) that if a system is inherently robust to state disturbances, then it is also robust to measurement disturbances. The proof essentially proceeds by reformulating the problem in terms of the measured state, so that the measurement error has a similar effect to the state disturbance; by bounding the measurement error, robust stability of the measured state implies robust stability of the true system state. Indeed, this is the approach taken in Allan et al. (2017) to show robustness to both types of disturbances.

3.3 Economic MPC

Thus far in this work, we have assumed that each MIMPC problem optimizes an objective function that has been designed specifically to stabilize a given setpoint. In particular, the cost functions are nonnegative a, have global minima at $\|x\| = 0$, and on average are increasing with increasing $\|x\|$. Such restrictions provide favorable stability properties, but they require that a feasible setpoint is provided.

Assuming setpoints are generated by an oracle in accordance with higher-level objectives, then stabilizing said setpoint will likely achieve the desired goals. However, many systems do not have the benefit of such a wise oracle. For example, systems that are inherently unsteady (e.g., sequential batch production environments) may not be able satisfy constraints by remaining at a static setpoint. In addition, if disturbances are highly time-varying, then costs could potentially be reduced by reacting asymmetrically to "good" and "bad" disturbances. However, as with standard tracking control, it is often nontrivial to decide what sequence of inputs best achieves a given goal, and thus systems can still benefit significantly by being able to optimize online. Therefore, we wish to extend tracking MIMPC to apply to cases where the *cost function*, rather than the setpoint, has been provided. Because such objective functions typically reflect some measure of economic performance, we refer to this case as "economic MPC."

3.3.1 Literature Review

Economic MPC was first discussed in Amrit et al. (2011) as an extension of tracking MPC. Although the controller's cost function does not satisfy positive definiteness or other properties, performance bounds and asymptotic stability to a given steady-state operating point have been established for time-invariant systems (Angeli et al., 2012; Diehl et al., 2011; Ellis et al., 2014). Under suitable conditions, it can also be shown that these properties hold without terminal constraints in the limit of a long prediction horizon (Grüne and Stieler, 2014; Müller and Grüne, 2015). If it is undesirable to choose a particular steady state a-priori, Müller et al. (2013) presents a method that allows the

optimizer to choose its own steady state to use as a terminal constraint. When certain dissipativity conditions do not hold, stability can be analyzed in a multiobjective framework (Griffith et al., 2017; Zavala, 2015), in which stability is enforced by including decrease of a tracking-MPC-like cost function as an explicit optimization constraint. Strategies have also been developed to apply robust and stochastic MPC techniques to economic MPC in order to achieve better average performance in the case of disturbances (Bayer et al., 2014, 2016).

For systems optimally operated in a periodic cycle, the previous steady-state results can generally be extended in a straightforward manner. By converting to deviation variables (Huang et al., 2011), stability of the periodic cycle is equivalent to stability of the origin in the shifted system. Various dissipativity conditions can be formulated in terms of deviation from the invariant set of a given periodic solution (Zanon et al., 2013, 2017b), once again demonstrating stability. In these works, the underlying system is still considered to be time-invariant. Truly time-varying systems have been considered under the more restrictive assumptions of a linear storage function and the existence of global Lipschitz constants for the model and stage cost (Zanon et al., 2013). It can also be shown that certain dissipativity conditions imply local linear equivalence of tracking and economic MPC (Zanon et al., 2017a), which confers the stability of tracking MPC without losing the tangible objective function of economic MPC. Finally, *necessity* of certain dissipativity conditions for stability is demonstrated in Muller et al. (2015). None of these works explicitly consider the inclusion of mixed-integer actuators, although as discussed in the previous chapter, it is likely that they are compatible with such input sets. However, because of the difficulty of global optimization for nonconvex problems (especially when discrete variables are present), we wish to extend some of these properties to cover suboptimal economic MPC.

3.3.2 Problem Formulation

For the results in this section, we follow Risbeck and Rawlings (2018b), with some extensions to consider suboptimal economic MPC. We consider once again the case of time-varying state-space systems evolving according to

$$x^+ = f(x, u, t)$$

defined for $(x, u) \in \mathbb{Z}(t) \subseteq \mathbb{X}(t) \times \mathbb{U}(t)$, and $t \in \mathbb{T}$. We again let $\mathbb{X} \coloneqq \bigcup_{t\in\mathbb{T}} \mathbb{X}(t)$ and $\mathbb{U} \coloneqq \bigcup_{t\in\mathbb{T}} \mathbb{U}(t)$, and also choose terminal sets $\mathbb{X}_f(t)$. We then recall the following definitions from Chapter 2:

$$\begin{aligned}
\mathcal{Z}_N(t) &\coloneqq \{(x, \mathbf{u}) : (\phi(k, x, \mathbf{u}, t), u(k)) \in \mathbb{Z}(t + k) \text{ for all } k \in \mathbb{I}_{[0,N)}, \phi(N, x, \mathbf{u}, t) \in \mathbb{X}_f(t + N)\} \\
\mathcal{X}_N(t) &\coloneqq \{x \in \mathbb{X} : \text{there exists } \mathbf{u} \in \mathbb{U}^N \text{ such that } (x, \mathbf{u}) \in \mathbb{Z}_N(t)\} \\
\mathcal{U}_N(x, t) &\coloneqq \{\mathbf{u} \in \mathbb{U}^N : (x, u) \in \mathcal{Z}_N(t)\},
\end{aligned}$$

which are defined for a chosen horizon $N \in \mathbb{I}_{\geq 0}$. However, we do not assume that the system has been shifted in any way, and we attach no special significance to the point $x = 0$.

As a cost function, we also take

$$V_N(x, \mathbf{u}, t) \coloneqq \sum_{k=0}^{N-1} \ell(x(k), u(k), t) + V_f(x(N), t+N),$$

with $x(k) \coloneqq \phi(k, x, \mathbf{u}, t)$, $k \in \mathbb{I}_{[0,N]}$. The stage cost $\ell(\,\cdot\,)$ and terminal cost $V_f(\,\cdot\,)$ serve a similar purpose as in tracking MPC, but as we will see, their required properties are much less restrictive.

To avoid the need to solve any optimization problems to optimality, we once again consider a suboptimal formulation with extended state $z \coloneqq (x, \mathbf{u}) \in \mathcal{Z}_N(t)$. The "optimization" problem is thus to choose any sequence $\check{\mathbf{u}}$ in $\check{\mathcal{U}}^*_N(z, t)$ with

$$\check{\mathcal{U}}^*(x, \mathbf{u}, t) \coloneqq \big\{\mathbf{u} \in \mathcal{U}_N(x, t) : V_N(x, \check{\mathbf{u}}, t) \leq V_N(x, \mathbf{u}, t)\big\}. \tag{3.6}$$

We do not require the technical condition that was used for tracking MPC. Note that the corresponding optimal economic MPC problem

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} V_N(x, \mathbf{u}, t) \text{ such that } \mathbf{u} \in \mathcal{U}_N(x, t)$$

is included within $\check{\mathcal{U}}^*(x, \mathbf{u}, t)$.

With this procedure, the closed-loop evolution is

$$z^+ \coloneqq \begin{pmatrix} x^+ \\ \mathbf{u}^+ \end{pmatrix} = \begin{pmatrix} f(x, \check{\kappa}(x, \mathbf{u}, t), t) \\ \zeta(x, \check{\mathbf{u}}, t) \end{pmatrix}, \quad \check{\mathbf{u}} \in \check{\mathcal{U}}^*_N(x, \mathbf{u}, t)$$

based on the warm-start update

$$\zeta(x, \mathbf{u}, t) \coloneqq \big\{(u(1), \dots, u(N-1), u_f) : u_f \in \kappa_f(\phi(N, x, \mathbf{u}, t), t+N)\big\}, \tag{3.7}$$

by applying an action from the terminal control law $\kappa_f(\,\cdot\,)$. Although both updates are, strictly speaking, a differential inclusion, we assume some selection rule is applied and note that the results hold regardless of the selection rule.

3.3.3 Asymptotic Performance

In this section, we consider the construction of the terminal ingredients $\mathbb{X}_f(t)$, $V_f(\,\cdot\,)$, and $\kappa_f(\,\cdot\,)$ such that the closed-loop system is nominally recursively feasible and satisfies certain bounds on closed-loop cost. As a basis for measuring cost, we assume that we are given an infinite time-varying (but fixed) reference trajectory $(\mathbf{x}_r, \mathbf{u}_r)$. This trajectory could be a fixed steady state, a periodic orbit, or a general time-varying path generated by arbitrary means. We then take the following mild assumptions:

Assumption 3.12: For each $t \in \mathbb{T}$, the set $\mathbb{X}(t)$ is closed, and the set $\mathbb{U}(t)$ is compact. The composite set $\mathbb{U}$ is bounded. The reference trajectory satisfies $(x_r(t), u_r(t)) \in \mathbb{Z}(t)$, $x_r(t) \in \mathbb{X}_f(t)$, and $x_r(t+1) = f(x_r(t), u_r(t), t)$.

Assumption 3.13: For each $t \in \mathbb{T}$, the terminal control set

$$\kappa_f(x,t) = \big\{u \in \mathbb{U}(t) : f(x,u,t) \in \mathbb{X}_f(t+1), \\ V_f(f(x,u,t),t+1) + \ell(x,u,t) \le V_f(x,t) + \ell(x_r(t),u_r(t),t)\big\}$$

is nonempty, and $V_f(x_r(t),u_r(t),t) = 0$.

Assumption 3.14: The cost functions $\ell(\,\cdot\,)$ and $V_f(\,\cdot\,)$ are bounded below on $\mathbb{Z}(t)$ and $\mathbb{X}_f(t)$ respectively, with the bounds holding uniformly for $t \in \mathbb{T}$.

Based on these assumptions, we have the following result as a generalization of Amrit et al. (2011).

Theorem 3.15 (Asymptotic performance of economic MPC): *Suppose Assumptions 3.12 to 3.14 are satisfied. Then, starting from any $z(t) \in \mathcal{Z}_N(t)$, the sets $\mathcal{Z}_N(t)$ are positive invariant and for any $T \in \mathbb{I}_{\ge 0}$,*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{k=t}^{t+T-1} \big(\ell(x(k),u(k),k) - \ell(x_r(k),u_r(k),k)\big) \le 0 \tag{3.8}$$

under closed-loop evolution (3.7).

Proof. From the properties of the terminal control law in Assumption 3.13, we have immediately that $\mathcal{U}_N(x,t)$ is invariant under the update $\zeta(\,\cdot\,)$. Thus, $\mathbf{u}^+ = \zeta(x,\check{\mathbf{u}},t) \in \mathcal{U}_N(x^+,t+1)$ for $x^+ = \phi(1,x,\check{\mathbf{u}},t)$, and therefore $\mathcal{Z}_N(t)$ is positive invariant.

For the performance bound, we first define the *shifted* optimal cost function

$$\bar{V}_N(x,\mathbf{u},t) \coloneqq V_N(x,\mathbf{u},t) - V_N(x_r(t),\mathbf{u}_r(t),t)$$

in which $\mathbf{u}_r(t) \coloneqq \mathbf{u}_{r,t:(t+N-1)}$ is the length-N subsequence of $\mathbf{u}_r$ starting at time t. Now, given the chosen optimal sequence $\check{\mathbf{u}}$, let u be its first element, and let $x_f \coloneqq \phi(N,x,\check{\mathbf{u}},t)$. From the update function $\zeta(\,\cdot\,)$, we have immediately that

$$\begin{aligned}\bar{V}_N(z^+,t+1) = \bar{V}_N(x,\check{\mathbf{u}},t) &- \ell(x,u,t) - V_f(x_f,t+N) \\ &+ \ell(x_f,u_f,t+N) + V_f(x_f^+,t+N+1) \\ &+ \ell(x_r(t),u_r(t),t) + \ell(x_r(t+N),u_r(t+N),t+N),\end{aligned}$$

in which $u_f \in \kappa_f(x_f,t+N)$ is the terminal control input, and $x_f^+ \coloneqq f(x_f,u_f,t+N)$. From the cost decrease condition in Assumption 3.13, we know that the sum of four of those terms is nonnegative, and thus we have

$$\begin{aligned}\bar{V}_N(z^+,t+1) &\le \bar{V}_N(x,\check{\mathbf{u}},t) - \ell(x,u,t) + \ell(x_r(t),u_r(t),t) \\ &\le \bar{V}_N(z,t) + \ell(x,u,t) + \ell(x_r(t),u_r(t),t)\end{aligned}$$

noting that the cost of $(x,\check{\mathbf{u}})$ is less than that of $(x,\mathbf{u})$.

Now, the previous cost bound applies for any $t' \geq t$. Thus, we rearrange and sum the bound for T timesteps to find

$$\bar{V}_N(z(t),t) - \bar{V}_N(z(t+T),t+T) \geq \sum_{k=t}^{t+T-1} \big(\ell(x(t),u(t),t) - \ell(x_r(t),u_r(t),t)\big).$$

By Assumption 3.14, we know that the second term on the right-hand side is bounded below. Therefore, there exists some constant $c \in \mathbb{R}$ such that

$$\bar{V}_N(z(t),t) - \bar{V}_N(z(t+T),t+T) \leq c.$$

Applying that bound and dividing by T, we have

$$\frac{1}{T} \sum_{k=t}^{t+T-1} \big(\ell(x(t),u(t),t) - \ell(x_r(t),u_r(t),t)\big) \leq \frac{c}{T},$$

and taking the $\limsup$ of both sides immediately gives the desired cost bound. □

Theorem 3.15 essentially states that, after an initial transient, the average closed-loop cost of the controlled trajectory will be no worse than the cost of the reference trajectory. If the reference was chosen poorly, then there is a significant opportunity for improvement, and the optimizer will do so provided that the horizon is sufficiently long. On the other hand, if the reference trajectory provides a near-optimal closed-loop cost, then the optimizer will not deviate significantly. Note also the extremely mild assumption on the stage cost $\ell(\,\cdot\,)$; by requiring only boundedness, we can choose almost any function, which means the optimizer can directly consider a tangible measure of economic performance.

Another point of note that we have not assumed any continuity properties for $f(\,\cdot\,)$, $\ell(\,\cdot\,)$, or $V_f(\,\cdot\,)$. Indeed, Theorem 3.15 holds regardless of these properties. However, although the suboptimal strategy does not explicitly require any optimization (aside from finding the initial input sequence $\mathbf{u}$), system performance is likely to benefit from direct optimization. Thus one would likely use continuous functions so that standard optimization techniques can be applied.

Finally, we note that it can be difficult to synthesize nontrivial terminal sets $\mathbb{X}_f(t)$ such that Assumption 3.13 is satisfied. If the system model $f(\,\cdot\,)$ and the stage costs $\ell(\,\cdot\,)$ are quadratic, then it may be possible to obtain suitable $V_f(\,\cdot\,)$ and $\mathbb{X}_f(t)$ by applying the theory of linear systems. However, for nonlinear or other exotic cases, we note that choosing $\mathbb{X}_f(t) \coloneqq \{x_r(t)\}$ trivially satisfies the assumption with $V_f(x,t) \equiv 0$ and $\kappa_f(x,t) \equiv \{u_r(t)\}$. This choice requires that the system terminate exactly on (and in phase with) the reference trajectory, and it is of course trivial to synthesize. Unfortunately, using an exact terminal equality constraint often leads to much smaller sets $\mathcal{X}_N(t)$ (i.e., a much smaller feasible region), but this effect can be partially mitigated by using a longer horizon and not necessarily solving problems to optimality. Thus, this terminal constraint can be applied to a wide variety of systems, which ensures a baseline level of profitability for the nominal closed-loop system.

3.3.4 Asymptotic Stability

While the result of the previous section demonstrates that the closed-loop cost of $x(t)$, $u(t)$ is asymptotically no worse than that of the reference trajectory $x_r(t)$, $u_r(t)$, in certain cases, it may be desirable to know that the system will actually converge to the reference trajectory. Thus, we describe conditions under which economic MPC asymptotically stabilizes the reference trajectory. For simplicity of exposition for this result, we will consider only optimal MPC. Thus, the system evolves according to

$$x^+ = f(x, \kappa_N(x,t), t), \;\; \kappa_N(x,t) = u^*(0), \;\; \mathbf{u}^* = \arg\min_{\mathbf{u}} V_N(x, \mathbf{u}, t) \text{ subject to } \mathbf{u} \in \mathcal{U}_N(x,t). \tag{3.9}$$

Note however that the results of this section hold also for suboptimal MPC, as long as the same tedious technical conditions from Section 2.2.5 are applied to avoid pathological behavior caused by a poor warm start. To ensure that the optimal cost $V_N^*(\,\cdot\,)$ is well-defined, we take the following assumption:

Assumption 3.16: The function $f(\,\cdot\,)$ is continuous. The functions $\ell(\,\cdot\,)$ and $V_f(\,\cdot\,)$ are lower semi-continuous.

To start, we begin by describing the notion of dissipativity, extended from (Angeli et al., 2012, Definition 4.1) to the time-varying case. We can of course recover the time-invariant definitions as a special case us a steady-state reference trajectory $(x_r(t), u_r(t)) = (x_s, u_s)$, in which case the storage function need not be time-varying.

Definition 3.17 (Dissipativity; time-varying): The system $x^+ = f(x,u,t)$ is dissipative with respect to a reference trajectory $\mathbf{x}_r$ and supply rate $s(x,u,t) : \mathbb{X} \times \mathbb{U} \times \mathbb{T} \to \mathbb{R}$ if there exists a storage function $\lambda(x,t) : \mathbb{X} \times \mathbb{T} \to \mathbb{R}$ satisfying $\lambda(x_r(t), t) = 0$ and

$$\lambda(f(x,u,t), t+1) - \lambda(x,t) \leq s(x(t), u(t), t)$$

for all $(x,u) \in \mathbb{Z}(t)$ and $t \in \mathbb{T}$.

The system is *strictly* dissipative if, in addition, there exists a $\mathcal{K}_\infty$ function $\alpha(\,\cdot\,)$ such that

$$\lambda(f(x,u,t), t+1) - \lambda(x,t) \leq s(x,u,t) - \alpha\left(|x - x_r(t)|\right)$$

for all all $(x,u) \in \mathbb{Z}(t)$ and $t \in \mathbb{T}$.

Assumption 3.18 (Strict Dissipativity): The system is strictly dissipative with supply rate $s(x,u,t) = \ell(x,u,t) - \ell(x_r(t), u_r(t), t)$.

Using the storage function, we define a "rotated" optimal cost function as follows

$$\tilde{V}_N^0(x,t) = V_N^0(x,t) + \lambda(x,t) - V_N(x_r(t), \mathbf{u}_r(t), t), \tag{3.10}$$

which is modified by the storage function $\lambda(\,\cdot\,)$ and also shifted by the cost of the reference trajectory (so that $\tilde{V}(\,\cdot\,) \equiv 0$ along the reference trajectory). Similar to Assumption 2.10 from Chapter 2, we make the following assumption about $\tilde{V}(\,\cdot\,)$.

Assumption 3.19 (Continuity of rotated optimal cost): The rotated optimal value function $\tilde{V}_N^0(x,t)$ is uniformly continuous at $(x_r(t),t)$ for $t \in \mathbb{T}$.

Following the logic of Proposition 2.11, this assumption implies the existence of a $\mathcal{K}_\infty$ function $\gamma(\,\cdot\,)$ such that

$$\tilde{V}_N^0(x,t) \leq \gamma(|x - x_r(t)|)$$

for all $x \in \mathcal{X}_N(t)$ and $t \in \mathbb{T}$. We then have the following theorem:

Theorem 3.20 (Asymptotic stability of economic MPC): *Suppose Assumptions 3.12, 3.13, 3.16, 3.18 and 3.19 are satisfied. Then the closed-loop system* (3.9) *is asymptotically stable, i.e., there exists a $\mathcal{KL}$ function $\beta(\,\cdot\,)$ such that for any initial time $t \in \mathbb{T}$, $x \in \mathcal{X}_N(t)$, and $k \in \mathbb{I}_{\geq 0}$,*

$$|x(t+k) - x_r(t+k)| \leq \beta\left(|x - x_r(t)|, k\right)$$

along closed-loop solutions.

Proof. The proof proceeds by showing that a suitably rotated version of the optimization problem satisfies the assumptions of Theorem 2.13 and then showing that the rotated optimization problem is equivalent to the nominal optimization problem.

To start, define the following rotated cost functions:

$$\begin{aligned}
\tilde{\ell}(x,u,t) &\coloneqq \ell(x,u,t) - \ell(x_r(t),u_r(t),t) + \lambda(x,t) - \lambda(f(x,u,t),t+1) \\
\tilde{V}_f(x,t) &\coloneqq V_f(x,t) - V_f(x_r(t),t) + \lambda(x,t)
\end{aligned}$$

We have immediately from Assumption 3.16 that the rotated cost functions satisfy Assumption 2.6. For the terminal cost, we have from Assumption 3.13 that

$$\begin{aligned}
\tilde{V}_f(x^+,t) + \tilde{\ell}(x,u,t) &= V_f(x^+,t) + \ell(x,u,t) + \lambda(x,t) - \ell(x_r(t),u_r(t),t) - V_f(x_r(t+1),t+1) \\
&\leq V_f(x,t) + \ell(x_r(t),u_r(t),t) + \lambda(x,t) - \ell(x_r(t),u_r(t),t) - V_f(x_r(t+1),t+1) \\
&= \tilde{V}_f(x,t) + V_f(x_r(t),t) - V_f(x_r(t+1),t+1) \\
&= \tilde{V}_f(x,t)
\end{aligned}$$

which gives Assumption 2.9. The rotated cost function is then

$$\tilde{V}_N(x,\mathbf{u},t) \coloneqq \sum_{k=0}^{N-1} \tilde{\ell}(x(k),u(k),t+k) + \tilde{V}_f(x(N),t+N)$$

with the analogous rotated optimization

$$\min_{\mathbf{u}} \tilde{V}_N(x,\mathbf{u},t) \text{ subject to } \mathbf{u} \in \mathcal{U}_N(x,t). \tag{3.11}$$

Let $\tilde{\kappa}(\,\cdot\,)$ denote the corresponding optimal first input.

By Assumption 3.18, we have that for all $x \in \mathcal{X}_N(t)$ and $t \in \mathbb{T}$,

$$\tilde{\ell}(x, u, t) \geq \alpha\left(|x - x_r(t)|\right).$$

Therefore, $\tilde{\ell}(\,\cdot\,)$ and $\tilde{V}_f(\,\cdot\,)$ satisfy Assumption 2.8. As discussed above, Assumption 3.19 implies Assumption 2.10, and Assumption 3.12 corresponds to Assumption 2.7. We conclude that the rotated system satisfies Theorem 2.13, and so the rotated system is asymptotically stable. Note that the original theorem is for stability to the origin, but we immediately recover stabilization of $\mathbf{x}_r$ using deviation variables $\bar{x} \coloneqq x - x_r(t)$, and $\bar{u} \coloneqq u - u_r(t)$.

To complete the proof, we show that the rotated optimization (3.11) is equivalent to nominal optimiation, i.e., that $\tilde{\kappa}_N(\,\cdot\,) \equiv \kappa_N(\,\cdot\,)$. Substituting the definitions of $\tilde{\ell}(\,\cdot\,)$ and $\tilde{V}_f(\,\cdot\,)$ into $\tilde{V}_N(\,\cdot\,)$ gives

$$\tilde{V}_N(x, \mathbf{u}, t) = V_N(x, \mathbf{u}, t) + \lambda(x, t) - V_N(x_r(t), \mathbf{u}_r(t), t).$$

Thus, for any x and t, the two objective functions $\tilde{V}(\,\cdot\,)$ and $V(\,\cdot\,)$ differ by additive constant, which means their optima occur for the same values of $\mathbf{u}$. Finally, since $\tilde{\kappa}_N(\,\cdot\,)$ is equivalent to $\kappa_N(\,\cdot\,)$, and since $\tilde{\kappa}_N(\,\cdot\,)$ stabilizes the reference trajectory, we conclude that $\kappa_N(\,\cdot\,)$ also stabilizes the reference trajectory. □

Based on Theorem 3.20, one can show that economic MPC stabilizes the reference trajectory by finding a valid storage function $\lambda(\,\cdot\,)$ for the chosen cost function. Various approaches exist for finding storage functions, e.g., appealing to strong duality of the *steady-state* optimization problem (Diehl et al., 2011), or by solving semidefinite programming problems (Zanon et al., 2017a). Note that any cost function can be made dissipative simply by adding a large enough penalty on $|x - x_r(t)|$, although at that point, the behavior is similar to tracking MPC. For more complex systems, especially with discrete actuators, finding a suitable $\lambda(\,\cdot\,)$ may not be possible, and thus the asymptotic performance result of Theorem 3.15 is likely to be of greater utility.

Remark 3.21: It can be shown (Risbeck and Rawlings, 2018b) that under the assumption of (non-strict) dissipativity and lower-boundedness of $\lambda(\,\cdot\,)$, that the performance bound of (3.8) can be strengthened to

$$\lim_{T\to\infty} \frac{1}{T} \sum_{k=t}^{t+T-1} \big(\ell(x(k), u(k), k) - \ell(x_r(k), u_r(k), k)\big) = 0.$$

That is, the cost will asymptotically equal the cost of the reference trajectory. It is also easy to show that the so-called "available storage function" (Muller et al., 2015)

$$S(x, t) \coloneqq \inf_{\substack{T \geq N \\ \mathbf{u} \in \mathcal{U}_T(x,t)}} V_T(x, \mathbf{u}, t) - V_T(x_r(t), \mathbf{u}_r, t)$$

is finite on $\mathcal{X}_N(t)$, then $\lambda(x, t) = -S(x, t)$ is a valid storage function for the system. This function provides a potential avenue for investigating possible dissipativity by extending the horizon and seeing if the optimal cost is

bounded. Note unfortunately that the available storage function will almost never provide strict dissipativity, as for any $x \in \mathcal{X}_N(t)$, there exists $u \in \mathbb{U}(t)$ such that

$$S(x,t) = \ell(x,u,t) - \ell(x_r(t),u_r(t),t) + S(f(x,u,t),t+1)$$

which means the dissipation inequality in Definition 3.17 is tight, and there is no slack for $\alpha(|x - x_r(t)|)$.

3.4 Peak Charges

In the formulation of economic MPC, it was assumed that the objective function can be written as a stagewise sum of costs $\ell(\,\cdot\,)$. Allowing the cost function to be time-varying can account for time-varying economic conditions that are becoming increasing prevalent in electricity markets (Albadi and El-Saadany, 2007), but this framework does not directly admit so-called "peak demand" charges that are assessed based on the maximum *rate* of electricity purchase (Berg and Savvides, 1983). These cost structures lead to an objective function that is written as the maximum value of some performance measure over some finite time period. These cost structures are common for industrial electricity customers and have been applied at times to residential customers (Taylor and Schwarz, 1990).

Due to the complex nature of this combination of cost functions, optimization techniques like economic MPC are a natural tool for decision making. These conditions can be optimized in real time to achieve lower closed-loop costs, for example by using energy storage techniques to reshape demand. Indeed, specific applications involving peak costs have been proposed in the literature (e.g., Cole et al. (2012); Jones and Peet (2017); Kumar et al. (2018); Ma et al. (2012a); Risbeck et al. (2017)), but there is generally no theoretical development to inform the closed-loop behavior of these methods. As pointed out in Scokaert and Rawlings (1999), naive inclusion of such max-over-time penalties can lead to pathological closed-loop behavior due to violation of the principle of optimality. To help shape the closed-loop response, previous peak values can be retained as part of the state so that the optimizer does not "forget" the peak so far. This modification restores the principle of optimality, thus allowing dynamic-programming-like techniques to be applied to find open-loop solutions Jones and Peet (2017). However, little attention has been devoted to the *closed-loop* properties of these price structures. Thus, we wish to develop a formulation of peak charges that is consistent with the economic MPC theory from the previous section.

3.4.1 Cost Structure

We consider peak charges that are assessed based on the maximum value of some scalar function over a given time window. We assume that the time windows are sequential and non-overlapping, but we do not require that they are uniform in length. This choice allows the treatment of, e.g., monthly demand charges, in which sequential windows

could be 31, 30, 29, or 28 days in duration. At the end of each window, the peak charge is assessed, and then the peak value resets for the next window.

To provide an explicit formulation, let $\mathcal{T}(t) \subset \mathbb{T}$ be the set of time points in the peak charge window that contains time $t \in \mathbb{T}$. We then let $\tau(t) \coloneqq 1 + \max \mathcal{T}(t)$ be the time point at which the peak charge is accessed. We denote as $h(\,\cdot\,) : \mathbb{X} \times \mathbb{U} \times \mathbb{T} \to \mathbb{R}_{\geq 0}$ the performance measure used to calculate the peak value. The economic cost is then calculated by $p(\,\cdot\,) : \mathbb{R}_{\geq 0} \to \mathbb{R}$. Thus, over a given time window $\mathcal{T}$, the system cost is given by

$$\sum_{t \in \mathcal{T}} \ell(x(t), u(t), t) + p\left(\max_{t \in \mathcal{T}} h(x(t), u(t), t)\right), \tag{3.12}$$

which is the sum of a standard stage cost $\ell(\,\cdot\,)$ with the additional peak cost $p(\,\cdot\,)$.

Due to the inclusion of the $\max$ term, the cost structure of (3.12) is not consistent with the economic MPC formulation from Section 3.3. Thus, we require a reformulation to apply economic MPC theory. As will be demonstrated in the next section, we can replace the explicit $\max$ over $\mathcal{T}$ with a recursive calculation that proceeds one time point at a time, thus fitting in to our economic MPC framework.

3.4.2 Augmented System Reformulate

We begin by defining the set

$$\mathbb{T}_w \coloneqq \{0\} \cup \{\tau(t) : t \in \mathbb{T}\}$$

of time points where the peak charge is assessed. We then define the binary coefficient

$$\sigma(t) \coloneqq \begin{cases} 1 & \text{if } t \in \mathbb{T}_w, \\ 0 & \text{otherwise.} \end{cases}$$

This parameter avoids the need for piecewise function definitions. To calculate $\max_{t \in \mathcal{T}} h(\,\cdot\,)$, we augment the system with an additional state variable y subject to dynamics

$$y^+ = \max((1 - \sigma(t))y,\ h(x, u, t)).$$

Within a given window $\mathcal{T}$, we have $\sigma(t) = 0$, and thus this function gives $y^+ = \max(y, h(x, u, t))$, calculating the new peak as the maximum of the previous peak and the current value of $h(\,\cdot\,)$. At the end of the window, we have $\sigma(t) = 1$, and thus $y^+ = h(x, u, t)$ is reset to be the current value of $h(\,\cdot\,)$ that starts the new peak charge window.

Defining the augmented state as $z \coloneqq (x, y)$, we thus have the following dynamics for the augmented system:

$$z^+ = \begin{pmatrix} x^+ \\ y^+ \end{pmatrix} \coloneqq F(z, u, t) = \begin{pmatrix} f(x, u, t) \\ \max((1 - \sigma(t))y,\ h(x, u, t)) \end{pmatrix}. \tag{3.13}$$

Similarly, the augmented cost function becomes

$$L(z, u, t) \coloneqq \ell(x, u, t) + \sigma(t)p(y), \tag{3.14}$$

which is the standard stage cost $\ell(x, u, t)$ for times $t \notin \mathbb{T}_w$ but includes the additional peak charge. With these augmentations, the standard cost function $\sum_{t\in\mathcal{T}} \ell(z(t), u(t), t)$ is equivalent to (3.12).

Remark 3.22: For simplicity, we have considered only a single nonoverlapping peak. Because $h(\,\cdot\,)$ is allowed to be time-varying, the formulation allows for certain time points to be excluded from the calculation. For example, an electricity demand charge may only be calculated over designated "peak" hours when the grid is congested. To account for such cases, one can simply set $h(\,\cdot\,,\,\cdot\,,t) = 0$ for time points t that are not subject to the peak. However, some systems may be subject to multiple peak charges assessed on different timescales, for example a monthly and a seasonal peak. Because these windows overlap, the above formulation is not directly applicable. Fortunately, these cases can be handled by adding an additional extra state y', y'', etc., for each separate peak. To avoid notational clutter we proceed under the assumption of only a single window for each time point, but the methods can be applied to these multi-window cases.

3.4.3 Terminal Set, Control Law, and Cost

With the peak charge now incorporated in a standard state-space formulation of economic MPC, standard techniques can be followed. In particular, a reference trajectory $(\mathbf{x}_r, \mathbf{u}_r)$ can be generated via optimization, and suitable terminal sets $\mathbb{X}_f(t)$ can be chosen. However, because the terminal cost decrease condition required by Assumption 3.13 now must consider the new cost augmentation, it is now even more difficult to synthesize the necessary $\mathbb{X}(t)$ and $V_f(\,\cdot\,)$. As remarked in the previous section, the simple choice of $\mathbb{X}_f(t) \coloneqq \{x_r(t)\}$ with $\kappa_f(x, t) \coloneqq \{u_r(t)\}$ and $V_f(x, t) \equiv 0$ does in fact satisfy Assumption 3.13. This choice leads to a constraint in the optimization problem that is equivalent to

$$\max\left(y, \max_{k\in\mathbb{I}_{[0:N)}} h(x(k), u(k), t+k)\right) = y_r(t+N).$$

In particular, this relationship implies that

$$h(x(k), u(k), t+k) \leq y_r(t+N), \quad k \in \mathbb{I}_{[0:N)},$$

i.e., that $y_r(t+N)$ is a hard upper bound on $h(\,\cdot\,)$. Thus, the feasible set $\mathcal{X}_N(t)$ is potentially significantly reduced compared to an equivalent system without a peak charge.

To avoid such reduction in the feasible space, we propose a more flexible terminal cost and constraint that is as easy to synthesize as the simple terminal constraint $\mathbb{X}_f(t) \coloneqq \{x_r(t)\}$. The terminal ingredients are chosen as follows:

$$\mathbb{X}_f(t) \coloneqq \{z = (x, y) \mid x = x_r(t)\}, \tag{3.15a}$$
$$\kappa_f(z, t) \coloneqq \{u_r(t)\}, \tag{3.15b}$$
$$V_f(z, t) \coloneqq p\left(\max(y, \check{y}_r(t))\right) - p(\hat{y}_r(t)). \tag{3.15c}$$

In particular the parameters in the definition of $V_f(\,\cdot\,)$ are the remaining reference peak $\check{y}_r(t)$ defined as

$$\check{y}_r(t) \coloneqq (1 - \sigma(t)) \max_{k \in \mathcal{T}(t), k \geq t} h(x_r(k), u_r(k), k), \tag{3.16}$$

and the reference peak $\hat{y}_r(t)$ of the current window given by

$$\hat{y}_r(t) \coloneqq \max_{k \in \mathcal{T}(t-1)} h(x_r(k), u_r(k), k). \tag{3.17}$$

We point out the time shift in (3.17) so that $\hat{y}_r(t) = y_r(t)$ gives the value of the peak that has just ended when $t \in \mathbb{T}_w$.

Intuitively, the terminal ingredients in (3.15) still require that the original states x terminate exactly on $x_r(t)$, but now the augmented state y is allowed to take any value. To prevent the optimizer from running up a large peak, the terminal value of y is assessed a cost of $p(\,\cdot\,)$, but critically the argument of $p(\,\cdot\,)$ includes the remaining peak of the reference trajectory in the current window. This extra term accounts for the fact that the system may still follow $x_r(t)$ for the remainder of the current window and thus incur its remaining peak; therefore, the optimizer is not rewarded for choosing a temporarily smaller peak. Note that the subtraction of $p(\hat{y}_r(t))$ is simply to enforce that $V_f(z_r(t), t) = 0$, as required by Assumption 3.13. For the chosen ingredients, we have the following proposition:

Proposition 3.23: *Suppose Assumption 3.12 holds. Then, the terminal set, control law, and cost defined in* (3.15) *satisfies Assumption 3.13.*

Proof. We note first that for a given terminal control law $\kappa_f(\,\cdot\,)$ under which $\mathbb{X}_f(t)$ is invariant, the sum

$$V_\infty^{\kappa_f}(x, t) = \sum_{k=t}^{\infty} \left(\ell(x(k), \kappa_f(x(k), k), k) - \ell(x_r(k), u_r(k), k)\right), \tag{3.18}$$

if finite, is a valid terminal cost, as the decrease of Assumption 3.13 is satisfied as an equality. The proof proceeds by establishing that $V_f(\,\cdot\,)$ as defined in (3.15) is precisely of the form of (3.18).

Recognizing that we are working with the extended state, we define the shifted cost functions

$$\begin{aligned}
\bar{L}(z, u, t) &\coloneqq L(z, u, t) - L(z_r(t), u_r(t), t), \\
\bar{\ell}(x, u, t) &\coloneqq \ell(x, u, t) - \ell(x_r(t), u_r(t), t) \\
\bar{p}(y, t) &\coloneqq p(y) - p(y_r(t)).
\end{aligned}$$

Next, we split (3.18) into two pieces as follows:

$$V_\infty^{\kappa_f}(z,t) = \sum_{\substack{k\in\mathcal{T}(t),\\ k\geq t}} \bar{L}(z(k),\kappa_f(z(k),k),k) + \sum_{k=\tau(t)}^{\infty} \bar{L}(z(k),\kappa_f(z(k),k),k).$$

For the first sum, we have

$$\sum_{\substack{k\in\mathcal{T}(t),\\ k\geq t}} \bar{L}(z(k),\kappa_f(z(k),k),k) = \sum_{\substack{k\in\mathcal{T}(t),\\ k\geq t}} \left(\bar{\ell}(x_r(k),u_r(k),k) + \sigma(k)\bar{p}(y(k),k)\right) = \sigma(t)\bar{p}(y(t),t),$$

because $\sigma(k) = 0$ for all k in the summation range except possibly for the first point t. For the second sum, we note that $z(k) \in \mathbb{X}_f(k)$, and therefore

$$\begin{aligned}\sum_{k=\tau(t)}^{\infty} \bar{L}(z(k),\kappa_f(z(k),k),k) &= \sum_{k=\tau(t)}^{\infty} \left(\bar{\ell}(x_r(k),u_r(k),k) + \sigma(k)\bar{p}(y(k),k)\right) \\ &= \sum_{k=\tau(t)}^{\infty} \sigma(k)\bar{p}(y(k),k) = \bar{p}(y(\tau(t)),\tau(t))\end{aligned}$$

Note that the last line follows because for $k = \tau(t)$, we have from (3.13) that

$$y(k+1) = h(x,u,k) = h(x_r(k),u_r(k),k) = y_r(k+1).$$

Therefore, $y(k) = y_r(k)$ for $k > \tau(t)$, and there is only the single nonzero term. Thus, we have

$$V_\infty^{\kappa_f}(z,t) = \sigma(t)\bar{p}(y(t),t) + \bar{p}(y(\tau(t)),\tau(t))$$

and we are nearly done.

To complete the proof, we examine $y(\tau(t))$. If $t \in \mathbb{T}_w$, then $\sigma(t) = 1$ and $y(\tau(t)) = y_r(\tau(t))$ because a new window has just begun. Thus,

$$V_\infty^{\kappa_f}(z,t) = \bar{p}(y(t),t) + \bar{p}(y_r(\tau(t)),\tau(t)) = \bar{p}(y(t),t) = p(y(t)) - p(y_r(t)) = V_f(z,t)$$

because $\check{y}_r(t) = 0$ and $\hat{y}_r(t) = y_r(t)$. Alternatively, if $t \notin \mathbb{T}_w$, then by composition of (3.13), we have that

$$\begin{aligned}y(\tau(t)) &= \max\left(y(t), \max_{k\in\{t,\ldots,\tau(t)-1\}} h(x(k),u(k),k)\right) \\ &= \max\left(y(t), \max_{k\in\mathcal{T}(t),k\geq t} h(x_r(k),u_r(k),k)\right) \\ &= \max(y(t),\check{y}_r(t)).\end{aligned}$$

Therefore,

$$V_\infty^{\kappa_f} = \bar{p}\left(\max(y(t),\check{y}_r(t)),\tau(t)\right) = p\left(\max(y(t),\check{y}_r(t))\right) - p(y_r(\tau(t))) = V_f(z,t)$$

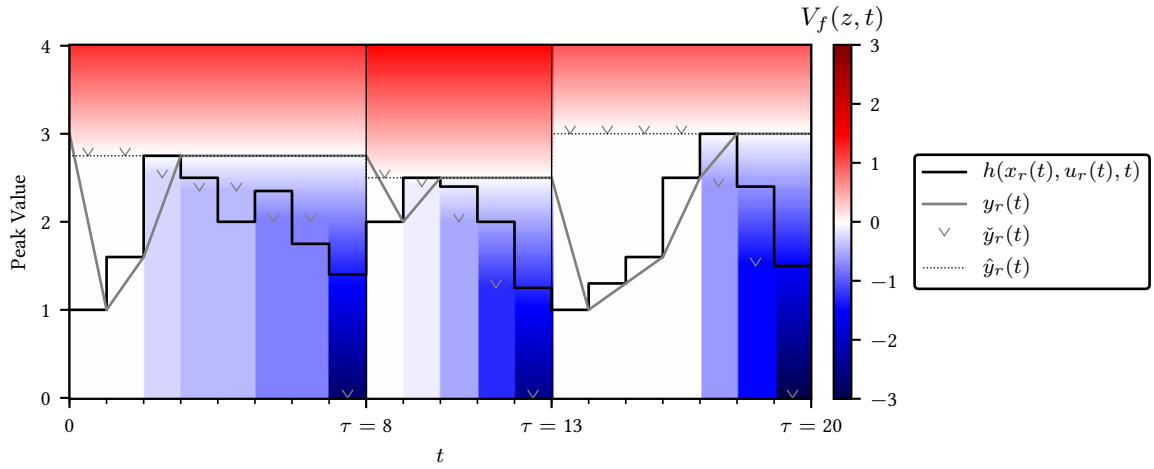


FIGURE 3.2: Illustration of $V_f(\,\cdot\,)$ and other paramters for the peak charge terminal cost. Colored bars show the value of $V_f(\,\cdot\,)$ for the time point to the right.

because $y_r(\tau(t)) = \hat{y}_r(t)$. Therefore, in either case, $V_f(z,t) = V_\infty^{\kappa_f}(z,t)$, and so $V_f(\,\cdot\,)$ is a valid terminal cost satisfying Assumption 3.13 as desired. □

While we have demonstrated that (3.15) defines a valid terminal region and control law, the structure of the terminal may still be unclear. We provide an illustration of $V_f(\,\cdot\,)$ in Figure 3.2. In this example, successive windows are 8, 5, and 7 time points in duration. For each time point, the cost function is increasing above $\hat{y}_r(t)$, as any exceedance of the nominal peak will eventually have to pay a higher peak cost. Below $\hat{y}_r(t)$, the terminal cost decreases until it reaches $\check{y}_r(t)$. This section reflects the fact that the peak of the reference trajectory has already passed and the system is below that value. Therefore, there is the possibility that the realized peak charge is below the reference value. However, below $\check{y}_r(t)$, $V_f(\,\cdot\,)$ is once again constant, because if the system decides to follow the reference trajectory for the *remainder* of the current window, then it will eventually experience a peak of $\check{y}_r(t)$. Thus, the optimizer should not be incentivized for coming in below this value, because the peak could potentially increase. Finally, we note that in the final period of each window, $\check{y}_r(t) = 0$, and so $V_f(\,\cdot\,)$ decreases all the way down to zero, as the system's terminal peak is exactly the level for which the peak charge is assessed. Thus, the terminal cost in (3.15) appropriately penalizes and rewards the system based on what remains in the current window.

REMARK 3.24: While the augmented formulation presented here fits within the framework of economic MPC, it is not particularly useful for optimization. Assuming, for example that $h(x,u,t) = u$, the equality constraint $y^+ = \max(y,u)$ is nonconvex and nonsmooth. However, if $p(\,\cdot\,)$ is convex and nondecreasing, the original penalty term

$$p\left(\max_{t\in\mathcal{T}} u(t)\right)$$

is in fact a convex function of the $u(t)$ variables. Thus, to actually implement this peak charge strategy in an optimization problem, one should define a single variable Y for each *window* rather than each time point, and include the constraints

$$Y \geq y, \quad Y \geq \check{y}_r, \quad Y \geq h(x(t), u(t), t) \text{ for } t \in \mathcal{T}$$

in which y is the current peak so far, and $\check{y}_r$ is the remaining reference peak in the current window $\mathcal{T}$. Including the term $\rho(Y)$ is equivalent to assessing the terminal cost for that window, as the optimizer will choose the smallest value of Y. The same strategy can also be applied for the charges $p(\,\cdot\,)$ in the middle of the horizon, simply omitting the constraint $Y \geq \check{y}_r$. Thus, if the system model $f(\,\cdot\,)$ is linear and the functions $\ell(\,\cdot\,)$ and $h(\,\cdot\,)$ are all convex, then this peak charge strategy can be implemented without destroying the convexity of the MPC optimization problem.

3.5 Illustrative Examples

3.5.1 Building Cooling

As a first example, we consider a small building cooling system. This system is essentially an extreme simplification of the application that will be discussed in Chapters 5 and 6. The system consists of a single building that is uniform in temperature and receives heat from the ambient. The goal is to maintain comfortable temperature levels, which is accomplished by running two small chiller units. Each chiller can be on or off, and if it is on, it must operate between specified (nonzero) minimum and maximum capacities. The objective function is to minimize the cost of electricity consumed in the chillers subject to strongly time-varying electricity prices. Thus, the system can save money by pre-cooling the building when electricity is cheap, which reduces future cooling requirements.

The system model is the discretization of the continuous-time model

$$m\frac{dT}{dt} = -k(T - T_{\text{amb}}) + q_{\text{amb}} - q.$$

Symbols are as follows:

- T: building temperature (state)
- T_{amb}: ambient outside temperature (time-varying parameter)
- q_{amb}: direct heating by the ambient (time-varying parameter)
- q: cooling by chillers (input)
- m: building mass (constant parameter)
- k: building heat transfer coefficient (constant parameter)

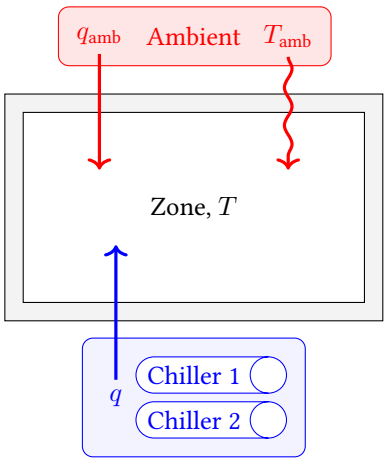


FIGURE 3.3: Diagram of simple building cooling system.

Using a constant sample time Δ, piecewise-constant holds on inputs, and piecewise-linear holds on time-varying parameters, a time-varying affine model of the form $f(x, u, t) = Ax + Bu + c(t)$ is obtained for the evolution of T. The system is illustrated in Figure 3.3

With N_{chiller} available chillers that can be either on or off, we add an additional discrete input variable v to choose how many chillers to activate. For each active chiller, its cooling duty must be within the range $[q_{\min}, q_{\max}]$, with $q_{\min} > 0$. Thus, we have the following input constraints for $u := (q, v) \in \mathbb{U}$:

$$\begin{aligned} q_{\min} v \leq q \leq q_{\max} v \\ v \in \{0, 1, \ldots, N_{\text{chiller}}\}. \end{aligned}$$

Note that q gives the *total* cooling provided by all chillers.

As a cost function, we consider minimization of electricity costs. Assuming a time-varying piecewise-constant price profile $\rho(t)$, the electricity cost for each period is given by $\rho(t)q$. In addition, we would like to enforce comfort constraints on the state T:

$$T_{\min}(t) \leq T(t) \leq T_{\max}(t).$$

However, to avoid infeasibility if the initial temperature happens to violate these bounds, we implement these bounds as soft constraints in the cost function. Thus, the economic cost function is

$$\ell_{\text{econ}}(x, u, t) = \rho(t)q + \sigma \max(0, T - T_{\max}(t)) + \sigma \max(0, T_{\min}(t) - T).$$

Numerical values of parameters are shown in Figure 3.4. The penalty coefficient $\sigma = 1000$ is chosen to be large relative to $\rho(t)$ so that temperature bound violations are avoided whenever possible.

To find a reference trajectory $(\mathbf{x}_r, \mathbf{u}_r)$, we solve an optimal control problem with the initial condition $x(0)$ free, but with the added constraint $x(0) = x(T)$ for a given period T. This gives a 24 h periodic cycle for $(\mathbf{x}_r, \mathbf{u}_r)$. We can

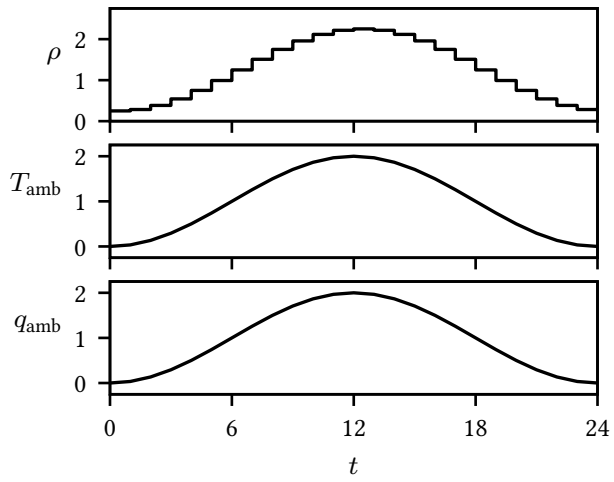


FIGURE 3.4: Values of time-varying parameters for the cooling example. All parameters are 24 hour periodic.

then track this sequence using tracking MPC, or optimize directly using economic MPC. For tracking MPC, we use the objective function

$$\ell_{\text{track}}(x,u,t) = |x - x_r(t)|^2 + 10\,|u - u_r(t)|^2 + \sigma \max(0, T - T_{\max}(t)) + \sigma \max(0, T_{\min}(t) - T).$$

That is, we use a standard quadratic tracking penalty, but we also include the cost penalty for violation of comfort bounds. The quadratic penalty on u is larger, as the economic cost is most closely associated with u.

For a terminal control law, we note that the system is open-loop stable, and so $\kappa_f(x,t) = \{u_r(t)\}$ does locally stabilize the system. However, determining a valid terminal cost is complicated by the presence of the soft penalties, in particular because $T_r(t) = T_{\min}(t)$ and $T_r(t) = T_{\max}(t)$ in some places. Thus, as a worst-case bound, we use the terminal cost

$$V_f(x,t) = \frac{1}{1-A^2}\,|x - x_r(t)|^2 + \frac{\sigma}{1-A}\,|x - x_r(t)|\,,$$

in which $A \approx 0.9$ is the time-invariant (scalar) state transition matrix. Here, the first term calculates the cost to go for the quadratic x penalty in $\ell_{\text{track}}(\,\cdot\,)$, and the second term is an upper bound on the cost to go for the soft temperature penalty. We note that under the terminal control law, $x^+ = Ax$ (in deviation variables $x - x_r(t)$). Thus, defining $x_{\min}(t)$ and $x_{\max}(t)$ as the deviation-variable values of $T_{\min}(t)$ and $T_{\max}(t)$, we have

$$\begin{aligned} V_f(x,t) - \ell(x,u,t) - V_f(x^+, t+1) &= \frac{1}{1-A^2}(x^2 - (Ax)^2) + \frac{\sigma}{1-A}(|x| - |Ax|) \\ &\quad - x^2 - \sigma \max(0, x - x_{\max}(t)) - \sigma \max(0, x_{\min}(t) - x) \\ &= \sigma\,|x| - \sigma \max(0, x - x_{\max}(t)) - \sigma \max(0, x_{\min}(t) - x) \\ &\geq 0 \end{aligned}$$

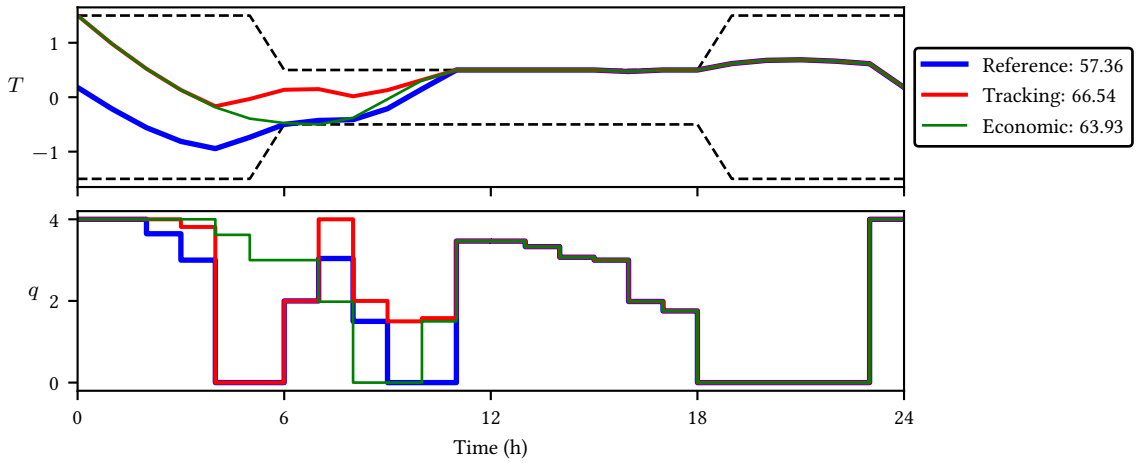


FIGURE 3.5: Closed-loop solutions to the simplified cooling problem under nominal conditions. Values in legend show closed-loop economic costs.

since the reference trajectory satisfies the soft constraints, and thus $x_{\max}(t) \geq 0$ and $x_{\min}(t) \leq 0$. Of course, using such a linear penalty as the terminal cost essentially ensures that x will terminate exactly on $x_r(t)$ whenever feasible, but the terminal cost is valid nonetheless. Since this bound is globally valid, we can take an arbitrarily large set for $\mathbb{X}_f(t)$.

To examine the nominal properties of tracking and economic MPC, we simulate the system under nominal conditions. In tracking MPC, we use the reference trajectory as a setpoint, while in economic MPC, we only use it as a terminal constraint. Figure 3.5 shows the closed-loop solution starting from a temperature higher than the reference. We notice that tracking MPC tries to balance deviation from the reference trajectory between x and u, and thus does not cool as aggressively as it should to achieve low cost. By contrast, economic MPC does not care about the reference value of u and thus pre-cools the building to take advantage of the lower electricity prices. This choice leads to roughly 5% cost improvement for economic MPC over tracking.

In real systems, it is unlikely that building models or forecasts of ambient conditions are accurate. Thus, we test the system under closed-loop tracking and economic MPC subject to disturbances. Note that both controllers experience the same disturbance sequences for corresponding initial conditions. Closed-loop trajectories are shown in Figure 3.6. Both controllers do experience soft temperature violations, but they are minimized when possible. Although there is not theory to tell us that the system should remain near the setpoint under economic MPC, performance is nevertheless satisfactory. To further evaluate each method, we show closed-loop economic costs and temperature deviations are shown in Table 3.1. From these values, we see that economic MPC now out-performs tracking MPC by 8% cost on average with roughly the same violation in soft temperature bounds. The difference is primarily because economic MPC is able to exploit "favorable" disturbances that reduce the necessary cooling; by contrast, tracking MPC tries to

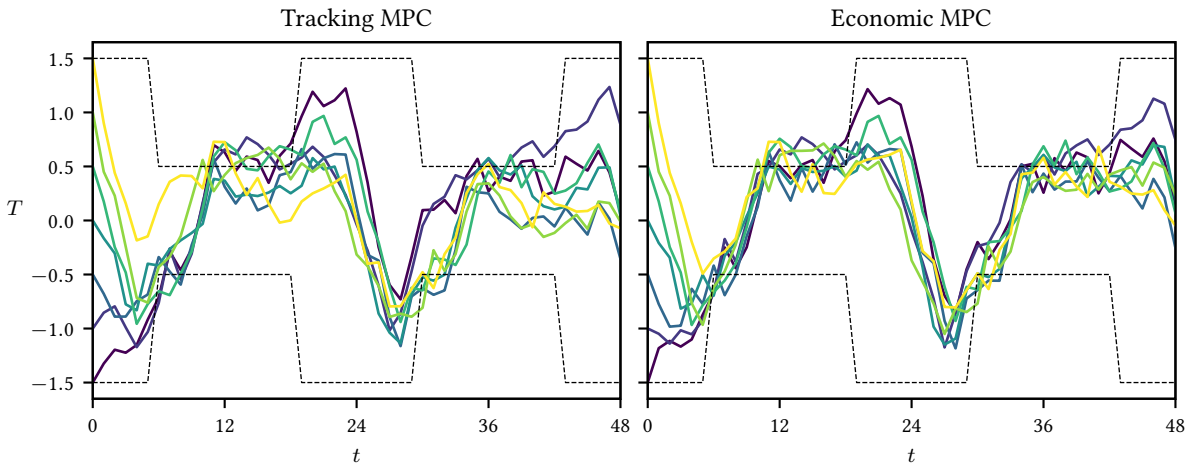


FIGURE 3.6: Closed-loop trajectories for the cooling example subject to disturabnces $|\mathbf{w}| \leq 0.25$.

TABLE 3.1: Closed-loop costs and temperature deviations for the cooling example subject to disturbances. Each cell gives electricity cost with total soft temperature bound violation in parentheses.

Starting T	Tracking MPC	Economic MPC
-1.5	133 (1.18)	128 (0.99)
-1	148 (2.05)	146 (2.06)
-0.5	113 (0.63)	97 (0.87)
0	124 (0.17)	112 (0.38)
0.5	126 (1.50)	123 (1.92)
1	136 (0.83)	119 (1.30)
1.5	140 (0.60)	123 (0.87)
Average	131 ± 11 (0.99 ± 0.58)	121 ± 14 (1.20 ± 0.56)

maintain the nominal level of cooling so as to minimize deviation from $\mathbf{u}_r$. Therefore, economic MPC performs better even when subjected to disturbances.

Finally, to verify the bound from Definition 3.2, we start the system on the reference trajectory and simulate closed-loop evolution under optimal tracking MPC subject to disturbances $|\mathbf{w}| \leq \omega$ for varying values of ω. Figure 3.7 shows the distribution of $x(t) - x_r(t)$ for these trials. From this figure, we see that the maximum deviation tends to increase as the size of the disturbance increases. This result is consistent with Definition 3.2 and Theorem 3.9, which prove that $|x(t)| \leq \sigma(\omega)$ for some $\mathcal{K}$ function $\sigma(\,\cdot\,)$. Note that there is no such guarantee for economic MPC, but results are generally consistent.

For simulation, the problem is formulated using CasADi (Andersson et al., 2018) via MPCTools (Risbeck and Rawlings, 2018a). Optimizations are performed using Gurobi (Gurobi Optimization, LLC, 2018).

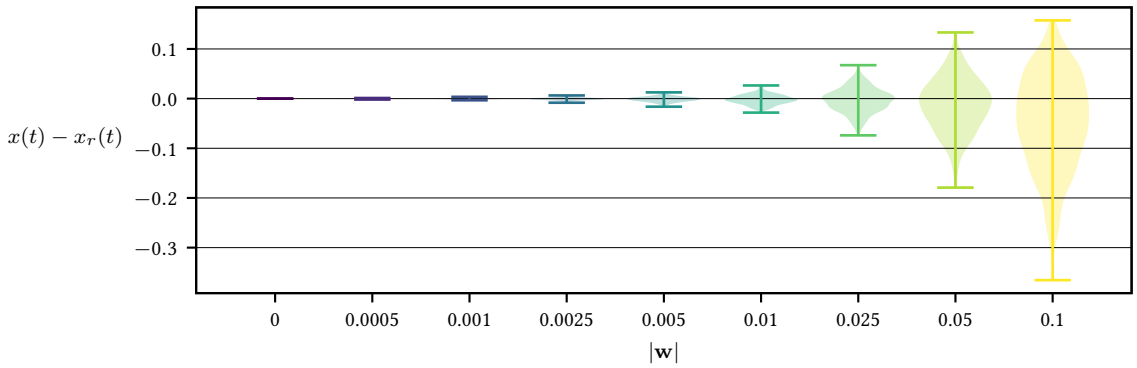


FIGURE 3.7: Violin plots of setpoint deviation for the simplified building cooling system. Bars show minimum and maximum values, while curves show a kernel density estimate from $T = 240$ simulation timesteps.

3.5.2 Energy Storage Optimization

We conclude with an energy storage optimization example from Risbeck and Rawlings (2018b). As in the previous example, the premise of this system is to minimize the electricity costs of running chillers to produce chilled water. The electricity pricing structure consists of time-varying use charges as before, but in addition, there is a peak charge assessed at the end of every month. In this case, we assume that the demand for chilled water is a known time-varying parameter, but the system has a large chilled water storage tank so as to temporally decouple the operation of chillers and the delivery of chilled water. This setup is essentially a simplification of the central energy plant optimization problem that is the focus of Chapter 5.

For the system model, the current level of the storage tank is given in the state $x \in [0, 500]$. The inputs are how much electricity to purchase, and how much chilled water to withdraw from the storage tank, which are subject to bounds $u \in [0, 500] \times [-500, 500]$. We assume that units have been nondimensionalized so that amounts of electricity and chilled water can be compared directly. With a timestep Δ, the system model and constraints are

$$x^+ = x - u_2\Delta, \qquad u_1 + u_2 \geq \delta(t).$$

We model the tank as a simple integrator so as to maximize the cost shifting that can occur. In a real system, the storage level would slowly decay as the chilled water heats up, which reduces the horizon needed to ensure optimal use of the storage tank. Over a given month consisting of time points $t \in \mathcal{T}$, the cost function is

$$\sum_{t\in\mathcal{T}} \rho(t)u_1(t)\Delta + \rho^{\text{peak}} \max_{t\in\mathcal{T}} u_1(t).$$

Applying the strategy from Section 3.4, we augment the model with an extra state y that evolves as $y^+ = \max(y, u_1)$ within each month to keep track of peak electricity purchase. The system is diagrammed in Figure 3.8, and the values of parameters $\rho(t)$ and $\delta(t)$ are shown in Figure 3.9. Note that these values are (rescaled) from real price and demand data provided by Johnson Controls.

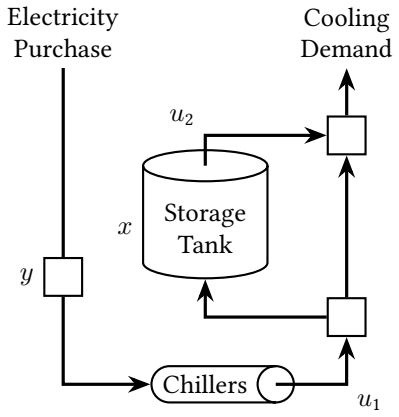


FIGURE 3.8: Diagram of energy storage optimization system. The extra state y is used to keep track the monthly maximum value of u_1.

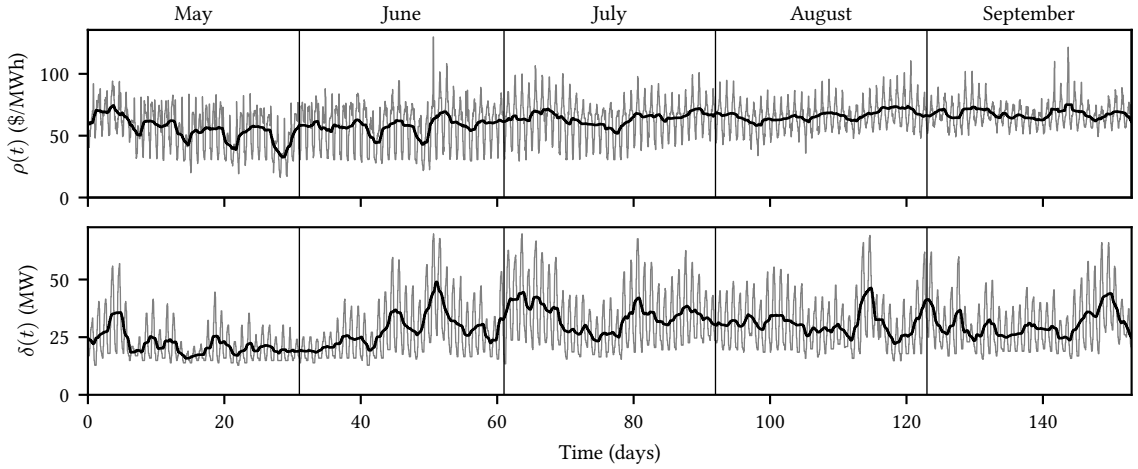


FIGURE 3.9: Electricity price and cooling demand parameters for energy storage optimization problem. Thick lines show 24 h moving average, while thin lines show hourly values.

To verify the closed-loop performance of the augmented peak charge system, we simulate the system under economic MPC and other heuristic closed-loop optimization strategies as follows:

- Scaled: the peak cost ρ^{peak} is scaled by N/T where N is the current prediction horizon, and T is the length of the current month.
- Naive: identical to Scaled except that the previous peak is not retained in each optimization (thus invalidating the principle of optimality).
- Increasing: the peak cost ρ^{peak} is scaled by $(t+N)/T$, thereby linearly increasing the weight throughout each month.
- Periodic: the optimal 5-month periodic cycle is calculated and used as a reference trajectory for economic MPC.
- Greedy: a 5-month solution is generated using a greedy strategy that optimizes each week in sequential, nonoverlapping optimization problems; the solution is then used as a reference trajectory for economic MPC.

Of these methods, the first three are heuristic strategies that do not use terminal constraints and provide no closed-loop guarantees, while the last two apply the economic MPC methods of Sections 3.3 and 3.4. Thus, the asymptotic performance of the Periodic and Greedy strategies is bounded by their corresponding reference trajectories, while anything could happen for the Scaled, Naive, and Increasing strategies. Each method uses a prediction horizon of one week with a 1 h timestep, and the storage tank begins empty in each case.

Closed-loop costs for each method are given in Table 3.2. All costs are given as percentages relative to the optimal solution, as obtained by solving a single large problem (with a horizon of 10 months to avoid any end effects). From this table, we see that the Naive and Scaled heuristics do not perform well, while the Increasing heuristic performs adequately despite not having any theoretical properties. By contrast, both economic MPC methods perform well. Illustrating Theorem 3.15, the Greedy strategy does asymptotically beat its reference trajectory, as the week-long horizon is sufficient to fill in the suboptimal gaps created by the greedy optimization strategy used to generate the trajectory. Meanwhile, the Periodic strategy actually achieves the optimal cost of the system. We note that the periodic reference actually beats the optimal trajectory in the month of May because it starts with a partially filled storage tank (due to the initial condition being optimized). Indeed, were the simulation extended, the cost of the Periodic strategy would slowly decay to the value of the periodic reference. Based on these values, we conjecture that the periodic reference is the long-term optimal solution, assuming 5-month periodicity of the system parameters.

To examine why each method performs as it does, we plot the daily peak use of electricity in Figure 3.10. The Naive strategy displays obvious bad behavior, as its daily peak use is non-monotonic. Because of the cost structure, once a peak has been reached, there is no incentive to stay below that value, and thus these values should be nondecreasing. However, because the Naive strategy does not remember its past peak, it continues to believe that it can "reset" the

TABLE 3.2: Cost performance for the energy storage optimization problem. Note that gray rows indicate reference trajectories used as terminal constraints for economic MPC strategies.

Method	May	June	July	August	September	Total
Naive	+48.63	+12.45	+27.00	+28.52	+1.10	+22.00
Scaled	+20.37	+9.76	+13.32	+20.29	−0.24	+12.21
Increasing	+0.29	+0.94	+0.11	+0.33	+0.18	+0.34
Greedy	−0.15	+0.77	+0.14	+0.23	+0.33	+0.27
Greedy Ref.	+0.65	+2.42	+0.29	+1.05	+0.08	+0.82
Periodic	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00
Periodic Ref.	−0.66	+0.00	+0.00	+0.00	+0.00	−0.09

peak and balance costs based on only future time periods. For both the Scaled and Naive strategies, we also notice very large spikes at the end of each month. Due to the demand charge weighting of these methods, in the final day of each month, the demand charge is significantly undervalued in the objective function. Thus the optimizer believes that it can purchase a very large amount of electricity in the current month, incurring only a very small peak costs, and then use those stored reserves to significantly reduce purchase (and thus peak charge) in the following month. Of course, as the subsequent month continues, these cost savings are not realized, and the end result is a higher than necessary demand charge.

Of the heuristic methods, the Increasing strategy performed best, primarily due to avoidance of the pathological behavior at the end of each month that is experienced by the other heuristic strategies. We see that the increasing weighting does lead to a stable peak throughout each month, but also means that said peak is chosen primarily based on demand at the beginning of the month. In cases like June when the average demand at the beginning of the month is very different from the end of the month, the peak is not properly chosen, and the system misses out on being able to exploit differences in time-varying prices. The two economic MPC strategies show similar behavior of flat peaks, as they use the parameter $\check{y}_r$ to forecast a reasonable value for for the remaining peak. Thus, they tend to better trade off use and demand charges throughout the month. Of course, the Periodic strategy's reference trajectory is generated using a very long prediction horizon, and thus it performs best as expected. In realistic systems, the optimization models are much more complex, and forecasts of price and demand are not as accurate, which means such long horizons are neither tractable nor informative. Thus, the Periodic strategy represents a best-case scenario. By contrast, the Greedy reference trajectory is generated only one week in advance of the closed-loop system, and thus it is a readily attainable trajectory with near-optimal performance. Thus, although the ideal scenario may not be realistic, the economic MPC strategies from this section do lead to improved closed-loop performance and can be applied in real systems.

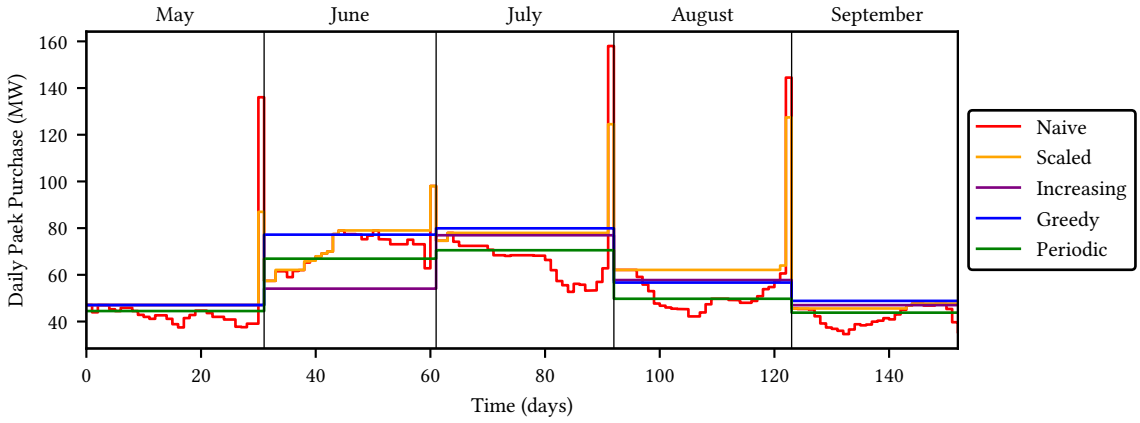


FIGURE 3.10: Evolution of monthly peak electricity use for energy storage optimization problem.

3.6 SUMMARY

In this chapter, we have presented two main extensions of tracking mixed-integer MPC. First, we have derived conditions under which tracking MPC is inherently robust to small state disturbances. Although the assumptions are slightly more restrictive than in the previous chapter, the main result essentially states that time-varying suboptimal tracking MPC with softened state constraints cannot be destabilized by arbitrarily small disturbances.

Second, we have discussed and proved asymptotic performance and stability theorems for economic MPC, in which the objective function is a tangible measure of system performance (such as economic profit), rather than a fictitious distance-from-setpoint measure as in tracking MPC. Under very mild assumptions, an arbitrary feasible reference trajectory can be used as a terminal constraint for economic MPC, and the system's asymptotic closed-loop performance will be no worse than that of the reference trajectory. We have also extended dissipativity-based stability theory to the time-varying setting. In addition, we have also presented an extended-state formulation to address peak charges within standard economic MPC. This reformulation is accompanied by a terminal set and cost function that avoids shrinking the feasible region for the system.

Finally, we have demonstrated robustness properties of tracking MPC and performance of economic MPC by means of two example systems, which are simplified versions of the problems addressed in detail in Chapter 6 and Chapter 5. These examples illustrate the benefits that can be achieved by applying MPC, and the extensions developed in this chapter serve to broaden the applicability of closed-loop online optimization techniques.

For a more extensive robustness result and an example of a robust system with a discontinuous optimal cost, see Allan et al. (2017). For more information about economic MPC and peak charges, including an example storage function, see Risbeck and Rawlings (2018b).

If once you start down the dark path, forever will it dominate your destiny. Consume [inventory] it will.

— Yoda
Star Wars: Episode V – The Empire Strikes Back

Chapter 4

Closed-Loop Scheduling

4.1 Introduction

In general, the problem of "scheduling" refers to the allocation of limited resources in order to complete a set of tasks. For example, in a chemical plant, the scheduling problem could be deciding which reactors to use to produce various products and when to produce them. Inherent in this problem class are many discrete decisions, such as unit/task assignments and whether to start a task at a given discrete time. As a result, finding optimal solutions to scheduling problems can be computationally demanding, although solution times are steadily decreasing as optimization methods improve.

Within the literature, there has been a significant amount of attention paid to the application of optimization methods for production scheduling (Harjunkoski et al., 2014; Méndez et al., 2006). Specific problems vary significantly, but scheduling applications and optimization models can generally be categorized based on the representation of time, the nature of material flow within the process, and the types of decisions made by the optimizer (Maravelias, 2012). Two classical formulations for production scheduling are the state-task network (STN), which was originally proposed by Kondili et al. (1993), as well as the slightly more general resource-task network, originally presented in Pantelides (1994). In the STN formulation, products and intermediate materials are treated abstractly as "states" that are produced and consumed by carrying out specific "tasks" in one or more units. The RTN formulation takes this approach a step further by unifying the states and units of the STN into a single category of "resources" that are produced and consumed over the course of finite-duration tasks. Both the STN and RTN are discrete-time formulations, and they

tend to perform at least as well as continuous-time counterparts (Méndez et al., 2006). Solution performance can also be improved through reformulations or model augmentations (Velez and Maravelias, 2013).

For our part, we wish to leverage the theoretical developments of Chapters 2 and 3 with respect to discrete-valued inputs in order to move optimization-based scheduling methods online using rolling-horizon strategies similar to MPC. This idea has already been applied to inventory control (Yi and Reklaitis, 2015) and supply chain management (Subramanian et al., 2014), and thus general production scheduling is a natural next venue. We start with some literature review of the scheduling aspects relevant to this work. We next illustrate how the closed-loop scheduling problem can be formulated as a state-space model via the classical problem of batch scheduling. Subsequent sections examine specific problems in more detail in addition to addressing the more general problem of integrated and scheduling and control.

4.1.1 Rescheduling and Closed-Loop Scheduling

For the most part, optimization formulations for scheduling problems are primarily concerned with the generation of a single schedule over a finite window of time. While efficient solution of this problem class is an important ingredient, a single schedule alone is insufficient for real use. Obviously, when the end of the current schedule is reached, it is necessary to generate a new schedule. In addition, when processes are subject to changes in internal characteristics (e.g., upsets and unit breakdowns) or external conditions (e.g., material costs and product demands), a schedule that was optimal at the beginning of the period may quickly become suboptimal or infeasible. Finally, due to the finite-horizon nature of most scheduling problems, open-loop optimal schedules illustrate the so-called "turnpike effect" (McKenzie, 1976) by which schedules deviate from the long-term optimal behavior. A possible manifestation is the optimizer becoming unnecessarily aggressive at the end of the horizon (e.g., by completely depleting stored inventory) by not considering potential ill effects that will realize later on. Because of these effects, inherent in every scheduling application is the process of "rescheduling," or determining a new schedule given updated system conditions.

To address need for rescheduling, two main classes of categories are reactive rescheduling and stochastic or robust scheduling (Li and Ierapetritou, 2008). In reactive rescheduling, new schedules are created by making a small set of modifications to the existing schedule in order to restore feasibility and/or optimality (Vin and Ierapetritou, 2000). These techniques are not applied at every step, but rather only after some specific triggering event has been realized (Mendez and Cerdá, 2004; Touretzky et al., 2017). Thus, the average optimization burden is low (as the incumbent schedule typically need not be revised), but when specific disturbances are realized, a modified or completely new schedule must be determined. The downside to these approaches is that they do not address the turnpike effect of schedules unless rescheduling is performed at regular intervals regardless of whether the system is operating

nominally. By contrast, the robust scheduling methods attempt to address potential disturbances *before* they realize by computing a-priori a schedule that is resilient to any realizations from a given uncertainty set (Sand et al., 2000; Shi and You, 2016). Such techniques can be computationally demanding, as they require the application of robust optimization techniques (Li and Floudas, 2014), but the extra effort can be beneficial when uncertainty is high. Less costly alternatives include the addition of nonzero safety stocks to hedge against disturbances without requiring full robust optimization (Eppen and Martin, 1988; You and Grossmann, 2008).

In contrast to these techniques, the idea of closed-loop scheduling is that a new optimal schedule should be computed at every time point regardless whether any disturbance has been observed. Schedules are thus implemented in a closed-loop fashion as employed by MPC. Of course, application of closed-loop scheduling requires that the optimization problems be tractable for online solution. One strategy to reduce the online computational burden is to solve the nominal scheduling problem as a parametric optimization problem, with the parameters being the relevant external influences (Kopanos and Pistikopoulos, 2014). Thus, the new optimal schedule can be obtained at each timestep via relatively quick evaluation of the parametric solution. However, problem sizes are limited in this case. It is demonstrated in Gupta and Maravelias (2016) and Gupta et al. (2016) that such closed-loop scheduling strategies generally lead to improved quality of the implemented schedule, compared to only rescheduling infrequently. Performance also tends to improve with faster rescheduling and longer horizons, although the dependence is non-monotonic. However, without careful attention to the closed-loop properties of the scheduling system, paradoxical or pathological closed-loop behavior can result Gupta and Maravelias (2016). It was first pointed out in Subramanian et al. (2012) that production scheduling problems can be put into state-space form as presented in Chapter 2. Thus, we would like to leverage this idea and apply insights from economic MPC theory to prevent such behavior.

4.1.2 Example: Batch Production

As a first example of closed-loop scheduling, we consider a simple model of profit maximization for batch production. We assume there are a set of tasks $i \in \mathbf{I}$, units $j \in \mathbf{J}$, and materials $k \in \mathbf{K}$, and further that each task/unit combination has fixed durations τ_{ik} and batch sizes β^+_{ijk}, β^-_{ijk} (for material production and consumption respectively). Assuming total profit is given by sales revenue based on prices π_k less the sum of inventory costs η_k and batch costs α_{ij}, one possible formulation is as follows:

$$\min \quad \sum_{t\in\mathbf{T}} \left(\sum_{i\in\mathbf{I}} \sum_{j\in\mathbf{J}} \alpha_{ij} W_{ijt} + \sum_{k\in\mathbf{K}} \left(\eta_k S_{kt} - \pi_k V_{kt}\right) \right), \tag{4.1a}$$

$$\text{s.t.} \quad S_{k(t+1)} = S_{kt} + \sum_{i\in\mathbf{I}} \sum_{j\in\mathbf{J}} \left(\beta^+_{ij} W_{ij(t-\tau_{ij})} - \beta^-_{ij} W_{ijt}\right) - V_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}, \tag{4.1b}$$

$$\sum_{i\in\mathbf{I}} \sum_{t'=t-\tau_{ij}+1}^{t} W_{ijt'} \leq 1, \quad j \in \mathbf{J}, t \in \mathbf{T}. \tag{4.1c}$$

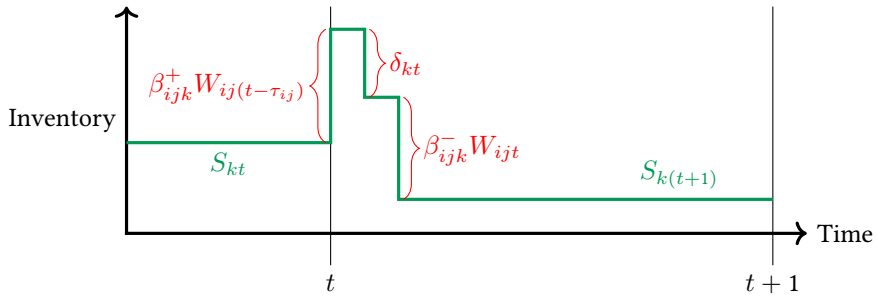


FIGURE 4.1: Diagram of inventory dynamics for simple batch scheduling model. Inventory costs are assessed based on the values S_{kt}.

Here, the variables $S_{kt} \in [0, \infty)$ are the inventory level at the beginning of time period t, $V_{kt} \in [V_{kt}^{\min}, V_{kt}^{\max}]$ are the sale amounts of resources k (subject to lower and upper bounds), and $W_{ijt} \in \{0, 1\}$ decide whether to start tasks in particular units. Note that we assume that at the beginning of each time period, production from completed batches is added before demand and consumption from incipient batches are subtracted as illustrated in Figure 4.1.

Because we wish to apply the techniques of Chapter 3 to this system, we have to determine a state-space model. On a first glance, the material inventories S_{kt} appear to be states, while the V_{kt} and W_{ijt} are inputs, with (4.1b) giving the dynamic model, and (4.1c) defining input constraints. Unfortunately, the constraint (4.1c) is *not* of the form $u(t) \in \mathbb{U}(t)$, but rather something more akin to $(u(t - \tau_{ij} + 1), \dots, u(t)) \in \mathcal{U}$ due to the constraints being enforced across time points. However, viewing (4.1c) as a modified rate-of-change constraint, we note that we can apply the standard technique (Rawlings et al., 2017b, Exercise 1.25) of augmenting the state to remember past inputs. In particular, we add new state variables $X_{ijnt} \in \{0, 1\}$ such that $X_{ijnt} = W_{ij(t-n)}$ for $n \in \{1, \dots, \tau_{ij}\}$. This so-called "lifting" of past inputs (Subramanian et al., 2012) allows the constraints and model to be expressed in state-space form. In particular, (4.1c) is rewritten as

$$W_{ijt} \leq 1 - \sum_{i \in \mathbf{I}} \sum_{n=1}^{\tau_{ij}-1} X_{ijnt}, \quad j \in Js, t \in \mathbf{T}, \tag{4.2}$$

and the dynamic model (4.1b) as

$$S_{k(t+1)} = S_{kt} + \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}} \left(\beta^+_{ij} X_{ij\tau_{ij}t} - \beta^-_{ij} W_{ijt} \right) - V_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}. \tag{4.3}$$

The evolution of the lifted inputs is then

$$X_{ijn(t+1)} = \begin{cases} W_{ijt}, & n = 1, \\ X_{ij(n-1)t}, & \text{otherwise.} \end{cases} \tag{4.4}$$

Thus, for state vector $x(t) = (S_{kt}, \dots, X_{ijnt}, \dots)$ and input vector $u(t) = (W_{ijt}, \dots, V_{kt}, \dots)$, the system model $f(\cdot)$ is given by (4.3) and (4.4), with state/input constraints $\mathbb{Z}(t)$ from (4.2) and stage cost

$$\ell(x, u, t) = \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}} \alpha_{ij} W_{ijt} + \sum_{k \in \mathbf{K}} \left(\eta_k S_{kt} - \pi_k V_{kt} \right).$$

In this form, we can consider application of the economic MPC strategies from Section 3.3. As mentioned in Subramanian et al. (2012) same strategy can be applied to general STN and RTN formulations to include features such as variable batchsizes, shared resource constraints, etc., that are common in scheduling models.

Of particular interest throughout this chapter is the application of terminal constraints to closed-loop scheduling problems. The simplest form of terminal constraints is to choose a reference trajectory $(x_r(t), u_r(t))$ and use the singleton sets $\mathbb{X}_f(t) \coloneqq \{x_r(t)\}$. Based on the state-space formulation of (4.1), the requirement that $x(T) = x_r(T)$ becomes

$$\begin{aligned} S_{kT} &= S^r_{kT}, \quad k \in \mathbf{K}, \\ W_{ijt'} &= W^r_{ijt'}, \quad i \in \mathbf{I}, j \in \mathbf{J}, t \in \{T - \tau_{ij}, \dots, T-1\}, \end{aligned}$$

in which S^r_{kt} and W^r_{ijt} give the values of those states in reference trajectory. Although these constraints are fairly restrictive (as they essentially fix the value of $u(t)$ in the final τ_{ij} time points), they ensure nominal recursive feasibility and the asymptotic performance bound of Theorem 3.15.

4.1.3 Integration of Scheduling and Control

Within most scheduling formulations, it is assumed that tasks can be represented a a finite-duration event, during which resource consumption is known. For example, in the RTN formulation, a changeover from Product A to Product B is modeled as a task that consumes a fictitious "Reactor in Product A State" resource at the beginning and then produces a corresponding "Reactor in Product B State" resource at the end. These abstractions are sufficient for bookkeeping, but they ignore the fact that the tasks actually represent the dynamic evolution of a unit and they are often implemented by a control layer. Because of the flexibility afforded by MPC, there is now increasing interest in integration of optimization-based scheduling and control. Traditionally, these layers have been considered separate for both historical and practical reasons (Harjunkoski et al., 2009). However, when the two systems are considered together, additional degrees of freedom in the control layer can lead to more efficient transitions, while the inclusion of unit dynamic models can prevent infeasibilities due to invalid operational assumptions.

Within the literature, various optimization strategies have been proposed to address combined scheduling and control. In Flores-Tlacuahuac and Grossmann (2006), a slot-based continuous-time formulation of the integrated scheduling and dynamic control problem was presented assuming a cyclic production schedule. By using collocation to approximate unit dynamic trajectories, the overall problem becomes a large mixed-integer nonlinear programming (MINLP) problem. A similar formulation using a discrete-time scheduling representation was presented in Nie et al. (2015). The addition of an additional planning layer on top was also presented in Gutiérrez-Limón et al. (2014). In all cases, direct solution of the integrated MINLP was challenging, and so the authors proposed various optimization

and decomposition strategies. To alleviate some of the computational burden, Du et al. (2015) propose replacing the full-order nonlinear dynamic equipment model with low-order linear approximations. The key insight here is that the equipment models are for the *closed-loop* behavior of the control system so that the decision variables are setpoints sent to the regulation layer, rather than the dynamic inputs themselves. However, while this simplification does lead to faster solution times, it also removes a potential source of additional flexibility. Other strategies and problem formulations can be found in Harjunkoski et al. (2009) and Engell and Harjunkoski (2012).

As in the scheduling-only case, most formulations are consider only the determination of open-loop solutions and do not discuss closed-loop implementations. Some exceptions are Zhuge and Ierapetritou (2012), which advocates re-optimization when deviation from the nominal trajectory becomes sufficiently large and presents some initial (though incomplete) theoretical closed-loop properties, and Chu and You (2012), which proposes replacing the full dynamic models by a set of candidate closed-loop trajectories (along with some other optimization strategies) to enable sufficiently fast solution. Addressing uncertainty in these problems remains a challenge, and is generally treated with similar classes of methods as in the scheduling-only problem (Dias and Ierapetritou, 2016). For our part, we would like to facilitate a closed-loop implementation of the combined problem, as will be presented in Sections 4.3 and 4.5.

4.2 Case Study: Batch Production

4.2.1 System Description

We begin with a simple batch production in the form of (4.1). The system consists of two units, Reactors 1 and 2, which perform the "Produce A" and "Produce B" tasks respectively. Dropping the j index (since it is identical to i), we have sets $\mathbf{I} = \{1, 2\}$, $\mathbf{K} = \{A, B\}$. A simple diagram with parameter values is shown in Figure 4.2. Note that the negative value of π_A indicates that the system has to pay a cost to dispose of A. Following (4.1), we have states and inputs

$$x \coloneqq \begin{pmatrix} X_{11} \\ X_{21} \\ X_{22} \\ S_A \\ S_B \end{pmatrix}, \qquad u \coloneqq \begin{pmatrix} W_1 \\ W_2 \\ V_A \\ V_B \end{pmatrix},$$

with a linear model and cost functions

$$f(x,u) = Ax + Bu, \qquad A \coloneqq \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 1 \end{pmatrix}, \qquad B \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\sqrt{2} & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

$$\ell(x,u) = qx + ru, \qquad q \coloneqq \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \qquad r \coloneqq \begin{pmatrix} 10 & 10 & 100 & -100 \end{pmatrix}$$

Thus, we can solve each optimization as an MILP.

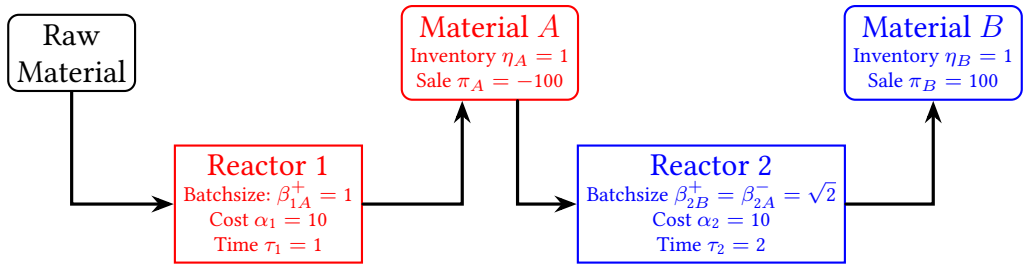


FIGURE 4.2: Diagram of simple batch production example. Values of β^+_{ik} and β^-_{ik} are zero unless specified.

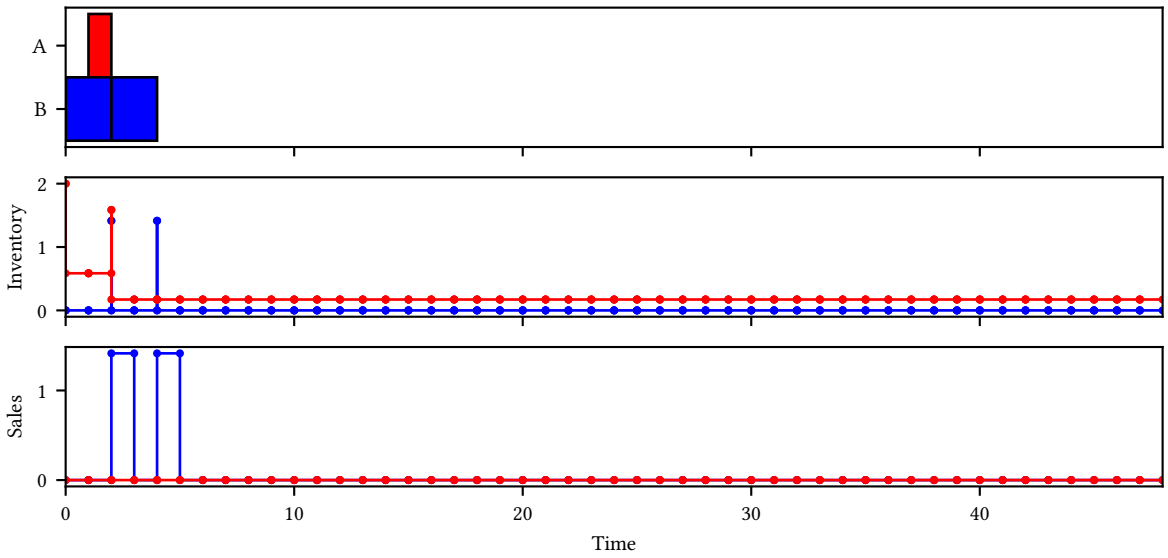


FIGURE 4.3: Closed-loop schedule for simple batch example using horizon $N = 8$ without terminal constraints. After $t = 4$, the system remains idle indefinitely.

4.2.2 NOMINAL CLOSED-LOOP BEHAVIOR

To start, we simulate a nominal closed-loop scheduling strategy with a horizon of $N = 8$. The system starts with 2 units of inventory for material A, and the *optimal* schedule is found at each timestep. The obtained closed-loop schedule is shown in Figure 4.3. We note that there are no missing tasks in this figure. The system is in fact idle after the three shown batches, and asymptotically the system *loses* money by paying inventory costs for material A without producing (or generating any revenue from) material B. We note also that this pathological behavior is not due to the horizon being too small. Increasing the horizon to $N = 32$, we obtain the closed-loop solution shown in Figure 4.4. Although the system does run longer than with the shorter horizon, the ending behavior is the same, and the system asymptotically loses money. Paradoxically, if we use a *shorter* horizon of $N = 5$, then we obtain the non-idle closed-loop behavior in Figure 4.5.

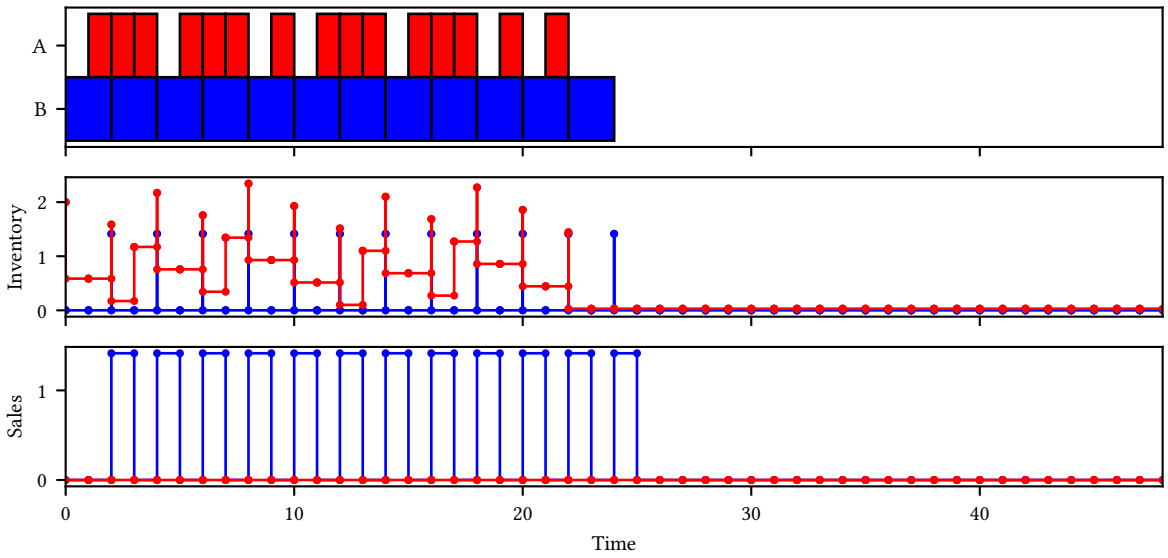


FIGURE 4.4: Closed-loop schedule for simple batch example using horizon $N = 32$ without terminal constraints. After $t = 25$, the system remains idle indefinitely.

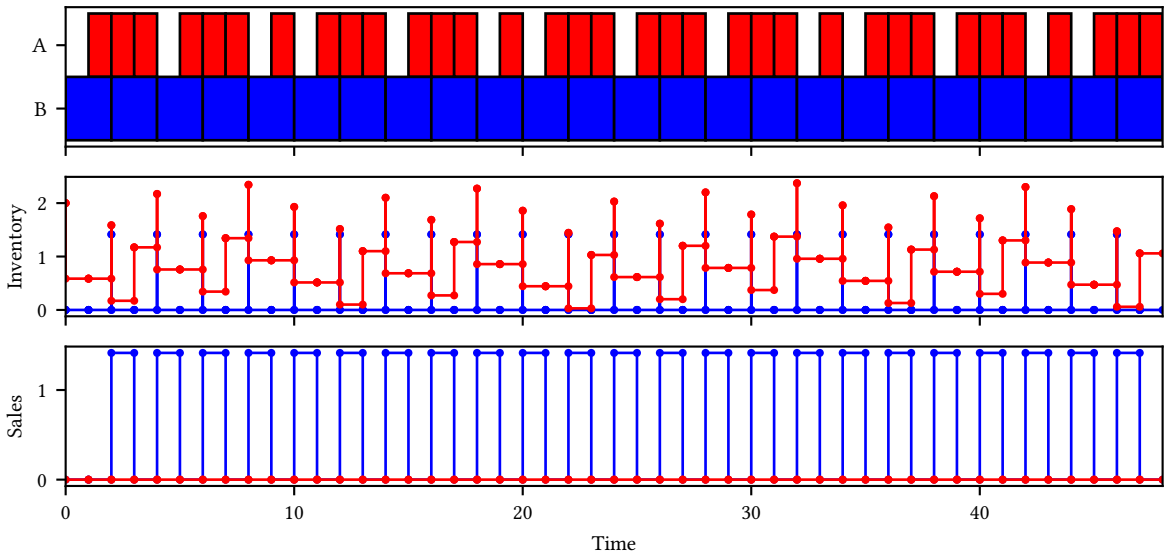


FIGURE 4.5: Closed-loop schedule for simple batch example using horizon $N = 5$ without terminal constraints. Unlike the two previous cases, the system does not become idle.

To examine *why* this behavior occurs for the system, we look at the open-loop predictions for each optimization. Figure 4.6 shows the evolution of the optimal schedule at each timestep. We see that the pathological behavior begins at time $t = 24$ due to the presence of an idle period at the beginning of the horizon. Because the optimal input takes no action in the first period, the state of the system is unchanged. Therefore, when the horizon is shifted forward and the optimization is repeated, exactly the same optimal solution is obtained. Thus, in the closed-loop solution, production is indefinitely postponed. Now, why does this occur with horizons of 8 and 32 but not 5? Assuming the system's initial inventory of A is below $\sqrt{2} - 1$, the system requires at least 5 timesteps to produce and sell a complete batch of B: two steps to produce 2 units of A, two more steps to run a batch of B, and a final step to sell the resulting product. Of course, each additional batch of B requires two additional time points. Therefore, if the horizon is an even number larger than 5, then there is an "extra" time period in which no additional batch of B can be processed. Since the system would have to pay inventory on any produced in this period, the optimizer does not run a batch of A in this period either, and thus the system is completely idle. The optimal location for this idle time period is wherever the inventory of A is lowest (as the value of $\ell(\,\cdot\,)$ during the idle period is minimized). If the initial inventory of A is very low, then the idle period is placed at the beginning in the open-loop optimal schedule, and the system never recovers. This observation leads to the following conjecture:

Conjecture 4.1: *For the batch production system in Figure 4.2, if the scheduling horizon is an even number, then for any initial inventory levels, the system is asymptotically idle under nominal closed-loop operation.*

To justify this conjecture, we note that, for all $t \in \mathbf{T}$ the inventory levels satisfy

$$S_{At} = S_{A0} + m_t - p_t\sqrt{2}, \quad m_t \coloneqq \sum_{t'=0}^{t-1} W_{1t'},\ p_t \coloneqq \sum_{t'=0}^{t-1} W_{2t'}, \tag{4.5}$$

in which m_t and p_t are integers. Thus, when the system is non-idle, the inventory levels S_{At} are unique (suppose $S_{At_1} = S_{At_2}$ for $t_1 \neq t_2$ with at least one batch between t_1 and t_2; we have $(m_{t_1} - m_{t_2}) - (p_{t_1} - p_{t_2})\sqrt{2}$, and at most one coefficient is zero; thus, we find either that a nonzero integer equals zero or that $\sqrt{2}$ is a rational number, both of which are contradictions). Therefore, for any finite prediction horizon N, it is only a matter of time before the *initial* inventory levels happen to be lowest. For even horizons, this property leads to idleness.

However, it is possible that once the horizon is sufficiently long, there are some inventory levels for which it is optimal to pay the penalty to dispose of some small amount of A. Based on the chosen cost coefficients, if a given amount of A cannot be used within 100 timesteps, then it is cheaper to just dispose of it. So, for horizons larger than 100, it is possible that the optimizer invokes this option, thus invalidating the relationship (4.5) for S_{At} and potentially avoiding idleness.

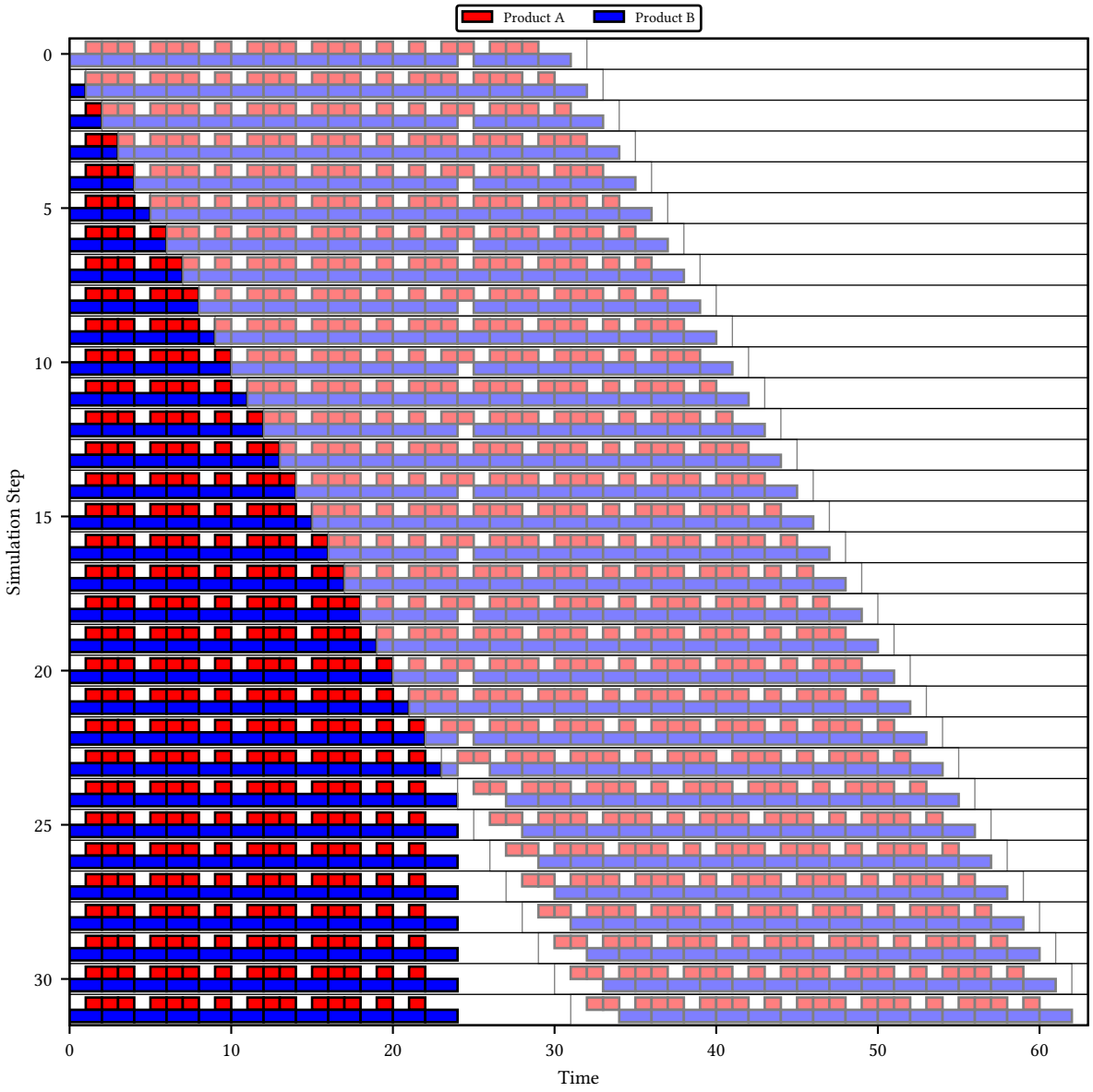


FIGURE 4.6: Closed-loop evolution of optimal solutions for the simple batch example with horizon $N = 32$. Shaded boxes show current optimal open-loop schedule.

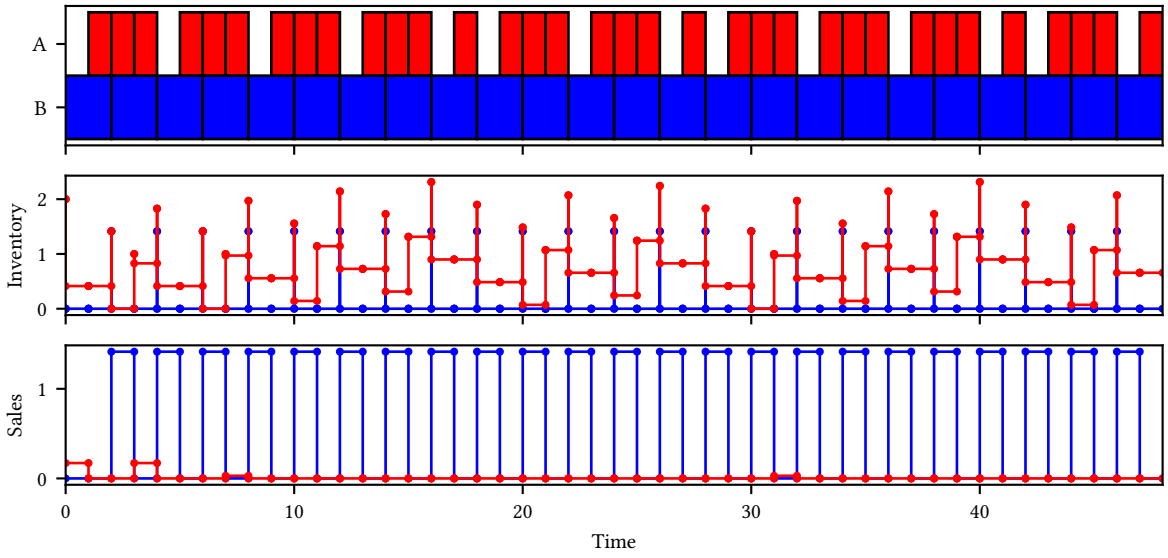


FIGURE 4.7: Closed-loop schedule for simple batch example with horizon $N = 8$ using the optimal $N = 24$ periodic cycle as a terminal constraint. In the first few periods, the system has to dispose of small amounts of A to match the phase of the terminal constraint.

4.2.3 Economic MPC Strategies

Although it is possible to avoid pathological closed-loop behavior for this example by using an odd-numbered horizon, we would instead like to exclude such behavior a-priori by applying the economic MPC techniques discussed in Section 3.3. To start, we begin by adding a terminal constraint to the system using a periodic reference trajectory. Because the system is time-invariant, there is no natural period to consider. However, by checking various periods, we find that the $N = 24$ periodic solution achieves high cost. From Theorem 3.15, we know that the asymptotic performance of the system can be no worse than that of the reference trajectory. Thus, the system cannot become idle. Using this solution as an exact terminal constraint, we obtain the closed-loop trajectory in Figure 4.7 with open-loop Gantt charts shown in Figure 4.8. We see from this figure that the plant does not become idle, and in fact it converges to the periodic reference trajectory within a few timesteps.

While the periodic terminal constraint does avoid pathological closed-loop behavior, the resulting closed-loop solution is somewhat undesirable due to the fact that the system has to dispose of small amounts of material A, both at the beginning to reach the terminal region, and throughout operation to achieve true periodicity. Fortunately, we note that there is no requirement that the reference trajectory is periodic, and thus we can consider generation of non-periodic terminal constraints. For this strategy, we construct a reference trajectory via application of the following heuristic control law:

- Start a batch of A if there is less than $\sqrt{2}$ inventory of A (Less than $\sqrt{2} - 1$ if a batch of B was just started)

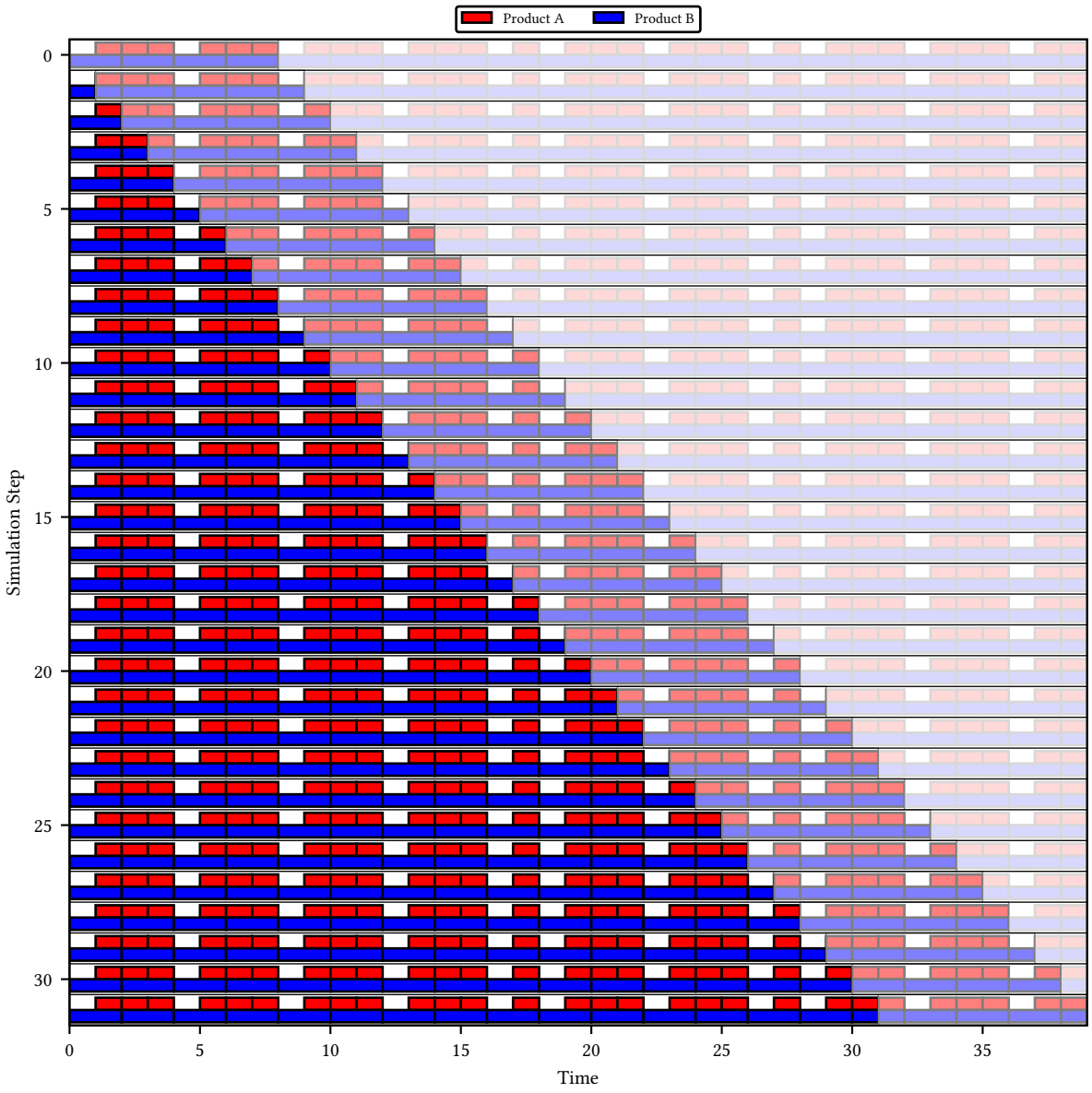


FIGURE 4.8: Closed-loop evolution of optimal solutions for the simple batch example with horizon $N = 8$ using a periodic reference trajectory as a terminal constraint. Shaded boxes show current optimal open-loop schedule, while light boxes show periodic reference.

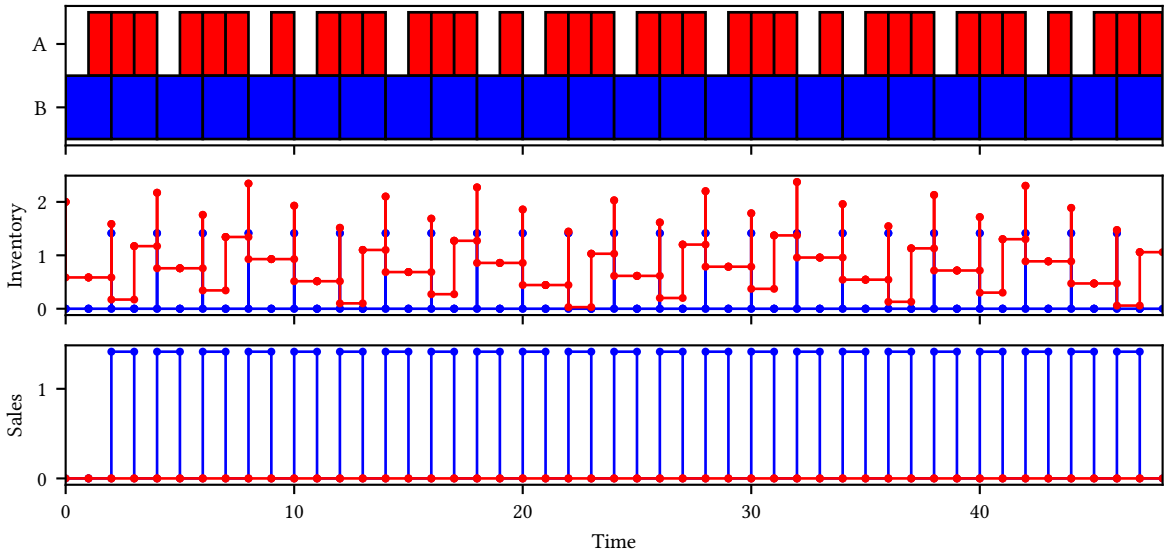


FIGURE 4.9: Closed-loop schedule for simple batch example using horizon $N = 8$ using heuristically-generated reference trajectory. The system does not dispose of any A.

- Start a batch of B if Reactor 2 is idle and there is at least $\sqrt{2}$ inventory of A
- Do not dispose of any A
- Sell all B

Applying this heuristic controller starting from the system's initial state, we generate a completely time-varying reference trajectory that can be used as a terminal constraint. With this reference, we use economic MPC to obtain the closed-loop schedule in Figure 4.9. With this terminal constraint, the system no longer needs to dispose of any A.

Although we have solved the issue of disposal of A, the exact terminal constraint does not provide very much flexibility to the system. Indeed, as observed in Figure 4.8, the system attaches to the reference trajectory within the first prediction horizon, and then no further changes are made to the closed-loop system. To provide additional flexibility, we wish to use a terminal constraint that provides a nontrivial range for the inventory levels S_A and S_B. As mentioned in Chapter 3, using any feasible terminal control law $\kappa_f(\cdot)$, a valid terminal cost is

$$V_\infty^{\kappa_f}(x,t) := \sum_{k=t}^{\infty} \big(\ell(x(k),\kappa_f(x(k))) - \ell(x_r(k),u_r(k))\big), \qquad x^+ = f(x,\kappa_f(x)),$$

assuming that the sum converges. Using the heuristic control law from above as $\kappa_f(\cdot)$, the above function is a valid terminal cost. However, because of the logical conditions in the heuristic rule, the resulting $V_f(\cdot)$ is highly discontinuous and thus it is not possible to embed within an optimization problem. Instead, we consider a terminal control law designed to out-perform an $N = 2$ periodic solution. The difficulty in this system is that if S_A is penalized too much in the terminal cost, then the optimizer will dispose of A early in the horizon, which leads to suboptimal

performance. Thus, we add a buffer inventory to the reference trajectory to establish a (suboptimal) baseline cost that the optimizer can then beat. We take

$$x_r(0) \coloneqq \begin{pmatrix} 1 \\ 0 \\ 1 \\ \sqrt{2} \\ 0 \end{pmatrix}, \quad u_r(0) \coloneqq \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \kappa_f(x,0) \coloneqq \begin{pmatrix} 1 \\ 1 \\ 0 \\ S_B \end{pmatrix},$$
$$\mathbb{X}_f(0) \coloneqq \{x \in \mathbb{X} : X_{11} = 1, X_{21} = 0, X_{22} = 1, S_A \in [\sqrt{2}-1, \sqrt{2}]\}, \quad V_f(x,0) \coloneqq -99 S_B$$

and

$$x_r(1) \coloneqq \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad u_r(1) \coloneqq \begin{pmatrix} 1 \\ 0 \\ 2-\sqrt{2} \\ 0 \end{pmatrix}, \quad \kappa_f(x,0) \coloneqq \begin{pmatrix} 1 \\ 0 \\ 2-\sqrt{2} \\ S_B \end{pmatrix},$$
$$\mathbb{X}_f(1) \coloneqq \{x \in \mathbb{X} : X_{11} = 1, X_{21} = 1, X_{22} = 0, S_A \in [0,1]\}, \quad V_f(x,1) \coloneqq -99 S_B$$

as terminal ingredients. These choices require the discrete states X_{int} to terminate exactly on the reference trajectory, but the inventories S_{kt} are more flexible.

To verify that the proposed terminal sets and constraints are valid, we have for $x \in \mathbb{X}_f(0)$,

$$\begin{aligned} V_f(x^+,1) + \ell(x,u,0) - \ell(x_r(0),u_r(0),0) &= 0 + (S_A - \sqrt{2}) + S_B - 100 S_B \\ &\leq (\sqrt{2} - \sqrt{2}) - 99 S_B = -99 S_B = V_f(x,0) \end{aligned}$$

by noting that $S_A \leq \sqrt{2}$ from the terminal constraint. Similarly, for $x \in \mathbb{X}_f(1)$, we find

$$\begin{aligned} V_f(x^+,0) + \ell(x,u,1) - \ell(x_r(1),u_r(1),1) &= 0 + (S_A - 1) + S_B - 100 S_B \\ &\leq (1-1) - 99 S_B = -99 S_B = V_f(x,0) \end{aligned}$$

using the bound $S_A \leq 1$. Thus, these choices satisfy Assumption 3.13 and are valid. Simulating with these terminal ingredients, we obtain the closed-loop solution shown in Figure 4.10. Although the asymptotic bound is different, we obtain the same solution as using the heuristic trajectory (shown in Figure 4.9).

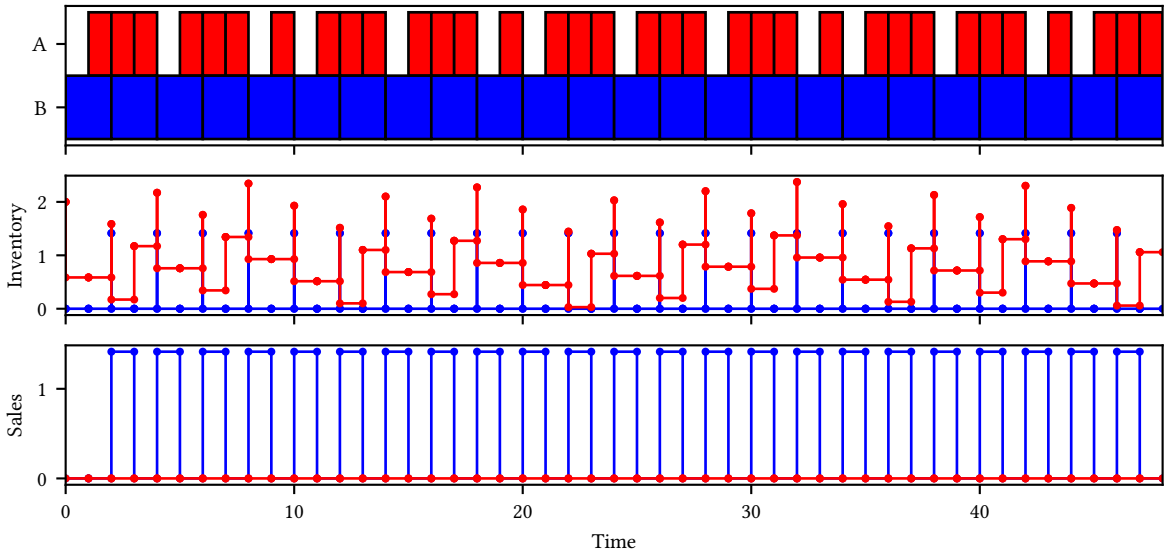


FIGURE 4.10: Closed-loop schedule for simple batch example using horizon $N = 8$ using the flexible terminal region and terminal cost. The solution is identical Figure 4.9.

4.2.4 TRANSIENT AND ASYMPTOTIC COST

To conclude this example, we examine the closed-loop costs of the various strategies. Without terminal constraints, there are no theoretical guarantees on transient or asymptotic cost. When terminal constraints are used, we have the transient cost bound

$$\frac{1}{T}\sum_{k=t}^{t+T-1}\left(\ell(x(k),u(k)) - \ell(x_r(k),u_r(k))\right) \leq \frac{V_N^*(x(t)) - V_N^*(x(t+T))}{T}$$

obtained from the proof of Theorem 3.15. Since both terms on the right-hand side are bounded, taking $T \to \infty$ gives the asymptotic bound of Theorem 3.15.

For each of the six closed-loop cases discussed so far, Table 4.1 closed-loop profit (i.e., negative closed-loop cost) for $T = 240$, along with transient and asymptotic bounds. From this table, we see that the three best-performing strategies are the $N = 5$ Nominal, $N = 8$ Heuristic, and $N = 8$ Terminal Cost. With $N = 5$ Nominal, we find a packed closed-loop schedule that does not dispose of any A, which is likely the globally optimal solution; however, we have no theory to know a-priori that this behavior should be achieved. For the $N = 8$ heuristic strategy, the closed-loop system follows the reference exactly and achieves the same closed-loop cost, although the constraint is very tight and would likely lead to suboptimal performance in the presence of disturbancess. The more flexible $N = 8$ Terminal Cost achieves the the same closed-loop schedule, but the bounds are not very tight due to the suboptimal reference trajectory. The $N = 8$ Periodic strategy leads to slightly lower closed-loop and asymptotic profit (due to the sale of small amounts of A. Finally, the $N = 8$ and $N = 32$ Nominal strategies have no theoretical bounds and

TABLE 4.1: Transient profit and bounds for simple batch production example. Profits are averaged over $T = 240$ steps of simulation.

Formulation	Transient Profit	Transient Bound	Asymptotic Bound
$N = 8$ Nominal	0.88	$-\infty$	$-\infty$
$N = 32$ Nominal	5.87	$-\infty$	$-\infty$
$N = 5$ Nominal	57.61	$-\infty$	$-\infty$
$N = 8$ Periodic	57.40	57.40	57.85
$N = 8$ Heuristic	57.61	57.61	57.93
$N = 8$ Terminal Cost	57.61	25.02	25.21

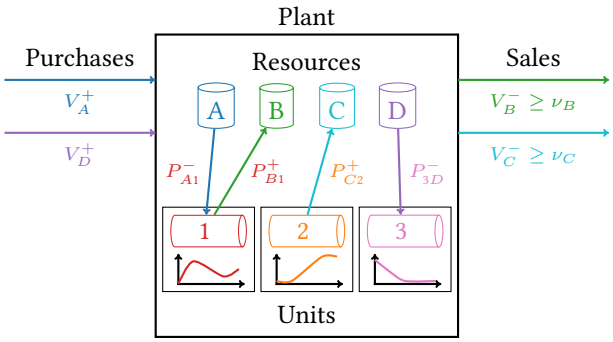


FIGURE 4.11: Diagram of combined scheduling and control system representation. Each unit evolves dynamically while producing and/or consuming resources that can then be sold for profit.

become asymptotically idle as already discussed. We thus conclude that the application of economic MPC strategies provides valuable closed-loop performance guarantees that exclude pathological behavior a-priori.

4.3 CONTINUOUS PRODUCTION WITH UNDERLYING DYNAMICS

We consider now the case of units undergoing continuous operation with transitions between operating points. The relevant sets are now operating regions $i \in \mathbf{I}$ with units $j \in \mathbf{J}$ and resources $k \in \mathbf{K}$. Governing the transitions between units are underlying dynamic state-space models for each unit.

4.3.1 COMBINED MODEL

In the combined formulation, operating modes are essentially predefined tasks for each unit, during which specific sets of resources are produced and consumed by the unit. Resources can additionally purchased from the market or sold as products to meet demand ν_{kt}. Unit production and consumption are determined based on the dynamic evolution of each unit. Inventory levels are bounded (with a bound of zero for resources that cannot be stored) and are accounted for by a material balance. Demand backlog can also be considered to prevent infeasibility when demand cannot be met. This representation is illustrated in Figure 4.11.

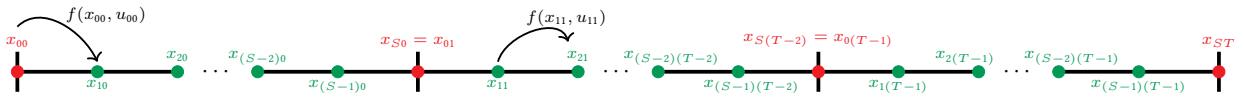


FIGURE 4.12: Diagram of scheduling and dynamic time discretization. Red circles are (shared) endpoints at scheduling times $t \in \bar{\mathbf{T}}$. Green circles are interior dynamic time points $s \in \bar{\mathbf{S}} \setminus \{0, S\}$. Subscripts on all symbols are st (with j omitted for clarity).

4.3.2 Dynamic Layer

In general, the underlying dynamics of each unit could evolve on a faster timescale than the higher-level scheduling layer. Thus, we define two different timescales as follows: intervals $t \in \mathbf{T} \coloneqq \{0, 1, \ldots, T-1\}$ (with time points $\bar{\mathbf{T}} \coloneqq \mathbf{T} \cup \{T\}$) for scheduling decisions, and within each $t \in \mathbf{T}$, the finer discretization $s \in \mathbf{S} \coloneqq \{0, \ldots, S-1\}$ (with points $\bar{\mathbf{S}} \coloneqq \mathbf{S} \cup \{S\}$). We assume that the dynamic s grid gives a fine enough discretization so that any intra-sample dynamics can be safely neglected. The dynamics of unit $j \in \mathbf{J}$ follow a time-invariant discrete-time model of the form $x^+ = f_j(x, u)$, which evolves on the s timescale. Thus, we require states x_{jst} for all scheduling intervals $t \in \mathbf{T}$ and dynamic points $s \in \bar{\mathbf{S}}$, with evolution given by

$$x_{j(s+1)t} = f_j(x_{jst}, u_{jst}), \qquad s \in \mathbf{S}, t \in \mathbf{T}, \tag{4.6}$$

$$x_{j0(t+1)} = x_{jSt}, \qquad t \in \mathbf{T}. \tag{4.7}$$

Dynamic constraints of the form $(x_{jst}, u_{jst}) \in \mathbb{Z}_j$ are enforced via

$$e_j(x_{jst}, u_{jst}) \leq 0 \quad j \in \mathbf{J}, t \in \mathbf{T}, s \in \mathbf{S}, \tag{4.8}$$

using suitably chosen functions $e_j(\,\cdot\,)$. The time grids and dynamic evolution are illustrated in Figure 4.12.

To consider the production of different products, we model each unit as being able to operate in one or more operating modes $i \in \mathbf{I}$. Each of these modes is associated with a region $\mathbb{X}_{ij}$ in state space. To determine the current operating mode, we use indicator variables $W_{ijt} \in \{0, 1\}$. For the unit to be in a given mode during a scheduling interval t, we require

$$W_{ijt} = 1 \implies x_{ist} \in \mathbb{X}_{ij}, \quad s \in \bar{\mathbf{S}},$$

i.e., that the system is in the requisite set throughout the entire interval. When the unit is in transition (or otherwise not in any particular mode), we use the special mode $0 \in \mathbf{I}$. Defining functions $g_{ij}(\,\cdot\,)$ to describe the sets $\mathbb{X}_{ij}$, one possible formulation is

$$g_{ij}(X_{jst}) \leq (1 - W_{ijt})\Omega_{ij}, \quad i \in \mathbf{I} \setminus \{0\}, j \in \mathbf{J}, s \in \bar{\mathbf{S}}, t \in \mathbf{T}, \tag{4.9}$$

using suitably large Ω_{ij}. Activation of mode 0 is then enforced by

$$\sum_{i\in\mathbf{I}} W_{ijt} = 1, \quad j \in \mathbf{J}, t \in \mathbf{T}. \tag{4.10}$$

Since the system could be in mode 0 anywhere in the state space, no specific constraints in the form of (4.9) are required.

Under this representation, the transition from mode i to i' is accomplished by choosing the underlying inputs u_{jst} to move the dynamic states x_{jst} from $\mathbb{X}_{ij}$ to $\mathbb{X}_{i'j}$. The sets $\mathbb{X}_{ij}$ could represent a single steady-state operating point $\mathbb{X}_{ij} = \{x_{ij}^{\text{ss}}\}$, but in general, they can be arbitrarily large. Note that if there is a local regulatory control system on the unit that cannot be removed, then the dynamic models $f_j(\,\cdot\,)$ represent the *controlled* trajectory ot the unit, while the inputs u are the setpoints sent to the regulatory control layer.

4.3.3 Scheduling Layer

At the higher scheduling layer, the decisions are generally about resource balances; unit operating modes are then chosen with respect to the resources that are produced and consumed in each. For flow of resources into and out of the system, we use variables V_{kt}^+ (for purchase) and V_{kt}^- (for sales). Inventory levels are tracked on the dynamic s timescale as follows:

$$L_{k(s+1)t} = L_{kst} + F_{kst}^+ - F_{kst}^- \qquad k \in \mathbf{K}, s \in \mathbf{S}, t \in \mathbf{T}, \tag{4.11}$$

$$L_{k0(t+1)} = L_{kSt} + V_{kt}^+ - V_{kt}^- \qquad k \in \mathbf{K}, t \in \mathbf{T}. \tag{4.12}$$

The variables F_{kst}^+ and F_{kst}^- give the total resource production and consumption across all resources, which are related to the underlying system states by

$$F_{kst}^+ = \sum_{j\in\mathbf{J}} W_{ijt} h_{ijk}^+(x_{jst}, u_{jst}) \qquad k \in \mathbf{K}, s \in \mathbf{S}, t \in \mathbf{T}, \tag{4.13}$$

$$F_{kst}^- = \sum_{j\in\mathbf{J}} W_{ijt} h_{ijk}^-(x_{jst}, u_{jst}) \qquad k \in \mathbf{K}, s \in \mathbf{S}, t \in \mathbf{T}. \tag{4.14}$$

Sales must at least meet prespecified demand ν_{kt}, while purchase must fall within established bounds. To ensure feasibility when demand is excessively high, we allow backlogging via H_{kt}. We thus have the following constraints for sales and backlog:

$$V_{kt}^- + H_{kt} \geq \nu_{kt}, \qquad k \in \mathbf{K}, t \in \mathbf{T}, \tag{4.15}$$

$$B_{k(t+1)} = B_{kt} + H_{kt}, \qquad k \in \mathbf{K}, t \in \mathbf{T}. \tag{4.16}$$

By heavily penalizing backlog B_{kt}, the optimizer will try to meet demand whenever possible.

4.3.4 Objective Function

The optimization objective is to maximize profit. Costs are assumed to come from the scheduling layer (resource purchase, inventory costs, and backlog penalties), while revenue is generated via resource sales. Thus, the objective function is

$$\min \sum_{k \in \mathbf{K}} \left(\sum_{t \in \mathbf{T}} \left(\eta_{kt}^{+} V_{kt}^{+} - \eta_{kt}^{-} V_{kt}^{-} \right) + \sum_{t \in \bar{\mathbf{T}}} \left(\alpha_k L_{kt} + \beta_k B_{kt} \right) \right). \tag{4.17}$$

Other terms could be added provided that they are summed over $t \in \mathbf{T}$ as in the economic MPC formulation from Section 3.3. This generality would allow cost or earliness minimization, but it does not allow straightforward treatment of makespan minimization.

4.3.5 State-Space Representation

The overall integrated model is given by constraints (4.6) to (4.16) with objective function (4.17). Although the underlying unit models are already in state-space form, we require a state-space formulation of the overall problem on the scheduling t timescale. To this end, we classify all decision variables as either states, inputs, or auxiliary variables used to aid in the problem formulation. For these purposes, we assume that auxiliary variables are uniquely specified based on the fixed values of states and inputs.

Based on these requirements, the overall system state is given by the values of the following variables for a given $t \in \bar{\mathbf{T}}$:

- The underlying dynamic states x_{j0t} at the beginning of the time interval (i.e., for $s = 0$)
- The inventory levels L_{j0t} (again for $s = 0$)
- The current backlog level B_{kt}

System inputs are the following:

- The underlying dynamic inputs u_{jst} for *all* dynamic intervals $s \in \mathbf{S}$
- The production variables W_{ijt}
- The purchase and sales decisions V_{kt}^{+} and V_{kt}^{-}
- The backlog increment H_{kt}

Auxiliary variables are thus the remaining variables as follows:

- The dynamic states x_{jst} for intermediate dynamic time points $s \in \bar{\mathbf{S}} \setminus \{0\}$ (i.e., $s \neq 0$), which are defined by (4.6)

- The intermediate inventory levels L_{kst} (again for $s \neq 0$), which are defined by (4.11)
- The production and consumption levels P^+_{kst} and P^-_{kst} (for all $s \in \mathbf{S}$), defined by (4.13) and (4.14)

With these definitions, the evolution of overall states is given by equations (4.7), (4.12) and (4.16), while input and state constraints are given by (4.8) to (4.10) and (4.15). With this representation, closed-loop properties can be investigated, e.g., by placing terminal costs and constraints on the appropriate X variables to provide recursive feasibility.

4.4 Dynamic-Aware Scheduling Model

While it is possible formulate a combined optimization problem for scheduling decisions and unit dynamic regulation, it is not necessarily possible to solve that formulation within a reasonable amount of time. Particularly due to the presence of so-called "big-M" constraints in (4.10), the formulation may not be very tight, and even finding feasible solutions may be a challenge. Thus, we wish to define a scheduling approximation that considers some of the flexibility of the underlying dynamic system without requiring the full dynamic models. The goal is to formulate a model in such a way that any feasible solution can be translated into a feasible dynamic trajectory for the system to follow.

To be specific, let (X, U) represent the states and inputs of the scheduling approximation, and let (x, u) be the states of the full integrated model. For each time point, we require that any feasible X and U can be mapped to a feasible x and u such that the respective stage costs satisfy

$$\ell(x, u, t) \leq L(X, U, T)$$

in which $L(\,\cdot\,)$ is the stage cost of the scheduling model. By generating a reference trajectory (X_r, U_r) for the scheduling model and using terminal constraints as in Section 3.3, we can use the above inequality and the performance bound from Theorem 3.15, the closed-loop x and u that is generated satisfies

$$\lim_{T\to\infty} \frac{1}{T} \sum_{k=t}^{t+T-1} \big(\ell(x(k), u(k), k) - L(X_r(k), U_r(k), k)\big).$$

That is, we generate a feasible trajectory for the composite system that is at least as good as the cost predicted by the scheduling system reference trajectory. This trajectory can then be followed in the nominal case, or tracked using tracking MPC on a per-unit basis when disturbances are present. Therefore, we have a bound on the performance of the composite system without having to solve any instances of the combined problem.

In the following subsections, we describe how the standard scheduling model from Section 4.3 can be augmented to generate feasible dynamic trajectories as desired. The dynamic-aware scheduling formulation remains an MILP, and thus it is not significantly more computationally challenging than the standard scheduling model.

4.4.1 Basic Transition Structure

To start, we describe the basic structure for unit transitions. We assume that each feasible transition from region i to i' has an associated set $m \in \mathbf{M}_{ii'j}$ of possible transition trajectories, each with fixed transition time $\tau_{ii'jm}$. We assume for brevity that $\tau_{ii'jm} \geq 1$, although the constraints for $\tau_{ii'jm} = 0$ are straightforward. These transitions are associated with trajectories $(x^{\Delta}_{ii'jmst}, u^{\Delta}_{ii'jmst})$ in the unit's dynamic models. To initiate a transition, the optimizer uses the discrete variable $Y_{ii'jt} \in \{0, 1\}$, with the system's trajectory obtained via convex combination of trajectories in $\mathbf{M}_{ii'j}$ using weights $U_{ii'jmt} \in [0, 1]$. This idea is similar to a model from Chu and You (2012), except that the optimizer is allowed to take a convex combination of candidates rather than choosing just one. This choice leads to constraints of the form

$$\sum_{m \in \mathbf{M}_{ii'j}} U_{ii'jmt} = Y_{ii'jt}, \quad i, i' \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}. \tag{4.18}$$

As in the batch scheduling problem, we must lift these variables to have a state-space representation. We use $Z_{ii'jmnt}$ to track the evolution of these variables, with $n \in \mathbf{N}_{ii'jm} \coloneqq \{1, \ldots, \tau_{ii'jm}\}$. For convenience, we split this update into two separate pieces as follows:

$$Z'_{ii'jmnt} \coloneqq \begin{cases} Z_{ii'jmnt} + U_{ii'jmt} & \text{if } n = \tau_{ii'jm} \\ Z_{ii'jmnt} & \text{else} \end{cases} \quad i, i' \in \mathbf{I}_j, j \in \mathbf{J}, m \in \mathbf{M}_{ii'j}, n \in \mathbf{N}_{ii'jm}, t \in \mathbf{T}, \tag{4.19}$$

$$Z_{ii'jmn(t+1)} \coloneqq \begin{cases} 0 & \text{if } n = \tau_{ii'jm} \\ Z'_{ii'jm(n+1)t} & \text{else} \end{cases} \quad i, i' \in \mathbf{I}_j, j \in \mathbf{J}, m \in \mathbf{M}_{ii'j}, n \in \mathbf{N}_{ii'jm}, t \in \mathbf{T}. \tag{4.20}$$

Here, the auxiliary variable $Z'_{ii'jmnt}$ gives the values of the multipliers considering any incipient transition at the current time t. After transitions are complete, we use the variable $X_{ijt} \in [0, 1]$, given by

$$X'_{ijt} = X_{ijt} - \sum_{i' \in \mathbf{I}} Y_{ii'jt}, \quad i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}, \tag{4.21}$$

$$X_{ij(t+1)} = X'_{ijt} + \sum_{i' \in \mathbf{I}} \sum_{m \in \mathbf{M}_{ii'j}} Z'_{i'ijm1t}, \quad i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}. \tag{4.22}$$

Thus, X'_{ijt} gets decremented when the system transitions away from that region, and it is incremented when transitions to that region are completed. Finally, we require

$$W_{ijt} \leq X'_{ijt}, \quad i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}. \tag{4.23}$$

This relationship allows the optimizer to select a given operating mode only if the unit has fully completed any transition to that operating mode.

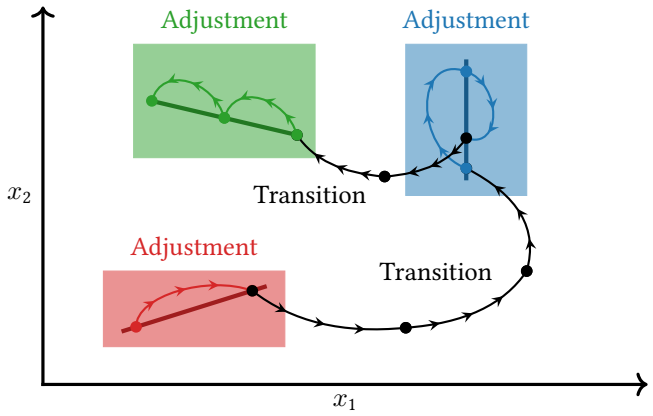


FIGURE 4.13: Diagram of unit transitions and adjustments. Original production regions are shown as shaded rectangles, approximate regions are shown in dark lines (the segment between the two points in the sets $\mathbf{P}_{ij}$). Transitions move from one production segment in another (possibly over multiple periods), while adjustments move within a production region in one period.

4.4.2 Operating Points

In the general formulation, the operating regions $\mathbb{X}_{ij}$ were allowed to be arbitrary sets in state space. To reduce this computational burden, we assume the operating regions are defined by the convex combination points indexed by $p \in \mathbf{P}_{ij}$, each with associated weights G_{ijpt}. In the unit's state space, each of these points represents a steady-state operating pair $(x^{\text{ss}}_{ijp}, u^{\text{ss}}_{ijp})$. These values evolve by

$$G'_{ijpt} = G_{ijpt} - \sum_{i' \in \mathbf{I}} \sum_{m \in \mathbf{M}_{ii'j}} \gamma^{-}_{ii'jmp} U_{ii'jmt}, \quad i \in \mathbf{I}, j \in \mathbf{J}, p \in \mathbf{P}_{ij}, t \in \mathbf{T}, \tag{4.24}$$

$$G_{ijp(t+1)} = G'_{ijpt} + \sum_{i' \in \mathbf{I}} \sum_{m \in \mathbf{M}_{i'ij}} \gamma^{+}_{i'ijmp} Z'_{i'ijm1t}, \quad i \in \mathbf{I}, j \in \mathbf{J}, p \in \mathbf{P}_{ij}, t \in \mathbf{T}. \tag{4.25}$$

Here, the parameters $\gamma^{-}_{ii'jmt}$ and $\gamma^{+}_{ii'jmt}$ define the points from which a given transition leaves $\mathbb{X}_{ij}$ and enters $\mathbb{X}_{i'j}$ for a given transition (i, i', m). To ensure that the optimizer selects transitions that leave from the correct operating point, we have the constraint

$$\sum_{p \in \mathbf{P}_{ij}} G'_{ijpt} = X'_{ijt}, \quad i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T}, \tag{4.26}$$

as well as the bounds $G_{ijpt} \in [0, 1]$.

Transitions within a given operating region are modeled exactly as in the previous section, noting that for this case $i = i'$. We refer to these changes transitions within an operation region as "adjustements" to differentiate between true transitions between operating modes. It is assumed that every adjustment included in the set $\mathbf{M}_{iij}$ can be made without leaving the original operating region $\mathbb{X}_{ij}$. An example of transitions and adjustments is shown in Figure 4.13.

4.4.3 Resource Balance

Because of the dynamic character of the system, we still keep track of resources on the faster s timescale. Thus, constraints (4.11) and (4.12) are exactly as before. However, the instantaneous production and consumption rates (constraints (4.13) and (4.14)) must be modified. Recalling that the operating points $p \in \mathbf{P}_{ij}$ are associated with steady states $(x^{\text{ss}}_{ijp}, u^{\text{ss}}_{ijp})$, and transitions $m \in \mathbf{M}_{ii'j}$ following trajectories $(x^{\Delta}_{ii'jmst}, u^{\Delta}_{ii'jmst})$, we can define parameters to give resource production/consumption rates as follows:

$$\begin{aligned}
\phi^{\pm}_{ijkp} &\coloneqq h^{\pm}_{ijk}(x^{\text{ss}}_{ijp}, u^{\text{ss}}_{ijp}), \quad i \in \mathbf{I}, j \in \mathbf{J}, k \in \mathbf{K}, p \in \mathbf{P}_{ij}, \\
\zeta^{\pm}_{ii'jkmns} &\coloneqq h^{\pm}_{0jk}(x^{\Delta}_{ii'jmst}, u^{\Delta}_{ii'jmst}), \quad i, i' \in \mathbf{I}, j \in \mathbf{J}, k \in \mathbf{K}, m \in \mathbf{M}_{ii'j}, s \in \mathbf{S}, n = \tau_{ii'j} - t.
\end{aligned}$$

These values give the production/consumption rates during stead-state operation and during transition. Here, the superscript $^{\pm}$ indicates that parameters are defined for both production $^{+}$ and consumption $^{-}$. With these parameters, we then have

$$F^{\pm}_{kst} = \sum_{j \in \mathbf{J}} \left(\sum_{i \in \mathbf{I}} W_{ijt} \left(\sum_{p \in \mathbf{P}_{ij}} \phi^{\pm}_{ijkp} G'_{ijpt} \right) + \sum_{i,i' \in \mathbf{I}} \sum_{m \in \mathbf{M}_{ii'j}} \sum_{n \in \mathbf{N}_{ii'jm}} \zeta^{\pm}_{ii'jkmns} Z'_{ii'jmnt} \right), \quad k \in \mathbf{K}, s \in \mathbf{S}, t \in \mathbf{T}. \quad (4.27)$$

Note that the nonlinearity between W_{ijt} and G'_{ijpt} can be linearized, as W_{ijt} is discrete. Finally, the sales and backlog are as in (4.15) and (4.16)

4.4.4 Discussion

As mentioned in the introduction, the goal of this model is to be able to map any solution onto a feasible trajectory for the underlying unit dynamic states. For the simple case of choosing singleton $\mathbf{M}_{ii'j}$ and $\mathbf{P}_{ij}$, the mapping is immediately clear: during a transition, the system follows the single trajectory $x^{\Delta}_{ii'jst}$; otherwise, the system is at steady state x^{ss}_{ij}. We illustrate these cases in Figure 4.14. When multiple trajectories and/or operating points are added, the optimizer is allowed to interpolate between them. Assuming the underlying dynamic models $f_j(\,\cdot\,)$ and output functions $h^{\pm}_{ijk}(\,\cdot\,)$ are affine and the feasible sets $\mathbb{Z}_j$ are convex, the interpolation procedure generates a feasible dynamic trajectory. In particular, we have

$$x_{jst} = \sum_{i \in \mathbf{I}} \sum_{p \in \mathbf{P}_{ij}} G_{ijpt} x^{\text{ss}}_{ijp} + \sum_{i,i' \in \mathbf{I}} \sum_{m \in \mathbf{M}_{ii'j}} Z'_{ii'jmnt} x^{\Delta}_{ii'jms(\tau_{ii'j}-n)},$$

with the corresponding expression for u_{jst}. However, if the underlying dynamics are nonlinear, then the interpolated trajectories are not necessarily feasible. It may be possible to find a nearby feasible trajectory by solving a small optimization problem, but this property is not guaranteed. If it is absolutely necessary to generate a feasible

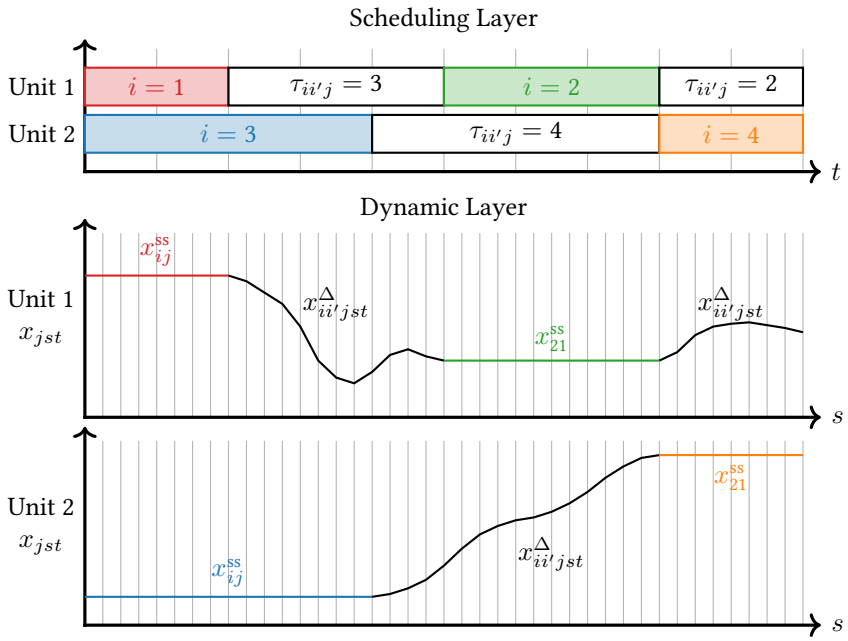


FIGURE 4.14: Mapping from scheduling decisions to underlying dynamic states. During production, units operate at x^{ss}_{ij}. During transitions, units follow $x^{\Delta}_{ii'jst}$.

dynamic trajectory, then the variables G_{ijpt} and $Z_{ii'jmnt}$ can be changed from continuous to discrete; that way, the interpolation formula will have only a single nonzero term which corresponds to a point on a known feasible dynamic trajectory. Of course, with this increase in the number of discrete variables, it may not be tractable to include many different transitions $\mathbf{M}_{ii'j}$ or operating points $\mathbf{P}_{ij}$. However, the original combined model is likely to be completely intractable as well.

The main value of the dynamic-aware scheduling model is that the underlying dynamic models need not be included in full, and the problem can be solved as an MILP. At first glance, this formulation appears to add a large number of variables due to the presence of the $Z_{ii'jmnt}$ variables. However, because these are defined by trivial equality constraints in (4.19) and (4.20), these variables can be immediately removed via presolve. Thus, in each scheduling time point, we have $I(I+1)J$ nontrivial discrete variables and $I(IM+P)J$ continuous variables. By contrast, the full integrated model requires IJ discrete indicator variables as well as IJS state and input vectors. For even modest state and input dimensions, these are roughly the same total number of variables, but the scheduling formulation is much tighter by avoiding the indicator variable constraints for $x_{jst} \in \mathbb{X}_{ij}$. When S is very large, the scheduling approximation can be significantly easier to solve compared to the combined formulation. In addition, if S is too large to efficiently solve the scheduling approximation, we can apply the conservative resampling procedure described in Section 4.7 to reduce the required number of s points, albeit at the cost of shrinking the feasible region.

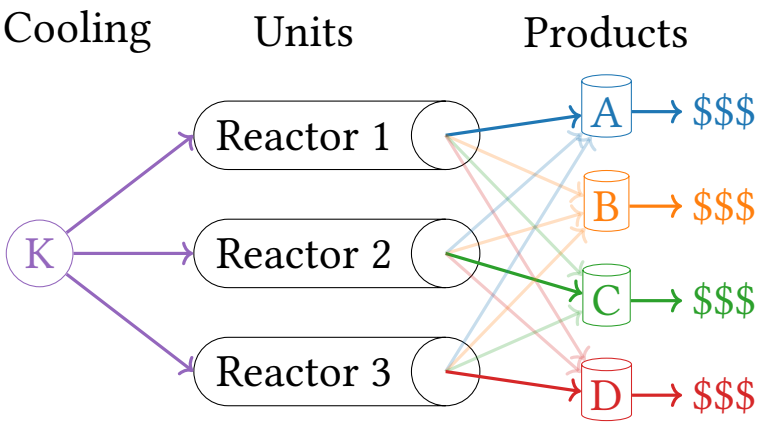


Figure 4.15: Diagram of multi-product reactor system. Reactors are identical and can make any product. All reactors share a common cooling source, which imposes a constraint on total instantaneous cooling.

4.5 Case Study: Multi-Product Reactor

To close this section, we consider a multi-product reactor adapted from Flores-Tlacuahuac and Grossmann (2006) and Du et al. (2015). We consider one or more identical reactors operating in parallel. Each reactor can make a slate of products; when a reactor changes products, there is a transition time during which nothing is produced. Reactor output is sent to holding tanks from which it is eventually sold, with different products having different sale prices. Product A is worth $200/unit, decreasing linearly to product D, which is worth $50/unit. Backlogs are penalized 100 times sale prices each period. Each product has a minimum daily demand that must be met or added to backlog (which accrues a cost penalty), which means the reactors must cycle between all products, rather than producing only the most profitable product. When there are multiple reactors, they all share a common cooling system. Based on available cooling capacity, certain combinations of products cannot be produced simultaneously. Thus, the optimizer must account for this limitation when choosing product sequences. A system with three reactors is illustrated in Figure 4.15. We wish to apply the closed-loop scheduling techniques discussed in this chapter and compare them to more traditional scheduling techniques.

4.5.1 Single-Reactor Nominal Case

We begin by simulating nominal operation of a single-reactor system. Because there is only a single reactor, the shared cooling constraint is never binding, and the reactor can freely make any product at any time. To start, we determine the optimal periodic solution for varying periods. With orders due every 24 h, it is natural to look for solutions whose periods are a multiple of 24 h. Given the demand and transition times, there is no feasible 24 h periodic solution, but optimal solutions for longer periods are shown in Figure 4.16. No solutions experience backlog. As seen in the figure, the longer horizons generally make more profit due to having fewer transitions over a given

period. Note also that for periods larger than 48, not all products have the same number of production runs in the horizon. Thus, periodic solutions that produce each product only once (as in Flores-Tlacuahuac and Grossmann (2006) and Du et al. (2015)) may not be optimal.

Using these periodic reference trajectories, we simulate the nominal closed-loop system for 30 days using various reference trajectories, rescheduling frequencies, and horizons. The system begins with enough inventory for half of a daily order. Closed-loop costs are shown in Table 4.2. In the first row, we approximate a standard scheduling strategy by rescheduling only every 12 h without using any terminal constraint. With the shorter horizons, this strategy leads to high levels of backlog and thus lower closed-loop profit. The first week of the closed-loop schedule for the $N = 48$ case is shown in Figure 4.17a. We see here that the system is consistently late delivering orders of product D due to its lack of foresight. To simulate a naive closed-loop scheduling solution, we repeat the simulation but now rescheduling every hour. For the shorter horizons, this faster rescheduling leads to better performance, but paradoxically, the $N = 72$ simulation is actually worse.

The remaining simulations all employ periodic reference trajectories, which are used as an exact terminal constraint and thus provide an asymptotic bound on closed-loop cost by Theorem 3.15. Note that because the system parameters are 24 h periodic, there are two equivalent $T = 48$ periodic solutions that differ only by phase (with three for $T = 72$). For comparison, we simulate using all of the equivalent reference trajectories to determine how sensitive the initial transient is to the phase of the reference trajectory. As observed in this table, not all of the economic MPC strategies meet their cost bounds over the simulation period. The $T = 48$(b) and $T = 72$(b) cases do exceed their cost bounds, which indicates that the phase of these reference trajectories is favorable for the system's initial condition. The remaining controllers apparently struggle to reach the reference trajectory within the initial days of the simulation, leading to a large number of transitions and temporarily high transient cost. We show the closed-loop schedule for $N = 24$ with the $T = 48$(b) reference trajectory in Figure 4.17b. However, Theorem 3.15 states that the average cost will eventually recover. To verify, we recalculate closed-loop costs, except that we exclude the first 6 days of simulation. These values are shown in Table 4.3. We see now that after the initial transient, all simulations at least meet their profit bound, with the longer $N = 72$ horizon able to beat the $T = 48$ reference trajectory. By contrast, the strategies without terminal constraints generally do not recover after the initial transient, and thus their poor performance continues indefinitely. Thus, the inclusion of terminal constraints leads to better closed-loop performance, although transient cost can be sensitive to the chosen reference trajectory.

4.5.2 Multiple-Reactor Nominal Case

To increase the difficulty of the problem, we repeat the previous simulations now using a system of three reactors in parallel. Minimum product demand is increased to compensate for the increased production capacity. When a

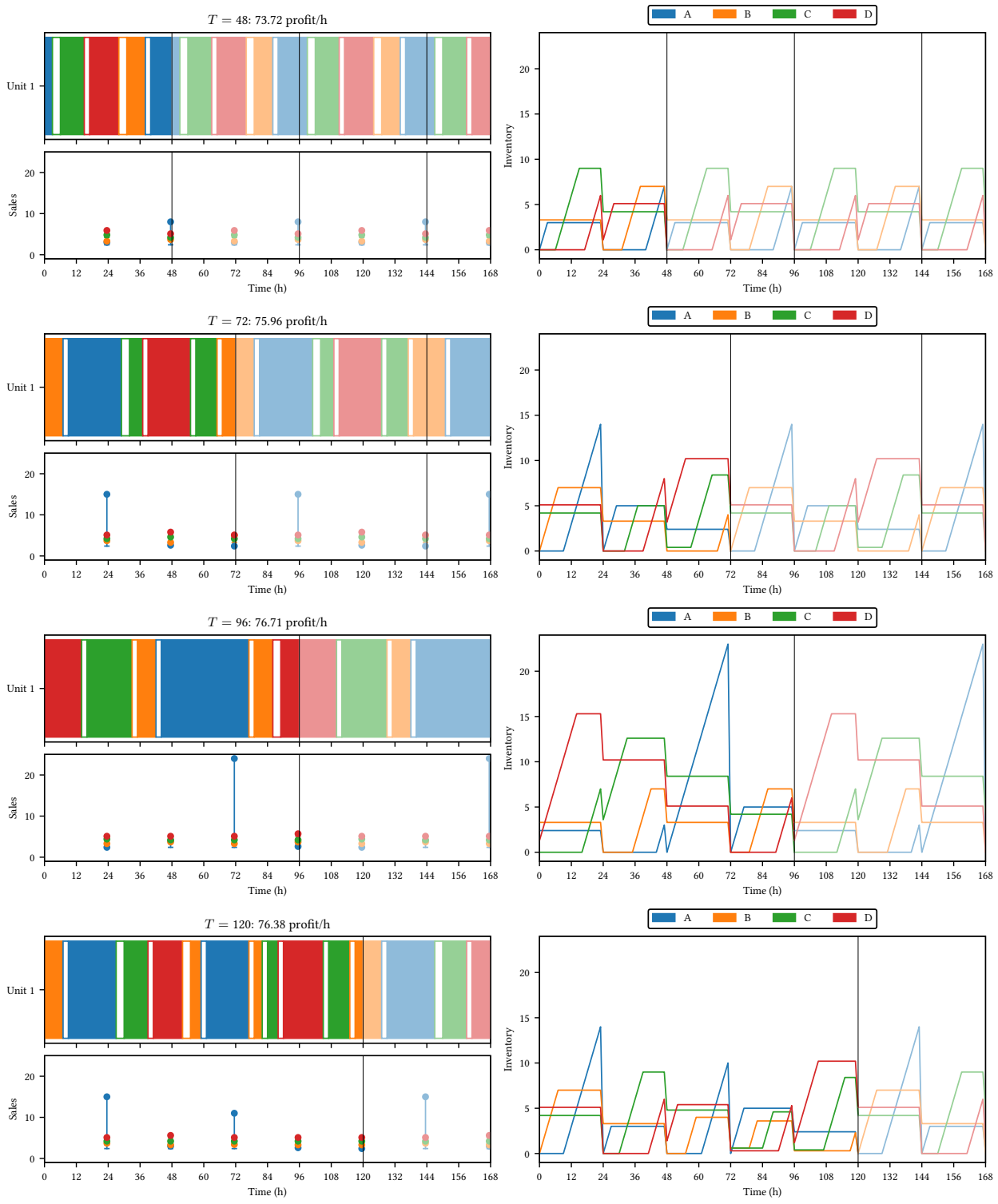
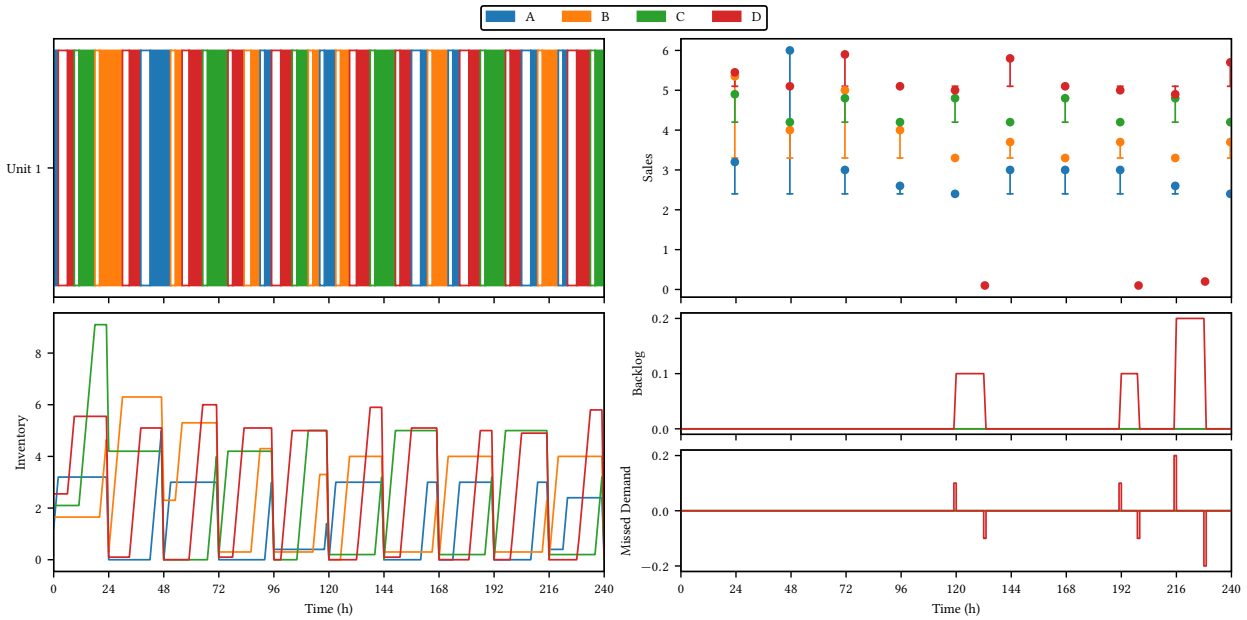
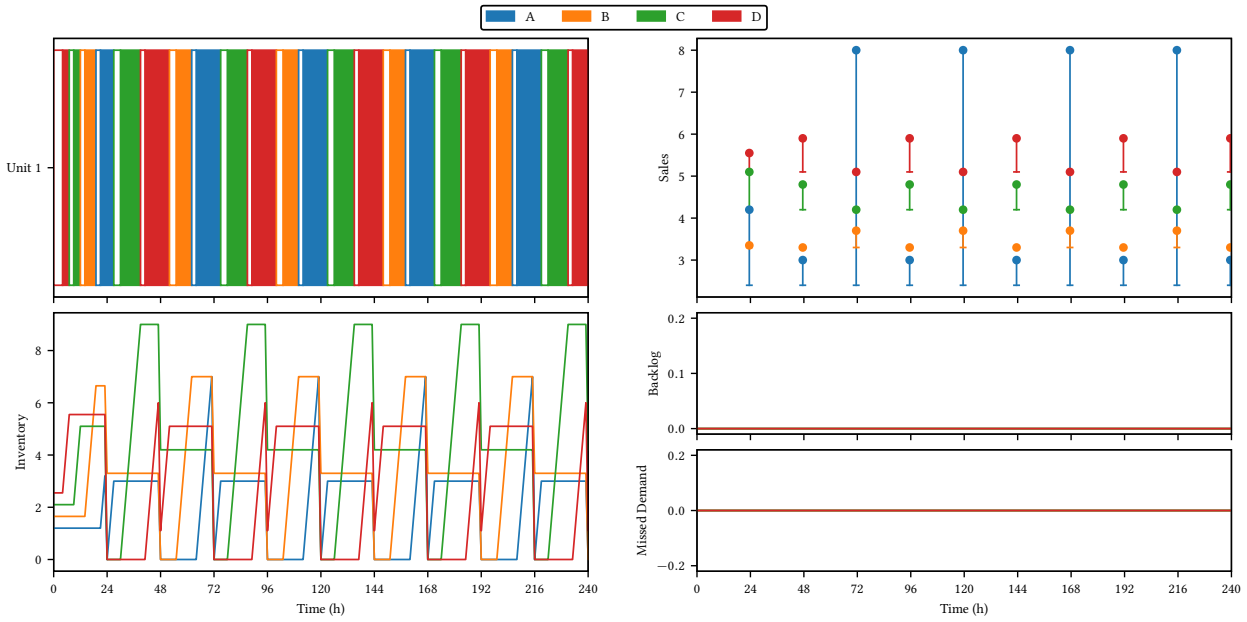


FIGURE 4.16: Periodic solutions for one-reactor system. Horizons and average profit are shown in each figure. In sales plot, circular markers show actual sale amount, while flat markers indicate minimum demand.



(A) $N = 48$ solution without any terminal constraints. Rescheduling occurs every 12 h.



(B) $N = 24$ solution with $T = 48$(b) periodic terminal constraint. Rescheduling occurs every hour.

FIGURE 4.17: Example closed-loop trajectories for the single-reactor problem.

TABLE 4.2: Closed-loop costs for nominal one-reactor case over 30 day simulations. Values are normalized to the $T = 48$ periodic reference trajectory.

Reference Trajectory	Reference Cost	Rescheduling Period	Horizon		
			$N = 24$	$N = 48$	$N = 72$
None	N/A	12 h	-364.94%	-22.37%	-4.08%
None	N/A	1 h	-39.55%	$+0.14\%$	-10.15%
$T = 48$(a)	$+0.00\%$	1 h	$+0.67\%$	$+0.74\%$	$+3.11\%$
$T = 48$(b)	$+0.00\%$	1 h	-1.28%	-1.00%	$+2.13\%$
$T = 72$(a)	$+3.04\%$	1 h	$+0.06\%$	$+1.30\%$	$+1.30\%$
$T = 72$(b)	$+3.04\%$	1 h	$+3.71\%$	$+3.92\%$	$+3.92\%$
$T = 72$(c)	$+3.04\%$	1 h	$+0.48\%$	$+1.99\%$	$+2.32\%$

TABLE 4.3: Closed-loop costs for nominal one-reactor case over 30 day simulations, excluding the first 6 days of startup. Values are normalized to the $T = 48$ periodic reference trajectory.

Reference Trajectory	Reference Cost	Rescheduling Period	Horizon		
			$N = 24$	$N = 48$	$N = 72$
None	N/A	12 h	-431.54%	-24.67%	-3.15%
None	N/A	1 h	-42.02%	$+1.06\%$	-10.72%
$T = 48$(a)	$+0.00\%$	1 h	-0.00%	-0.00%	$+1.68\%$
$T = 48$(b)	$+0.00\%$	1 h	-0.00%	-0.00%	$+1.74\%$
$T = 72$(a)	$+3.04\%$	1 h	$+3.04\%$	$+3.04\%$	$+3.04\%$
$T = 72$(b)	$+3.04\%$	1 h	$+3.04\%$	$+3.04\%$	$+3.04\%$
$T = 72$(c)	$+3.04\%$	1 h	$+3.04\%$	$+3.04\%$	$+3.04\%$

reactor is operating at the steady-state operating point for product A, 4 units of cooling are required, with 3 units for B, 2 units for C, and 1 unit for D. The overall cooling system can provide only 9 units of cooling at any given time. Thus, for each reactor producing Product A, another reactor must be producing either C or D. As before, we begin by determining optimal periodic solutions; two such solutions are shown in Figure 4.18. We note that in the case of the $T = 120$ solution, the optimal solution actually contains a small amount of backlog. This behavior could of course be removed by increasing the backlog penalty, but it also indicates that looser order deadlines could lead to much higher profit.

As before, we simulate a variety of reference trajectories, rescheduling frequencies, and horizons. The system once again starts with a half day's order in inventory, and simulations last 30 days. Closed-loop costs are shown in Table 4.4. From these values, we see qualitatively similar behavior to the single-reactor case, although the greater flexibility of having three reactors leads to better performance for the simulations without terminal constraints. In this case, the initial condition appears to be favorable to the system, as all economic MPC strategies out-perform their theoretical bounds, with longer horizons performing slightly better. We note in particular that the $N = 24$ case with $T = 96$ terminal constraint out-performs the $N = 72$ case without any terminal constraint. Both solutions are shown in Figure 4.19. Despite the short horizon of the $N = 24$, the terminal constraint essentially allows the

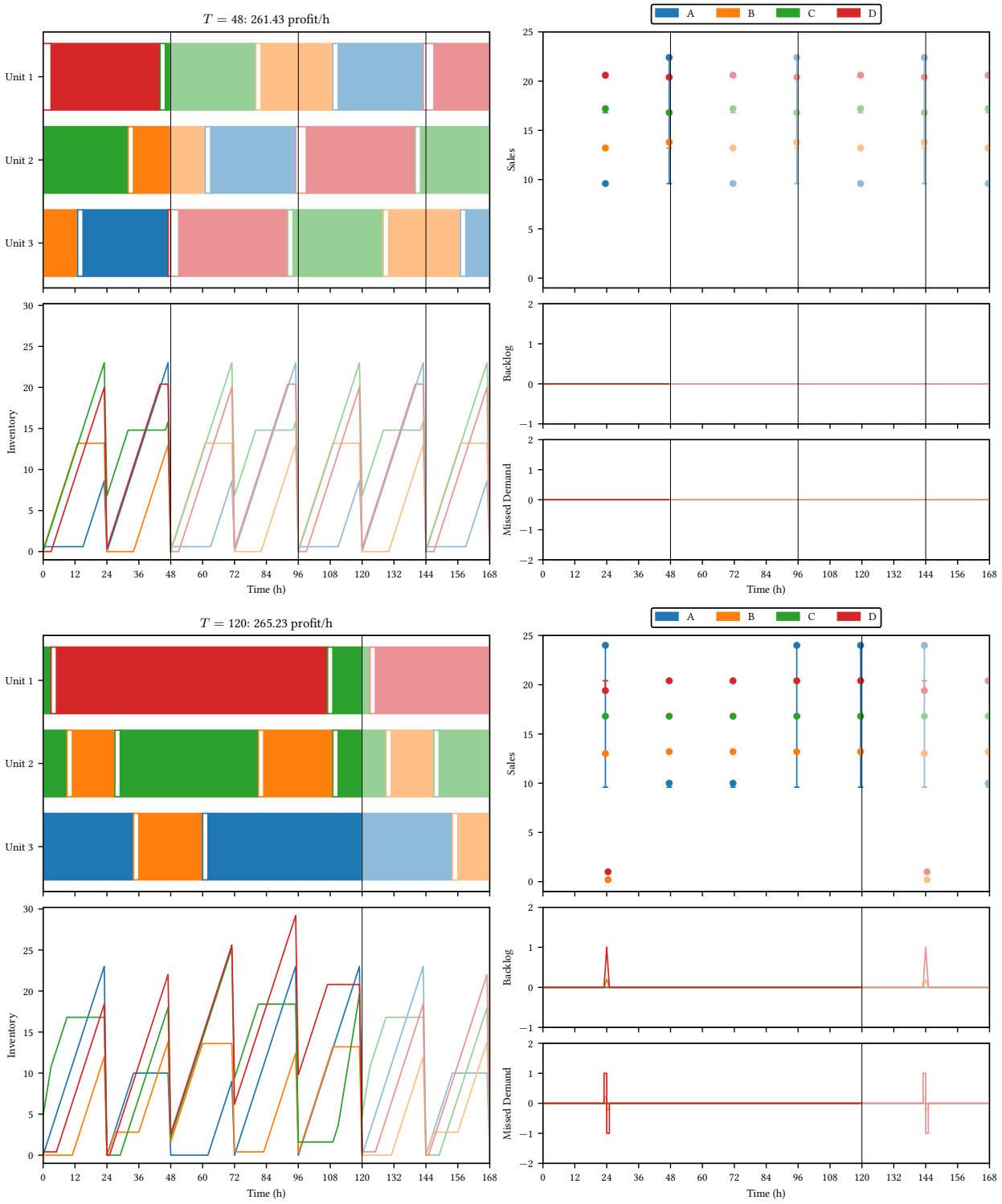


FIGURE 4.18: Periodic solutions for the three-reactor system.

TABLE 4.4: Closed-loop costs for nominal three-reactor case over 30 days of simulation. Values are normalized to the $T = 48$ periodic reference trajectory.

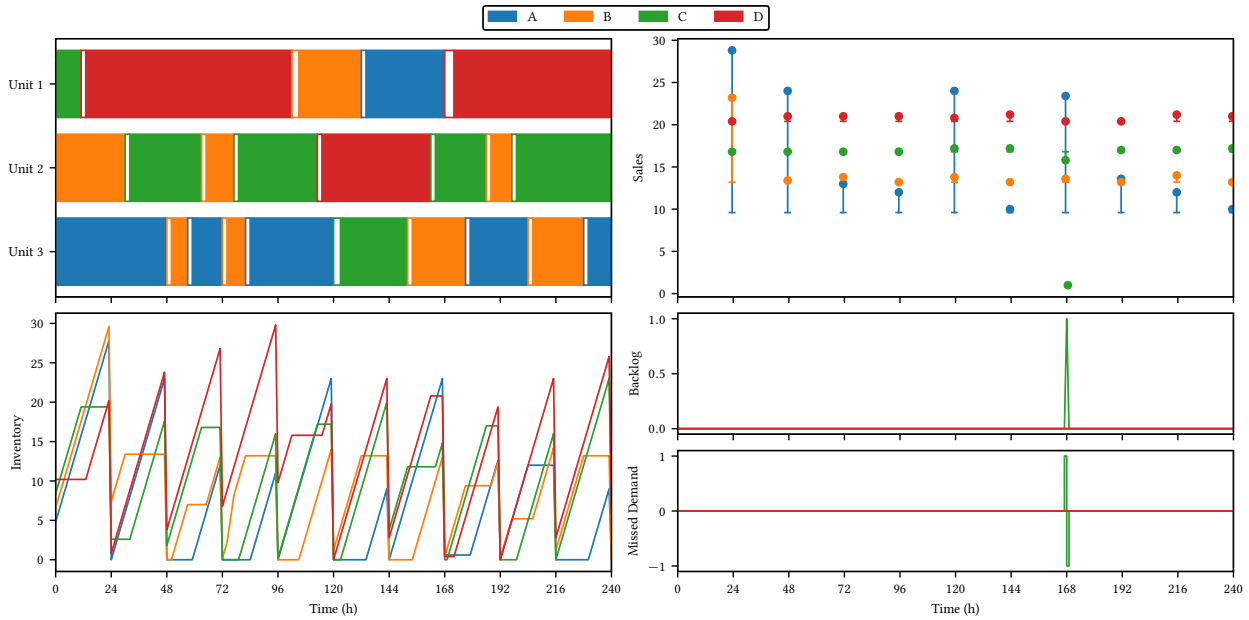
Reference Trajectory	Reference Cost	Rescheduling Period	Horizon		
			$N = 24$	$N = 48$	$N = 72$
None	N/A	12 h	-16.76%	$+1.34\%$	$+0.46\%$
None	N/A	1 h	-4.11%	-0.05%	$+1.17\%$
$T = 48$(a)	$+0.00\%$	1 h	$+1.67\%$	$+1.74\%$	$+2.29\%$
$T = 48$(b)	$+0.00\%$	1 h	$+1.18\%$	$+1.45\%$	$+2.11\%$
$T = 96$(a)	$+1.80\%$	1 h	$+3.36\%$	$+3.38\%$	$+3.38\%$
$T = 96$(b)	$+1.80\%$	1 h	$+2.89\%$	$+2.89\%$	$+2.91\%$
$T = 96$(c)	$+1.80\%$	1 h	$+3.38\%$	$+3.42\%$	$+3.42\%$
$T = 96$(d)	$+1.80\%$	1 h	$+2.87\%$	$+2.88\%$	$+2.97\%$

optimizer to plan ahead and avoid unnecessary transitions. Thus, terminal constraints provide a means to reduce online computational load, as they generally reduce the necessary horizon to achieve good performance.

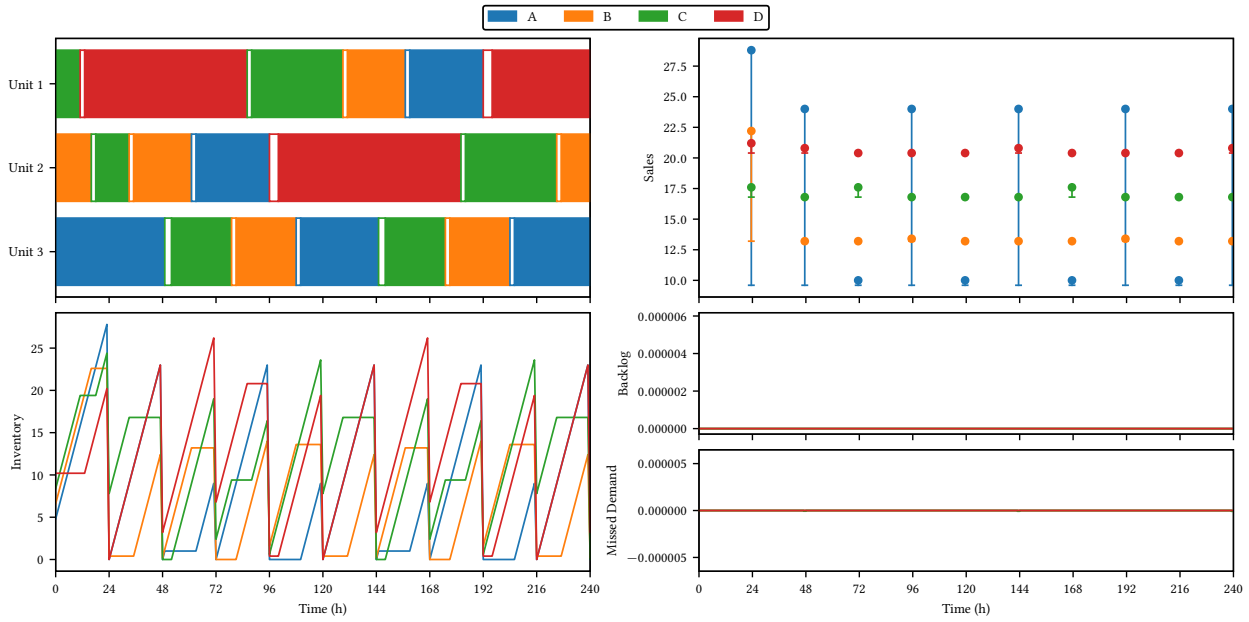
4.5.3 Single-Reactor With Yield Disturbances

While the nominal performance properties are useful, real systems are subject to disturbances that can invalidate nominal properties. Thus, we wish to examine whether the economic MPC strategy of including a terminal constraint is still advantageous when the reference trajectory is not necessarily feasible due to disturbances. To investigate, we simulate the system in closed loop subject to $\pm 15\%$ random disturbances in product yield. At each timestep, after finding the optimal schedule, the yield disturbance is added by updating product inventory (either increasing or decreasing). Note that if the reactor output was immediately being sold, any negative yield disturbance leads to a backlog rather than negative value of inventory. We simulate $N = 48$ horizons for 10 days, both with and without a $T = 48$ periodic reference trajectory. For simplicity, we consider only a single reactor.

Figure 4.20 shows the distribution of closed-loop profit and backlog costs across 250 simulations with and without terminal constraints. In particular, we note that mean profit is 3.9% higher and mean backlog costs are 0.5% lower (both values normalized to the nominal $T = 48$ periodic profit) when the terminal constraint is included compared to when it is not. Note that on a relative basis, the terminal constraints reduce backlog by roughly 75%, although both values are small (approximately 0.5% of total cost). Thus, despite the lack of theoretical properties, the terminal constraint still appears to be beneficial, as it essentially anchors the optimizer to a quality trajectory that is feasible on average. We note that for feasibility purposes, the terminal constraints have been softened using a large linear penalty function; thus, if the optimizer can reach the reference trajectory (i.e., $x \in \mathcal{X}_N(t)$ in the language of Chapter 3), then it will do so, and if not, the optimizer will find a nearby solution. It is possible that a run of bad disturbances could push the system far from the reference trajectory, which could lead to poor performance when the optimizer struggles to simultaneously lower backlog while also producing the extra product needed to meet the reference trajectory. In



(A) $N = 72$ solution without any terminal constraints.



(B) $N = 24$ solution with $T = 96$(c) periodic terminal constraint.

FIGURE 4.19: Example closed-loop trajectories for the three-reactor problem. Both use hourly rescheduling.

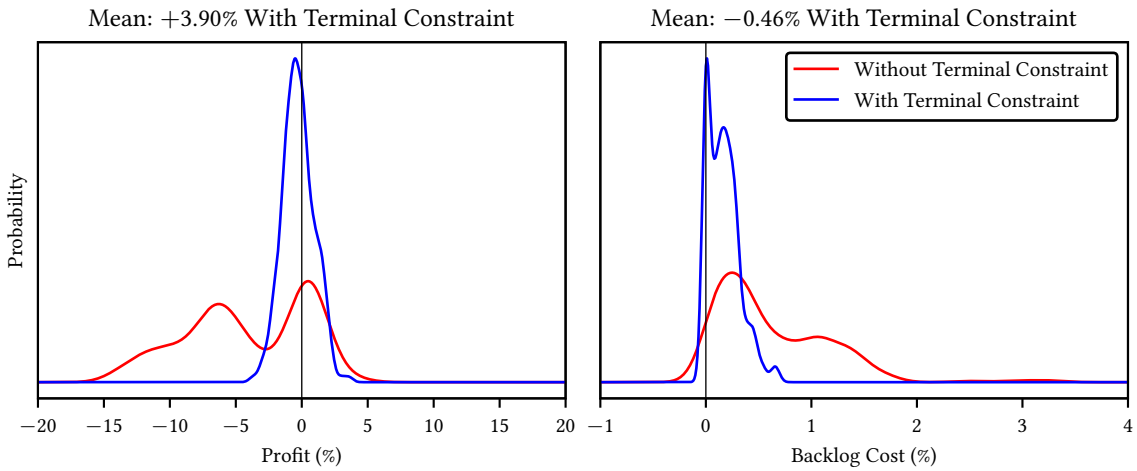


FIGURE 4.20: Kernel density estimate for profit and backlog costs for 250 trials of the yield disturbance simulation. Values are normalized to the profit of the $T = 48$ periodic reference trajectory.

such cases, a practical solution is to temporarily switch or remove the terminal constraint until the system recovers. However, for this system, such behavior was not observed. We conclude that inclusion of the terminal constraint is still generally helpful in the presence of disturbances, and thus is still useful despite the lack of theoretical properties.

4.5.4 Underlying Unit Dynamics

As a final example, we consider the addition of underlying unit dynamics. Each reactor has two state-space outputs: temperature T, and production P. These outputs are affected by inputs cooling flow F_c and throughput F, which are constrained to $F_c \in [0, 1]$, and $F \in [-0.1, 1.1]$. Note that F_c corresponds to use of the cooling resource, which shares a total constraint across all three reactors. The system model is linear with five states. Step responses are shown in Figure 4.21. There is an operating region associated with each product A, B, C, and D, and the sets $\mathbb{X}_{ij}$ are defined in terms of temperature as illustrated in Figure 4.22.

We compare the full integrated model and the approximate dynamic-aware model by trying to find the optimal $T = 48$ h periodic solution for both. For the approximate model, we start by defining the catalog of operating points and transitions for each operating region. As operating points, we take the steady-states with T at the central values and the production rate F at 1 (nominal production) and 1.1 (extra production), which gives two elements in $\mathbf{P}_{ij}$ (indicated with markers in Figure 4.22). Note that these values back off slightly from the bounds so that adjustments between the operating points can be completed in finite time. Thus, the approximate model will slightly underestimate the the production capacity of each reactor. The transitions are then taken between the corresponding points of each $\mathbb{X}_k$. Note that, due to the time constants of the reactor dynamics, transitions to and from the steady-states take one to two periods longer than the corresponding transitions between (nonsteady) points in the operating regions. These extra moves are required to zero out the low-amplitude transient dynamics near the nominal operating temperatures.

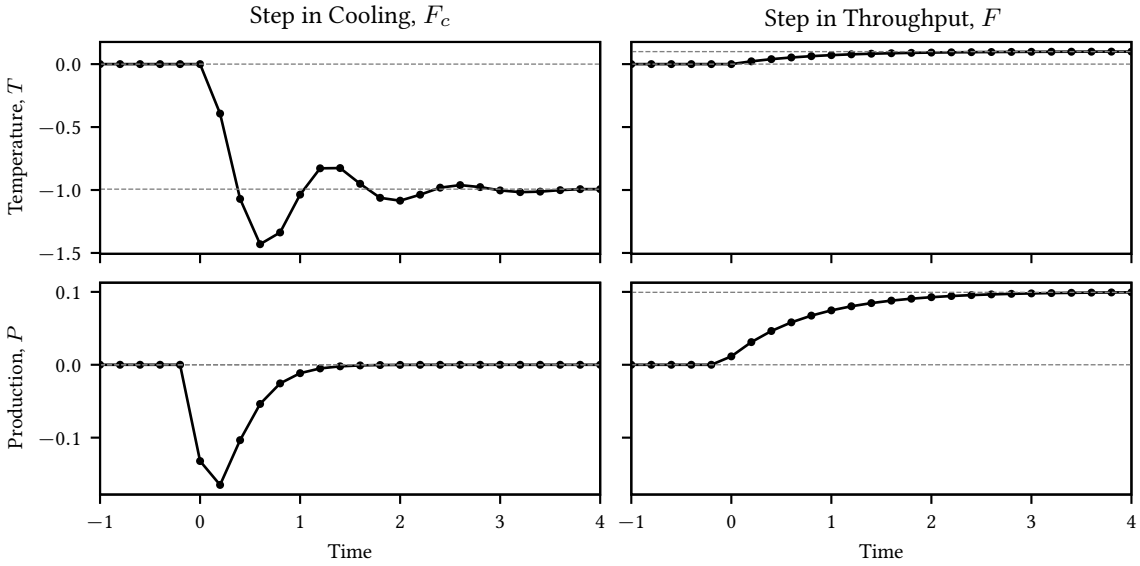


FIGURE 4.21: Step responses for underlying reactor dynamic model.

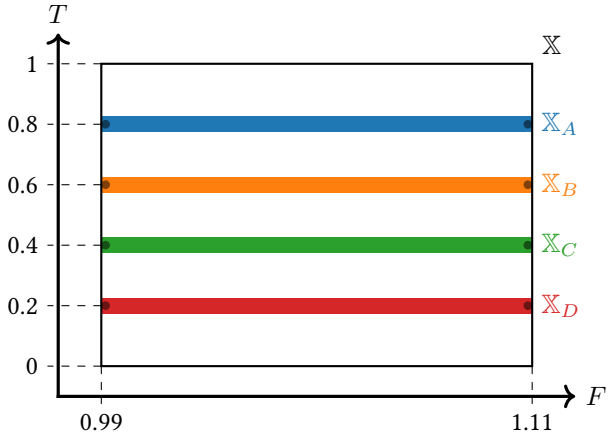


FIGURE 4.22: Operating point sets $\mathbb{X}_i$ for the reactor. Each $\mathbb{X}_i$ has a span of ± 0.1 temperature units. Circular markers show operating points used in the approximate model.

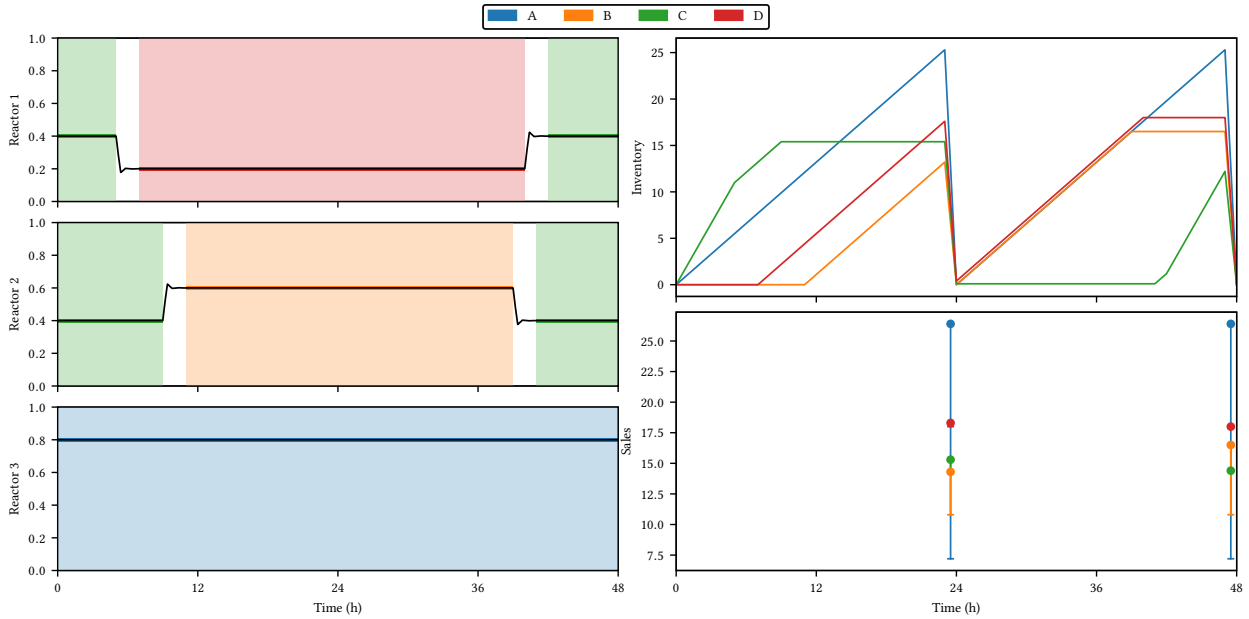


FIGURE 4.23: Solution to the approximate dynamic-aware model for the three-reactor system with dynamics. The unnecessarily long transition times of the approximate model are apparent.

Thus, the integrated model may be able to find faster solutions, as reactors are not required to reach an exact steady state before starting production.

We begin by solving the approximate model to obtain a feasible product transition schedule. We use this solution as a guess for the binary W_{ijt} variables in the integrated solution. The optimizer is then given time to find a better solution. We present the two solutions in Figures 4.23 and 4.24 respectively. From these figures, we see that the unnecessarily long transition times in the approximate model are the main source of suboptimality. Using the initial guess from the approximate model immediately leads to a solution to the full solution that is 1% better; after 10,000 s of optimization time, the final full-model solution is 3% better than than the approximate solution. However, even this final trajectory has some obvious suboptimalities, and the true optimal solution is likely a few percent better.

To conclude this example, we compare the solution times for the approximate model and the full model with and without the feasible initial guess. Solution progress is shown in Figure 4.25. From this figure, we see quite clearly that the approximate model is significantly easier to solve than the full model. The solver obtains a quality suboptimal solution to the approximate model within 10 s, and the optimal solution is found after 150 s. When the full model is given feasible values of W_{ijt} (obtained from the approximate solution) as an initial guess, the optimizer immediately finds a slightly better solution (due to the full model's slightly larger operating region). Unfortunately, solution progress is slow after that point, not finding a better solution until about 500 s. By contrast, when the full model is not given a feasible guess, lower bound progress is largely unaffected, but unfortunately the optimizer is unable to

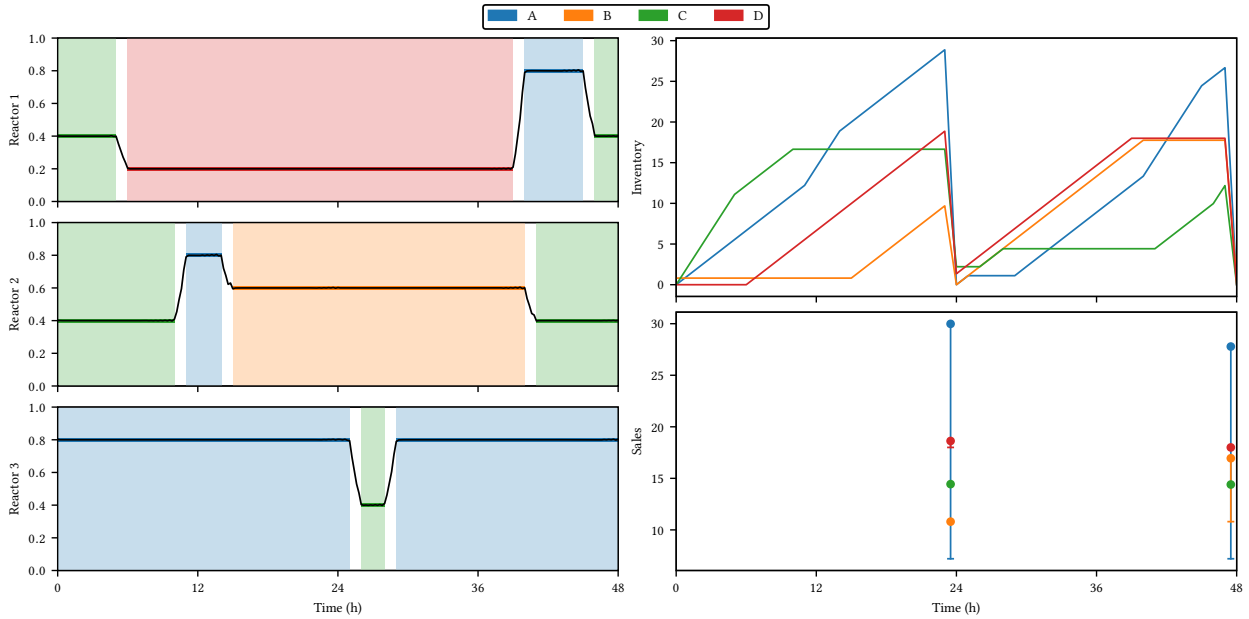


FIGURE 4.24: Solution to the full integrated model for the three-reactor system with dynamics. Some obvious suboptimalities can be observed, in particular the short run of product C in Reactor 3 that could be swapped to Reactor 2.

find a feasible solution after 10,000 s. This result indicates that either specialized techniques need to be developed to solve the combined formulation, or some form of approximation needs to be applied.

4.6 SUMMARY

In this chapter, we have presented a state-space formulation for closed-loop scheduling problems that can be used with economic MPC. Following Subramanian et al. (2012), we demonstrate that simple batch scheduling models can be put in state-space form by augmenting the state with a history of past task assignment variables. Through a simple example system, we have also demonstrated that naive rescheduling can lead to pathological closed-loop behavior, in this case a completely idle plant. However, by using a reference trajectory and a terminal cost/constraint, the theoretical results from Chapter 3 (which are compatible with the discrete decision variables prevalent in scheduling problems) provide nominal closed-loop performance bounds that prevent such behavior a-priori. We then present a state-space formulation for combined scheduling and control of units undergoing continuous production. This model is also compatible with economic MPC but can be challenging to solve. Thus, we have also presented a dynamic-aware scheduling model that can generate a feasible trajectory for the underlying unit dynamics without having to embed the full dynamic models. Finally, we have illustrated these methods using a system of parallel multi-product reactors. By including terminal constraints, closed-loop profit is increased relative to formulations without terminal constraints,

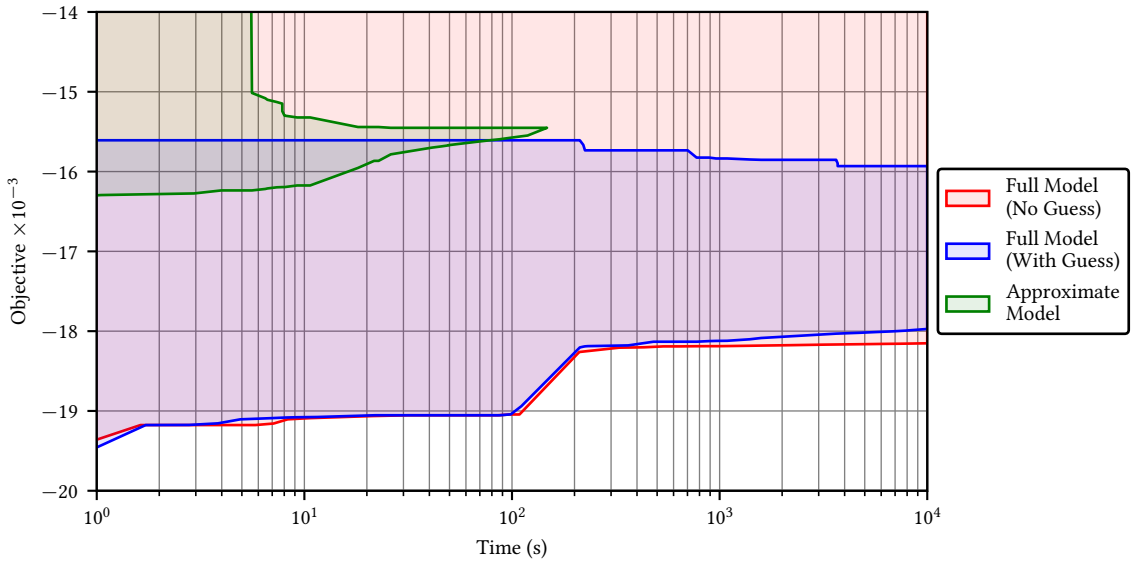


FIGURE 4.25: Upper and lower bound progress for the two dynamic reactor models. The objective function is to minimize negative profit. Note that, without the initial guess, the full model does not find a feasible solution within 10,000 s.

both in the nominal case and when disturbances are present. Thus, by formulating rescheduling problems as MPC, theoretical closed-loop properties can be derived to avoid unexpected behavior.

For some additional discussion of closed-loop scheduling, see the introduction to Risbeck et al. (2017). For additional closed-loop scheduling and economic MPC simulation results, see Risbeck et al. (2018a).

4.7 Appendix: Conservative Resampling

In Section 4.4, we proposed a dynamic-aware scheduling model that can generate feasible dynamic trajectories without including the full unit dynamic models. This formulation generally gets rid of the underlying dynamic s timescale, which reduces the necessary number of decision variables. However, in the inventory and production constraints (4.11), (4.12) and (4.27) still require variables and constraints on the s timescale. When the s discretization is very fine, the total number of points S can be very large, and thus the large number of variables and constraints can lead to slow solution times.

To alleviate this burden, we describe a *conservative resampling* procedure that allows the $\zeta^{\pm}_{ii'jnks}$ parameters to be redefined on a slower timescale in order to ensure satisfaction of the F and L constraints. To start, we define a new dynamic time grid $s \in \mathbf{S}' \coloneqq \{0, 1, \dots, S'-1\}$ with $0 < S' \leq S$. The constraints on F and L will be enforced on this timescale; thus, by choosing $S' < S$, the total number of constraints will be reduced, at a cost of a smaller feasible set. For each $s' \in \mathbf{S}'$, choose a value $\sigma_{s'} \in \mathbf{S}$ such that $\sigma_0 = 0$ and $\sigma_{s'+1} > \sigma_{s'}$. Define $\sigma_{S'} \coloneqq S$ and let $\bar{\mathbf{S}}' \coloneqq \mathbf{S}' \cup \{S'\}$ analogous to $\bar{\mathbf{S}}$.

For notational simplicity, we neglect the contribution of production and consumption for units at a steady-state operating point. These values are given by $\phi^{\pm}_{ijkp}G'_{ijpt}$ in (4.27). Since they are constant throughout the production range, they can be resampled without any loss of fidelity simply by rescaling based on the effective timesteps $\sigma_{s'+1} - \sigma_{s'}$. Thus, we consider the modified version of (4.27) as follows:

$$F^{\pm}_{kst} = \sum_{j\in\mathbf{J}} \sum_{i,i'\in\mathbf{I}} \sum_{m\in\mathbf{M}_{ii'j}} \zeta^{\pm}_{ii'jkmns} Z'_{ii'jmnt}, \quad k \in \mathbf{K}, s \in \mathbf{S}, t \in \mathbf{T}. \tag{4.28}$$

Here, the only contributions are due to the transition values $\zeta^{\pm}_{ii'jkmns}$.

4.7.1 Inventory Constraints

For the constraints on L_{skt}, define the parameter

$$\lambda_{ii'jkns} \coloneqq \sum_{s''=0}^{s-1} (\zeta^{+}_{ii'jkns} - \zeta^{-}_{ii'jkns})$$

Thus, on the s timescale, we have from (4.11) that

$$L_{kst} = L_{k0t} + \sum_{i,i'\in\mathbf{I}} \sum_{j\in\mathbf{J}} \sum_{n\in\mathbf{N}_{ii'j}} \lambda_{ii'jkns}.$$

Letting $\lambda^{*}_{ii'jkn} \coloneqq \lambda_{ii'jknS}$, the inventory balance on the t timescale becomes

$$L_{k(t+1)} = L_{kt} + \sum_{i,i'\in\mathbf{I}} \sum_{j\in\mathbf{J}} \sum_{n\in\mathbf{N}} \lambda^{*}_{ii'jkn} z'_{ii'jnt} + V^{+}_{kt} - V^{-}_{kt}.$$

This equation is exact and remains the same regardless of resampling. Thus, the bounds L_{kst} at the endpoints $s = 0$ and $s = S$ can be enforced via bounds on L_{kt} and $L_{k(t+1)}$. To address the interior points, we make use of the new timescale as follows. For each $s' \in \bar{\mathbf{S}}'$, choose parameters $\lambda^{\min}_{ii'jkns'}$ and $\lambda^{\max}_{ii'jkns'}$ such that

$$\lambda_{ii'jkns} \geq \lambda^{\min}_{ii'jkns'} + \frac{s - \sigma_{s'}}{\sigma_{s'+1} - \sigma_{s'}} (\lambda^{\min}_{ii'jkn(s'+1)} - \lambda^{\min}_{ii'jkns'})$$
$$\lambda_{ii'jkns} \leq \lambda^{\max}_{ii'jkns'} + \frac{s - \sigma_{s'}}{\sigma_{s'+1} - \sigma_{s'}} (\lambda^{\max}_{ii'jkn(s'+1)} - \lambda^{\max}_{ii'jkns'})$$

for all s satisfying $\sigma_{s'} \leq s \leq \sigma_{s'+1}$. That is, the $\lambda^{\min}_{ii'jkns'}$ and $\lambda^{\max}_{ii'jkns'}$ are chosen so that they are piecewise-linear under- and over-approximators of the original net production rates.

With these values, enforcing the constraints

$$L_{kt} + \sum_{i,i'\in\mathbf{I}} \sum_{j\in\mathbf{J}} \sum_{n\in\mathbf{N}} \lambda^{\min}_{ii'jkns'} \geq 0,$$

on the resampled $s' \in \bar{\mathbf{S}}'$ timescale ensures that $L_{kst} \geq 0$ for all $s \in \bar{\mathbf{S}}$ on the original dynamic timescale. Similar expressions using $\lambda^{\max}_{ii'jkns'}$ ensure satisfaction of upper bounds $L_{kst} \leq \bar{L}_k$. We illustrate this procedure in Figure 4.27. As shown in this example, the initial values of $\lambda^{\min}_{ii'jkns}$ and $\lambda^{\max}_{ii'jkns}$ for $s' = 0$ can potentially be nonzero to give a tighter overall approximation.

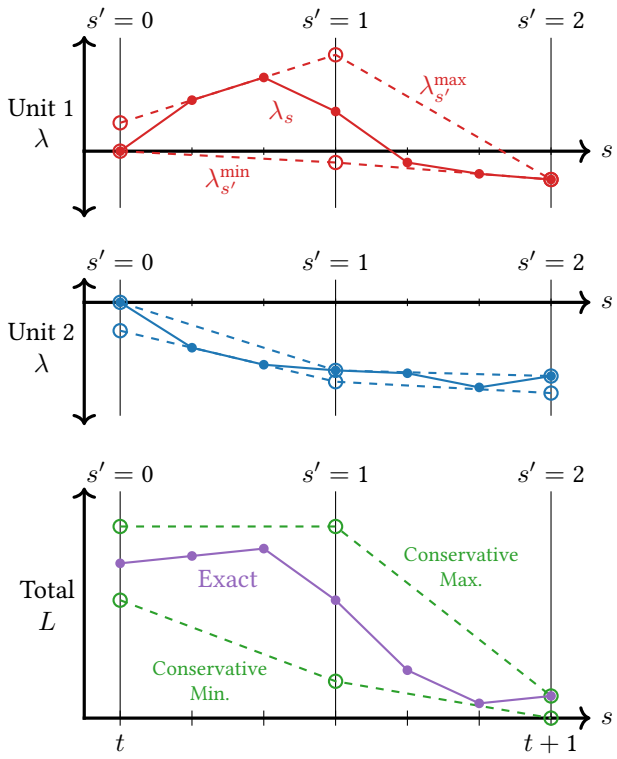


FIGURE 4.26: Illustration of conservative inventory constraints. The top two axes show cumulative net resource production for individual units, while bottom axes shows total inventory level. Using the conservative bounds, constraints can be enforced via constraints on the resampled s' timescale, rather than the true s timescale.

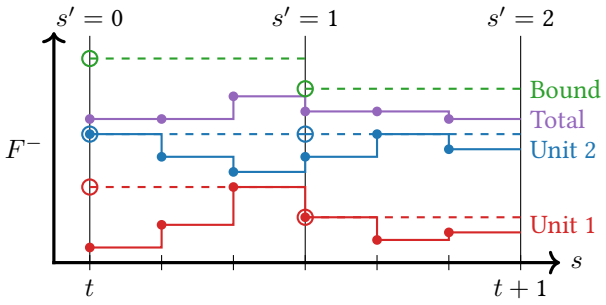


FIGURE 4.27: Illustration of conservative bound on resource consumption. Solid lines show actual $\zeta^-_{ii'jkns}$, while dashed lines show resampled $\hat{\zeta}^-_{ii'jkns'}$.

4.7.2 PRODUCTION/CONSUMPTION RATE CONSTRAINTS

For the constraints, on F, we recall that we would like to enforce bounds on F^+_{kst} and F^-_{kst} as defined in (4.28). To enforce these constraints, we define parameters

$$\hat{\zeta}^+_{ii'jkns'} \coloneqq \max_{s \in \hat{\mathbf{S}}_{s'}} \zeta^+_{ii'jkns}$$
$$\hat{\zeta}^-_{ii'jkns'} \coloneqq \max_{s \in \hat{\mathbf{S}}_{s'}} \zeta^-_{ii'jkns}$$

for sets

$$\hat{\mathbf{S}}_{s'} \coloneqq \{s \in \mathbf{S} \mid \sigma_{s'} \leq s < \sigma_{s'+1}\}.$$

Therefore, enforcing the constraint

$$\sum_{i,i' \in \mathbf{I}} \sum_{j \in \mathbf{J}} \sum_{n \in \mathbf{N}_{ii'j}} \hat{\zeta}^+_{ii'jkns'} z'_{ii'jnt} \leq \bar{P}^+_k$$

enforces that $F^+_{kst} \leq \bar{F}^+_k$ for all $s \in \hat{\mathbf{S}}_{s'}$. Similar expressions are written for F^-_{kst}. We illustrate in Figure 4.27. By enforcing (conservative) bounds on the s' timescale, original bounds on the s timescale are satisfied.

For the loser now
Will be later to win
For the [electricity prices] they are a-changin'

— Bob Dylan
The Times They Are A-Changin'

Chapter 5

Central Energy Plant Optimization

5.1 Introduction

As a specific application of the techniques developed in Chapters 2 and 3, we consider the online optimization of central energy plants for large-scale HVAC systems. To heat or cool campuses or large buildings, it is most efficient to produce hot and chilled water in centralized high-capacity equipment, and then pump the water to where heating and/or cooling is needed. This equipment is often organized into a central plant that contains multiple parallel pieces of equipment to meet the heating and cooling needs of the building or buildings being served (Powell et al., 2013). Running the equipment requires electricity, which can be purchased from the utility market or generated on site. At any given time, the job of the central plant is to meet the primary heating and cooling demands of the system, and it is of course desirable to do so at the highest possible efficiency or lowest possible cost.

Due to variable equipment capacities, when demand is below the maximum capacity of the plant, there many degrees of freedom for meeting current demand. For example, when cooling demand is 10 MW and the central plant contains two chillers, each with a 10 MW capacity, the demand could be met by having both chillers operate at 50% capacity or by having one chiller operate at 100% with the other chiller shut off. However, due to variable equipment efficiency, different equipment configurations that supply the same total load may have vastly different utility requirements. In addition, equipment that couples directly with the ambient (e.g., cooling towers) experiences time-varying efficiency due to time-varying ambient conditions (Braun, 2007b). Thus the overall plant efficiency is determined both by the configuration of active equipment (which are decision variables) *and* by current ambient conditions (which are parameters).

A significant additional complication is that electricity and other utility prices are not necessarily constant. Large customers are often assessed time-varying electricity prices, which can be either a fixed schedule or based on real-time market conditions (Albadi and El-Saadany, 2007). Thus, purchasing 1 MWh of electricity at 12:00 noon may be significantly more expensive than purchasing the same amount at 12:00 midnight. In addition to time-varying use charges, the monthly utility bill may contain a peak demand charge, which is calculated from the peak instantaneous *rate* of electricity usage over the previous period (Berg and Savvides, 1983). Unfortunately these two charges are competing: use charges are minimized by purchasing large amounts of electricity at various points throughout the day when it is cheap, while demand charges are minimized by purchasing a small amount of electricity consistently. Therefore, the optimal strategy lies somewhere between these two extremes.

Due to the two sources of time-variability, it is advantageous to install some form of thermal energy storage (TES). Coming in the form of ice or water tanks, TES allows the production of the central plant to be temporally decoupled from consumption by the buildings (Touretzky and Baldea, 2016). For example, a well-insulated tank allows chilled water to be produced at night when electricity is cheap and equipment is efficient, stored until the next afternoon, and then used to meet the high afternoon demand. This time shifting reduces the need to purchase electricity or operate equipment when electricity is expensive and/or efficiency is low, which can significantly reduce the cost of operating the central plant.

Finally, due to the large overlap between hot and chilled water demand (as was illustrated in Figure 1.6a), overall plant efficiency can be improved by employing heat-recovery chillers (HRCs). These units operate similarly to conventional chillers except that they reject waste heat back into a process stream of hot water, rather than into a waste stream of cooling water that must be processed in cooling towers. Therefore, HRCs produce hot water essentially for free, but they can only do so if there is simultaneous demand for both the hot and chilled water streams (Wenzel et al., 2014).

As a result of all of these factors, operating central plants at the lowest possible cost requires making a very large number of non-obvious and at times non-intuitive decisions: at each hour throughout the day, an operator must decide which equipment to activate, what level to operate each piece of active equipment, and how much water to send to or withdraw from TES. Unfortunately, the most common strategy used in existing central facilities is for plant managers to schedule these decisions by hand based on heuristic methods or prior intuition. Although very experienced operators may achieve adequate performance under nominal conditions, efficiency can suffer significantly when disturbances occur (e.g., atypical ambient conditions, units taken out of service for maintenance, or the main operator going on vacation) (Wenzel et al., 2016). To alleviate this human burden, we propose the application of mixed-integer economic MPC for this problem. Using models for equipment and TES, as well as forecasts for demand and utility prices, the cost-optimal schedule can be determined for a fixed horizon, and the decisions can be implemented in closed loop. The end result is more consistent performance and lower cost.

5.1.1 Problem Statement

Based on the previous discussion, we wish to formulate an optimization problem to make the following decisions:

- Is each chiller, HRC, boiler, or piece of auxiliary equipment active?
- For each active unit, what is its operating point?
- How much energy is being charged or discharged from TES?

These decisions must be made at each time point on a pre-defined discrete time grid. Once these decisions have been made, models for utility consumption of each unit are used to calculate utility requirements, which then determine operating costs. The goal is to minimize operating costs while respecting the following constraints:

- Heating and cooling demand must be satisfied.
- Equipment cannot be switched on or off too rapidly.
- Minimum and maximum capacities for active equipment cannot be violated.
- TES capacity must not be exceeded.

It is assumed that forecasts of reasonable accuracy are available for utility prices and heating/cooling demand. Such forecasts can be obtained from historical and real-time data (ElBsat and Wenzel, 2016; Zavala et al., 2009). For our purposes, we do not care how forecasts are obtained but simply that they are available. Of course, these forecasts can be inaccurate and may be continuously updated; thus, it is necessary that the optimization problem can be solved in real time, so that a receding horizon implementation can be used.

There are a number of features that make this problem particularly challenging. Among them are as follows:

- The decisions are both discrete (e.g., on/off) and continuous (e.g., how much) in nature.
- Units have nonzero minimum capacities and efficiencies that vary with load, which give rise to nonlinear, nonconvex equipment models.
- The cost function includes both short-term time-varying prices and long-term peak demand charges.
- TES capacity is potentially very large, so long prediction horizons are necessary for most effective use.

Because of these difficulties, many strategies in the literature either propose near-optimal heuristics, focus on smaller-scale instances, or decompose the problem. In order to achieve the best possible performance, we strive to not make any of these compromises and instead focus on a formulation of the complete problem that is tractable for realistically sized central plants.

5.1.2 Literature Review

Due to the presence of both discrete and continuous decision variables, a natural tool is mixed-integer programming (MIP). A special subclass of MIP is mixed-integer linear programming (MILP), in which the objective function and constraints are all linear functions of the decision variables. Although MILP can be very challenging, many large instances can be solved using state-of-the-art solvers such as Gurobi (Gurobi Optimization, LLC, 2018) and SCIP (Achterberg, 2009). MILP optimization models have been employed for various applications in building efficiency and sustainability including selection of optimal energy source across renewable and non-renewable options (Ashouri et al., 2013), using life-cycle analysis to optimize building retrofits (Gustafsson, 1998), and scheduling appliance usage across households in a microgrid (Zhang et al., 2013). More general mixed-integer nonlinear programming (MINLP) allows the inclusion of nonlinear constraints. General MINLP remains challenging, although progress is being made (Belotti et al., 2013), and solvers like BARON (Sahinidis, 2018) can solve some moderately sized instances to optimality. MINLP formulations can consider equpiment models or energy balances in more detail, (Candanedo et al., 2013; Dagdougui et al., 2012; Hajiah and Krarti, 2012; Ma et al., 2012b; Touretzky and Baldea, 2016; Trifkovic et al., 2014). However, such models are generally limited to small systems with few pieces of equipment and/or few degrees of freedom.

Because of challenges associated with large-scale MIP problems, many decomposition strategies have been proposed for central energy plant optimization. A common architecture is to delegate discrete equipment selection to a small subproblem, either via heuristics (Braun, 2007a,b; Braun and Diderrich, 1990) or solving a small MIP problem for single-period equipment selection (Ali et al., 2013; Ardakani et al., 2008; Geem, 2011; Lee et al., 2011; Lu et al., 2004). These subproblems can then be used to construct an aggregate representation of plant performance without reference to individual equipment on/off states (Powell et al., 2013; Wenzel et al., 2014), which allows longer-horizon TES optimization to be formulated using all continuous variables. Depending on the model used for TES, it may be necessary to include a discrete variable to choose whether the TES is charging or discharging, e.g., if the model is asymmetric. Similarly to the previous strategies, methods such as dynamic programming (Henze et al., 2008) or backward reach set computation (Behl et al., 2012) can be used to predetermine these variables, which also serves to reduce or remove discrete decision variables from further optimization. However, because these decompositions generally lead to temporal decoupling of on/off variables, explicit constraints on rapid equipment switching cannot be included. While aggregate rate-of-change penalties could discourage rapid switching, it is nevertheless possible that units are switched on and off repeatedly in response to small system load variations. In addition, although some formulations address simultaneous heating and cooling (Wenzel et al., 2014), most formulations consider only chilled water production and sometimes only a single chiller.

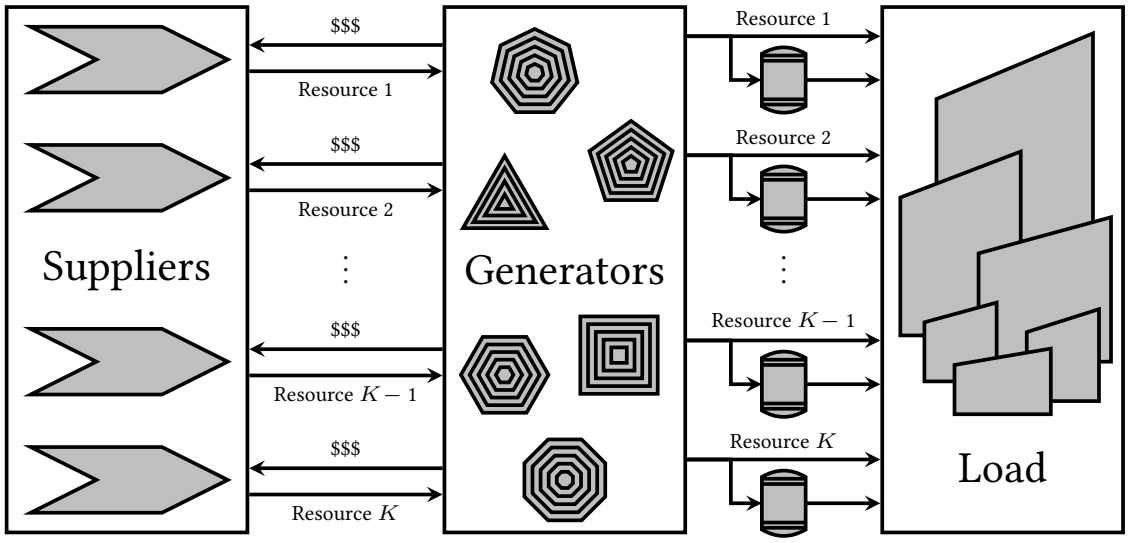


Figure 5.1: Abstract representation of central energy plant optimization model.

To address the limitations of existing strategies, we develop an MILP formulation to address the entire central energy plant optimization problem without decomposing into subproblems. We demonstrate that this formulation is tractable for realistically-sized instances, and we develop further extensions to increase fidelity and improve closed-loop performance. Note that in this section, we are only considering operation of the central plant in order to meet a fixed schedule (or forecast) of demand. Many strategies have been proposed in the literature for small-scale systems in which the cooling demand is not a parameter but rather a decision variable that couples with a dynamic model for building temperature. We hold off on reviewing these works until Chapter 6.

5.2 Problem Formulation

To facilitate a concise problem formulation, we consider an abstract representation of the central plant in terms of "generators" and "resources". In this language, a generator is any piece of equipment (main or auxiliary), while a resource is any material or energy source that flows into or out of the plant. The central plant operates by purchasing some set of resources from the utility market, consuming those resources in generators to produce different resources, and then delivering the produced resources to the load (i.e., buildings), possibly passing through storage along the way. This abstract representation is diagrammed in Figure 5.1. Note that this model originally appeared in Risbeck et al. (2015) and was expanded in Risbeck et al. (2017).

Throughout this section, we use boldface upper-case Roman letters for sets (e.g., $\mathbf{T}$), lower-case Roman letters for subscripts, upper-case Roman letters for decision variables, and Greek letters for parameters.

5.2.1 Sets

For the abstract problem definition, we use the following sets:

- Generator types $j \in \mathbf{J}$
- Resources $k \in \mathbf{K}$
- Time intervals $t \in \mathbf{T}$

Note that the set $\mathbf{T} \coloneqq \{1, 2, \ldots, T\}$ is ordered, and the (constant) length of each time period is denoted by Δ. To calculate peak charges assessed over multiple (possibly overlapping) windows, we define the following:

- Peak charge windows $w \in \mathbf{W}$
- Subset of time points $\mathbf{T}_w \subseteq \mathbf{T}$ included in window w

Finally, for the auxiliary variables used for piecewise-linear generator models, we define the following sets:

- Interpolation regions $m \in \mathbf{M}$
- Subset of interpolation regions $\mathbf{M}_j \subseteq \mathbf{M}$ for generator j
- Interpolation points $n \in \mathbf{N}$
- Subset of interpolation points $\mathbf{N}_{jm}$ in region m for generator j

These sets are explained more in depth in Section 5.3.2.

5.2.2 Parameters

In this section, we list all model parameters. Most parameters are real-valued, although parameters restricted to be integers are noted. Subscripts on all parameters are as in the previous section.

To calculate the cost of a given schedule, it is necessary to know (or have a forecast of) resource prices and demands. These and other parameters used to define the cost function:

- Forecast ϕ_{kt} of resource demand
- Forecast ρ_{kt} of resource prices
- Peak demand charge costs ρ^{max}_{kw}
- Minimum bound Π^{max}_{kw} for peak demand
- Penalty ω_k for backlogged demand
- Upper bounds Π_k for resource purchase

Other parameters are used to predict equipment performance or constrain equipment usage:

- Number μ_j of identical type-j generators (integer-valued)
- Interpolation points ζ_{jknt} for generator models
- Equipment on, off restrictions η_{jt}^{+}, η_{jt}^{-} (integer-valued)
- Decay rate σ_k for storage tanks
- Minimum on, off dwell times δ_j^{+}, δ_j^{-} for generators (integer-valued)
- Bound Υ_k for single-period storage charge/discharge
- Storage capacity Σ_k

These parameters may be updated as equipment characteristics change, but they are generally known values.

5.2.3 Variables

In this section, we list all decision variables in the model. Subscripts correspond to the appropriate sets. Bounds and integrality restrictions are given for each variable.

Analogous to the parameters, the following variables are used to calculate costs:

- Resource purchase rates $P_{kt} \in [0, \Pi_k]$
- Peak resource purchase rates $P_{kw}^{\max} \in [\Pi_{kw}^{\max}, \infty)$
- Total cost $C_k \in [0, \infty)$ of resource k
- Backlog change $H_{kt} \in (-\infty, \phi_{kt}]$
- Cumulative backlog level $B_{kt} \in [0, \infty)$

Equipment operation is determined by the following decision variables:

- Number $U_{jt} \in \{0, \ldots, \mu_j\}$ of type-j generators currently on
- Number $U_{jt}^{+}, U_{jt}^{-} \in \{0, \ldots, \mu_j\}$ of generators newly switched on, off
- Net resource consumption $Q_{jkt} \in (-\infty, \infty)$
- Net storage discharge rate $Y_{kt} \in [-\Upsilon_k, \Upsilon_k]$
- Storage level $S_{kt} \in [0, \Sigma_k]$

Finally, to model piecewise-linear operating surfaces for generators, the following auxiliary decision variables are used:

- Number of generators $V_{jmt} \in \{0, \ldots, \mu_j\}$ operating in interpoltaion region m
- Weight $Z_{jmnt} \in [0, \mu_j]$ of interpolation point n in region m

Note that variables B_{kt}, U_{kt}, and S_{kt} represent states of the system, and thus they possess initial values B_{k0}, U_{k0}, and S_{k0} that are fixed parameters for the purposes of optimization. All other variables are inputs to the system (or auxiliary optimization variables) and and thus do not require values for $t = 0$.

5.2.4 Constraints

The overall objective function is the sum of real utility costs and fictitious backlog penalties as follows:

$$\min \quad \sum_{k \in \mathbf{K}} C_k + \sum_{k \in \mathbf{K}} \sum_{t \in \mathbf{T}} \omega_{kt} B_{kt}. \tag{5.1}$$

Note that the penalty coefficients ω_k should be set to large values so that $B_{kt} = 0$ whenever feasible. In such cases, the objective function is simply the real utility costs that would be assessed in the current schedule. Real utility cost is calculated via

$$C_k = \sum_{t \in \mathbf{T}} \rho_{kt} P_{kt} + \sum_{w \in \mathbf{W}} \rho_{kw}^{\text{max}} P_{kw}^{\text{max}}, \quad k \in \mathbf{K}. \tag{5.2}$$

This constraint gives the sum of time-varying use charges and peak demand charges, with the peak rates calculated using

$$P_{kw}^{\text{max}} \geq P_{kt}, \quad w \in \mathbf{W}, t \in \mathbf{T}_w. \tag{5.3}$$

Recall that P_{kw}^{max} has a lower bound Π_{kw}^{max}. As mentioned in Section 3.4, it is desirable to have a good estimate for the nominal remaining peak to use as this bound; that way, the true peak cost ρ_{kw}^{max} can then be included in the objective function. In the absence of such an estimate, one should re-weight the peak charge term in the objective function based on the current horizon so that it does not dominate and lead to an unsustainably low peak. In either case, the peak observed so far is included in the bound Π_{kw}^{max} to avoid violating the Principle of Optimality.

The main operating constraint of the central plant is that all resource demand must be satisfied. To ensure that each optimization problem has a solution, we allow the optimizer to backlog demand (albeit at a heavy penalty). Mathematically, this constraint is expressed as

$$\sum_{j \in \mathbf{J}} Q_{jkt} + Y_{kt} + P_{kt} + H_{kt} \geq \phi_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}, \tag{5.4}$$

which requires demand ϕ_{kt} to be met by the sum of current production Q_{jkt}, storage discharge Y_{kt}, direct purchase P_{kt}, and backlog H_{kt}. The inclusion of the bound $H_{kt} \leq \phi_{kt}$ ensures that only external demand (i.e., not *internal* demand due to consumption of resources in generators) can be backlogged. For example, to enforce that a sufficient number of pumps are active, chillers are modeled as consuming an internal "volumetric flow" resource that is produced

by pumps. Because there is no external demand for volumetric flow, the upper bound on H_{kt} ensures that the optimizer must activate the required number of pumps whenever chillers are active; without the bound, the optimizer could potentially choose to leave the pumps idle and incur the backlog penalty.

Equipment on/off switching is determined by the following three constraints:

$$U_{jt} = U_{j(t-1)} + U^+_{jt} - U^-_{jt}, \quad j \in \mathbf{J}, t \in \mathbf{T}, \tag{5.5}$$

$$U_{jt} \geq \eta^+_{jt} + \sum_{\tau=0}^{\delta^+_j - 1} U^+_{j(t-\tau)}, \quad j \in \mathbf{J}, t \in \mathbf{T}, \tag{5.6}$$

$$\mu_j - U_{jt} \geq \eta^-_{jt} + \sum_{\tau=0}^{\delta^-_j - 1} U^-_{j(t-\tau)}, \quad j \in \mathbf{J}, t \in \mathbf{T}. \tag{5.7}$$

Here, the first constraint updates the number of on and off units due to current switching events, while the second and third equations enforce lower bounds on the number of on and off units respectively (note that the expression $\mu_j - U_{jt}$ gives the number of type-j generators that are switched off in period t). The parameters η^+_{jt} and η^-_{jt} are used to account for switching events that take place outside of the current optimization horizon, and they ensure that dwell times are actually enforced in the closed-loop schedule.

Equipment production rates are calculated according to

$$Q_{jkt} = \sum_{m \in \mathbf{M}_j} \sum_{n \in \mathbf{N}_{jm}} \zeta_{jknt} Z_{jmnt}, \quad j \in \mathbf{J}, k \in \mathbf{K}, t \in \mathbf{T}. \tag{5.8}$$

To ensure that the weights Z_{jmnt} define a valid point on the piecewise-linear operating surface, the following constraints are enforced:

$$U_{jt} = \sum_{m \in \mathbf{M}_j} V_{jmt}, \quad j \in \mathbf{J}, t \in \mathbf{T}, \tag{5.9}$$

$$V_{jmt} = \sum_{n \in \mathbf{N}_{jm}} Z_{jmnt}, \quad j \in \mathbf{J}, m \in \mathbf{M}_j, t \in \mathbf{T}. \tag{5.10}$$

The first constraint enforces that one interpolation region is selected for each active unit, while the second constraint ensures that a convex combination of points in region m is selected.

Storage tank dynamics are modeled using the simple first-order model

$$S_{kt} = \sigma_k S_{k(t-1)} - Y_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}. \tag{5.11}$$

For a perfectly insulated tank, $\sigma_k = 1$. Conceptually, this model is the Δ-discretization of the continuous-time model $dS_k/dt = (\log(\sigma_k)/\Delta)S_k - (1/\Delta)Y_{kt}$. In the true discretization, there is a coefficient of $(\sigma_k - 1)/\log(\sigma_k)$ in front of Y_{kt} in (5.11). This term can be included for increased accuracy if desired (and it approaches unity as $\sigma_k \to 1$).

Finally, cumulative backlog is modeled as a simple integrator

$$B_{kt} = B_{k(t-1)} + H_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}. \tag{5.12}$$

By tracking and penalizing *cumulative* backlog, any missed demand in a given period must be met in a later period or face additional penalty. For example, if the demand for cooling cannot be met, then building temperature will increase above nominal levels; thus, additional cooling above the nominal demand is necessary in the following period to restore nominal temperatures. However, depending on the nature of the demand, meeting demand later may not be a viable option. For example, if there is missed demand for electricity that is used for building lighting, then the lights will simply shut off, and there is no need for additional electricity in future periods. In such cases, backlog need not be tracked, and $H_{kt} \in [0, \phi_{kt}]$ can be penalized directly in the objective function.

The overall optimization problem consists of constraints (5.2) to (5.12) with objective function (5.1). It is a mixed-integer linear programming (MILP) problem.

5.2.5 Remarks

Remark 5.1: A key premise of the model is that generators operate at steady state throughout each time period, and jumps to new steady states happen instantaneously. In reality, there are transient dynamics associated with changing operating points, especially when a generator is switched on or off. Fortunately, with a timestep of 1 h, there is usually sufficient timescale separation for the dynamics of part-load changes, which are on the timescale of minutes, and thus they can be safely neglected at this resolution. For startup and shutdown dynamics, effects are more pronounced, but their impact is minimized by enforcing dwell times on equipment to prevent switching on and then immediately off in the subsequent period.

Should a smaller timestep be utilized, these effects could be attenuated via rate-of-change constraints, e.g.,

$$|Q_{jkt} - Q_{jk(t-1)}| \leq \xi_{jk} + \xi^{+}_{jk} U^{+}_{jt} + \xi^{-}_{jk} U^{-}_{jt}, \quad j \in \mathbf{J}, k \in \mathbf{K}, t \in \mathbf{T}.$$

In this form, ξ_{jk} gives the maximum one-period change in Q_{jkt} when the unit is on in both periods, while ξ^{+}_{jt} and ξ^{-}_{jt} allow this bound to be tightened or loosened when the generator has just been switched on or off.

Remark 5.2: As written, the optimization problem is not in state-space form due to the constraints (5.6) and (5.7) that consider variables across time points. However, as discussed in Chapter 4, the model can be transcribed into state-space form by "lifting" the necessary input variables (that is, augmenting the state to include a finite history of past values). Following Section 3.4, the state is also augmented to include the past history for peak demand charges. After making this change, the variables are categorized as in Table 5.1. Recall that auxiliary variables are used to aid in the formulation of the optimization problem and are uniquely defined by some function (expressed as model

TABLE 5.1: State-space variable types for central plant optimization model.

Category	Variables
States	U_{jt}, S_{kt}, B_{kt}, past history of U_{jt}^{+} and U_{jt}^{-}, previous peak purchase $P_{kw}^{\max}$
Inputs	Q_{jkt}, Y_{kt}, P_{kt}, H_{kt}, U_{jt}^{+}, U_{jt}^{-}
Auxiliary Variables	C_k, $P_{kw}^{\max}$, V_{jmt}, Z_{jmnt}

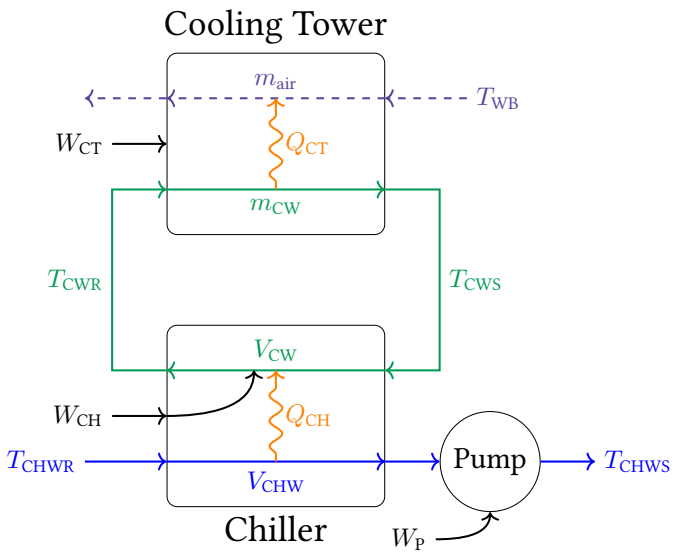
constraints) of the states and inputs. Note also the slightly different time indexing used in this section: states are indexed $t \in \{0, 1, \ldots, T\}$, while inputs and auxiliary variables are indexed as $t \in \{1, 2, \ldots, T\}$, which gives evolution equations of the form $x_t = f(x_{t-1}, u_t)$.

REMARK 5.3: Due to the inclusion of nonconvex equipment models, it can be difficult to close the optimality gap to 0%. However, because these models are approximate and certain parameters are forecasts, finding a solution whose objective function is 0.1% better does not necessarily mean that the corresponding schedule will be 0.1% better upon implementation. Thus, it is not worthwhile to spend a significant amount of time proving optimality for each instance. In addition, because the optimization is implemented in closed loop, minor imperfections in a given schedule can potentially be corrected at the next timestep, or they may even become irrelevant due to changes in forecasts. Therefore, for the purposes of this chapter, we define "solving" a given instance as finding a solution with an optimality gap of 1% or less within the given time limit. The typical solution progress is that a quality suboptimal solution (with a gap of 1% to 5%) is found within the first 30 s, followed by improvement over the next minute, after which the objective function stagnates, and progress is mainly on the solution lower bound (see Figure 5.14 for an example of this behavior). Fortunately, this pattern is very favorable with the desired performance.

5.3 EQUIPMENT MODELS

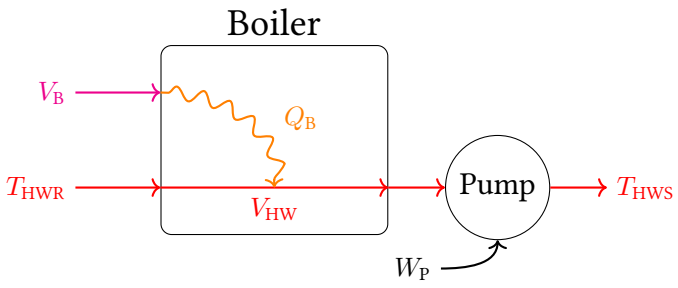
To illustrate the equipment models, we start by showing the energy and material flow for a chiller/pump/cooling tower system in Figure 5.2. After collecting heat from the buildings, chilled water returns and enters the chiller. Using electricity to power a refrigeration cycle, the chiller rejects heat Q_{CH} from the chilled water stream, thereby lowering its temperature from T_{CHWR} to T_{CHWS}. The rejected heat (along with a fraction of the electricity input) is transferred to cooling water stream, which is then sent to a cooling tower. Inside the cooling tower, the cooling water is contacted with ambient air, reducing the water temperature from T_{CWR} to T_{CWS}. The overall effect is that heat is rejected from the chilled water stream back to the ambient. Note that chiller performance depends on the temperatures T_{CHWR} and T_{CWS} of both entering water streams; as a simplifying assumption, these values can be taken as fixed parameters.

The material and energy flows for HRCs is identical to conventional chillers except that the heat is rejected directly into a process stream of hot water, rather than into a waste stream of cooling water. Thus, the heat is ultimately



Variables	
T_{WB}	Ambient wet-bulb temperature
T_{CWS}, T_{CWR}; m_{CW}, V_{CW}	Cooling water supply, return temperatures; mass, volume flows
T_{CHWS}, T_{CHWR}; V_{CHW}	Chilled water supply, return temperatures; volume flow
Q_{CH}, Q_{CT}	Cooling load for chiller, cooling tower
W_{CH}, W_{CT}, W_P	Electricity use by chiller, cooling tower, pump

FIGURE 5.2: Mass and energy flows for a single chiller, pump, and cooling tower. The chiller rejects heat from the return water supply into a stream of cooling water The cooling tower rejects heat from the cooling water supply back to the ambient.

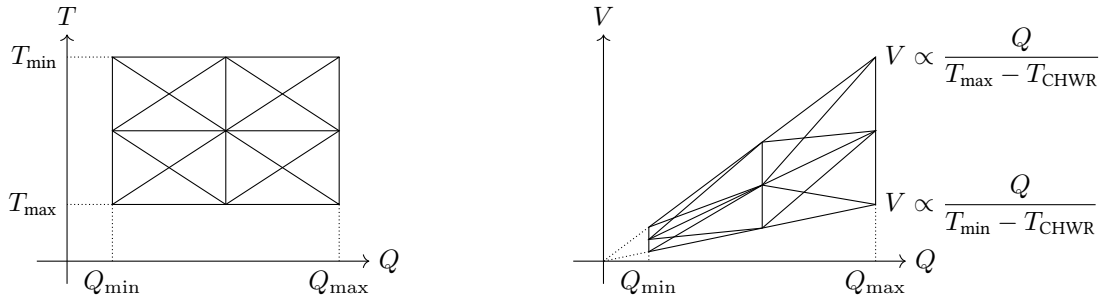


Variables	
V_B	Boiler natural gas use
T_{HWS}, T_{HWR}; V_{HW}	Hot water supply, return temperatures; volume flow
Q_B	Boiler heating load
W_P	Electricity use by pump

FIGURE 5.3: Mass and energy flows for a single boiler and pump. Natural gas is burned in the chiller to heat the hot water stream.

transferred to whichever process consumes the hot water. For direct heating, natural gas boilers are used. The boiler/pump system is diagrammed in Figure 5.3.

For the purposes of cost calculation, the "main" equipment is chillers, HRCs, and boilers. These units meet the heating and cooling demands placed on the central plant, and they account for the majority of resource use. All other units are referred to as "auxiliary" equipment, which includes cooling towers as well as pumps for hot and chilled water. To reduce computational requirements, the auxiliary equipment can be omitted from the formulation; in this case, the main equipment models should be augmented to include the external resource consumption of auxiliary equipment. We give the nonlinear functional forms for all equipment models in the following section.



(A) Domain and polytopic partition of chiller model in original $(Q_{\text{CH}}, T_{\text{CHWS}})$ space.

(B) Domain and polytopic partition of chiller model in transformed $(Q_{\text{CH}}, V_{\text{CHW}})$ space.

FIGURE 5.4: Domain of chiller model in original and transformed space. The symbols Q, T, and V refer to Q_{CH}, T_{CHWS}, and V_{CHW} respectively.

5.3.1 Nonlinear Models

Conventional and heat-recovery chillers are modeled using the semi-empirical Gordon-Ng model Lee et al. (2012), defined as follows:

$$W_{\text{CH}} \coloneqq \left(Q_{\text{CH}} + a_1 T_{\text{CHWS}} + a_2 \left(1 - \frac{T_{\text{CHWS}}}{T_{\text{CWS}}}\right)\right) \frac{T_{\text{CWS}}}{T_{\text{CHWS}} - a_3 Q_{\text{CH}}} - Q_{\text{CH}}. \tag{5.13}$$

The parameters a_1, a_2, and a_3 can be obtained regression using manufacturer or measured data, and they are different for each type of chiller. For the purposes of optimization, the cooling water supply temperature T_{CWS} is considered a fixed parameter. Chilled water supply temperature T_{CHWS} can be fixed as well, but it may be advantageous to optimize it as well. Unfortunately direct inclusion of T_{CHWS} as a variable is not compatible with the resource-balance-based formulation, as temperature is an intensive variable. However, since we know that chiller cooling load Q_{CH} is proportional to $V_{\text{CHW}}(T_{\text{CHWR}} - T_{\text{CHWS}})$, we can use the extensive variable $V_{\text{CHW}} \propto Q_{\text{CH}}/(T_{\text{CHWR}} - T_{\text{CHWS}})$ as a surrogate for T_{CHWS}. Note that T_{CHWR} is taken as a fixed parameter. We illustrate this transformation in Figure 5.4; the final model has two independent variables Q_{CH} and V_{CHW} as in Figure 5.4b. In the case of constant T_{CHWS}, the only independent variable is Q_{CH}, and V_{CHW} becomes a dependent variable calculated using the same relationship. In either case, chillers and HRCs have nonzero minimum capacity that must be satisfied for each active unit.

For direct heating, natural gas boilers are used. For simplicity, we assume that hot water return and supply temperatures T_{HWR} and T_{HWS} are fixed. If T_{HWS} can be varied, then V_{HW} can be used as an extensive surrogate variable as in the case of chillers. For natural gas consumption, we use a simple constant-efficiency model

$$Q_{\text{B}} \coloneqq \eta V_{\text{B}}, \tag{5.14}$$

in which η is the product of the heat of combustion for natural gas and the heat-transfer efficiency of the boiler. Note that we use the symbol V_{B} for natural gas flow because natural gas is typically sold on a volume basis. Minimum

capacity is set to 20% of maximum capacity. In the case more complicated heating equipment, a correspondingly more complicated model can be used. Because we focus on cooling, this simple model is sufficient for our purposes.

Heat transfer in cooling towers is modeled using a simplified effectiveness formula Jin et al. (2007), while electricity consumption is calculated using a simple cubic fit Braun and Diderrich (1990). These equations are as follows:

$$Q_{\text{CT}} \coloneqq \frac{c_1(m_{\text{CW}})^{c_3}}{1 + c_2\left(\frac{m_{\text{CW}}}{m_{\text{air}}}\right)^{c_3}}(T_{\text{CWR}} - T_{\text{WB}}) \tag{5.15a}$$

$$W_{\text{CT}} \coloneqq \kappa(m_{\text{air}})^3 \tag{5.15b}$$

With known T_{WB} and fixed m_{CW}, (5.15a) can be rearranged to solve for the required m_{air}, which is then used in (5.15b) for electricity calculation. Note that T_{CWR} is calculated from the relationship $Q_{\text{CT}} \propto m_{\text{CW}}(T_{\text{CWR}} - T_{\text{CWS}})$. Model coefficients c_1, c_2, c_3, and κ are obtained from data.

Finally, pumps are modeled with a black-box empirical model

$$W_{\text{P}} \coloneqq b_1 \ln(1 + b_2 V_{\text{P}}) + b_3 V_{\text{P}} + b_4, \tag{5.16}$$

in which V_{P} is either chilled water flow V_{CHW} or hot water flow V_{HW} depending on the stream of interest. Regression coefficients b_1 through b_4 are different for each type of pump. As with chillers and boilers, pumps have nonzero minimum capacity.

5.3.2 Piecewise Linear Formulation

To avoid an MINLP formulation for the optimization problem, we approximate the nonlinear equipment models from the previous section using piecewise-linear functions. Here, we mention only the information relevant to the piecewise-linear formulation in (5.8) to (5.10). A more general discussion of piecewise-linear approximation and its use in MILP can be bound in Section 5.6.

The nonlinear equipment models enforce a relationship among the Q_k variables for each piece of equipment and time point (subscripts j and t are omitted for clarity in this section). In general, the "operating surface" is a lower-dimensional set of the form

$$\mathcal{Q} \coloneqq \{0\} \cup \{(Q_1, \ldots, Q_K) : (Q_1, \ldots, Q_k) = f(Q_{k+1}, \ldots, Q_K)\}$$

for each generator. Note that the (typically isolated) point at zero is included for when a generator is off. We wish to construct a set $\tilde{\mathcal{Q}} \approx \mathcal{Q}$ that can be embedded in the MILP optimization model. Enforcing $(Q_1, \ldots, Q_K) \in \tilde{\mathcal{Q}}$ will ensure that the various resource production and consumption rates are calculated according to accurate piecewise-liner approximations of the true nonlinear model. We refer to the set $\tilde{\mathcal{Q}}$ as the piecewise-linear operating surface.

To determine the piecewise-linear operating surface, the first step is to identify which resources are independent variables, and which are dependent variables. For example, the chiller model in (5.13) gives $(Q_{k_3}, Q_{k_4}) = f(Q_{k_1}, Q_{k_2})$ and pump models (5.16) give $Q_{k_3} = f(Q_{k_2})$, both with resources $k_1 =$ Chilled Water, $k_2 =$ Chilled Water Volume, $k_3 =$ Electricity, and $k_4 =$ Cooling Water. Next, a grid is chosen over the independent variables, and the nonlinear function is evaluated at each lattice point $n \in \mathbf{N}$, stored as the parameter ζ_{kn}. We set $\zeta_{kn} < 0$ if resource k is being consumed, and $\zeta_{kn} > 0$ if resource k is being produced. Finally, the rectangular regions are triangulated to give a set of subdomains $m \in \mathbf{M}$, with the lattice-point vertices of each subdomain stored in the set $\mathbf{N}_m$. Accuracy can be adjusted for each piece of equipment by modifying the triangulations of each rectangular region, or by using a coarser or finer grid in the previous step.

Inside the optimization problem, we choose a point on the operating surface by taking a convex combination of the vertices n for a subdomain, as determined by the auxiliary variables Z_{mn}. To ensure that only vertices $n \in \mathbf{N}_m$ for the currently active subdomain are given nonzero weight, the discrete variables V_m are used, with the weights for each subdomain m summing to the (integer) value V_m according to (5.10). We illustrate the parameter, set, and variable definitions for a one-dimensional case in Figure 5.5. Note that for the one-dimensional case, no triangulation is required, while for three or more dimensions, each triangulation may have a different number of subdomains (see, e.g., Figure 5.22b).

To reduce symmetry in the model (which can increase the number of branches necessary to find an optimal solution), we use a symmetry-free formulation in the case of multiple identical generators. Because the model requires only the sum $\sum_j Q_{jk}$ in (5.4), we formulate the piecewise-linear model so that the variable Q_{jk} calculates *total* resource production and consumption by all generators of type j. This property is achieved simply by setting integer upper bounds on the variables U_{jt} and V_{jmt} so that operating points can be selected for each active generator.

5.4 Simulations

In the examples that follow, we use a discrete time grid with a sample time of 1 h. Although there is theoretically 1 h time available for each optimization, it is desirable to obtain solutions much more quickly to respond to disturbances or updated forecasts. Thus, to be of practical use, each optimization must be solvable within 10 to 15 min. All MILP problems are solved using Gurobi 8.0 (Gurobi Optimization, LLC, 2018) on a standard Linux desktop computer.

5.4.1 Basic Model

To start, we solve the optimization problem for a central plant that provides heating and cooling. As illustrated in Figure 5.6, the central plant consists of four chillers, two HRCs, one boiler, four chilled water pumps, two hot water pumps, and five cooling towers. Equipment models are identical within the same equipment type. The prediction

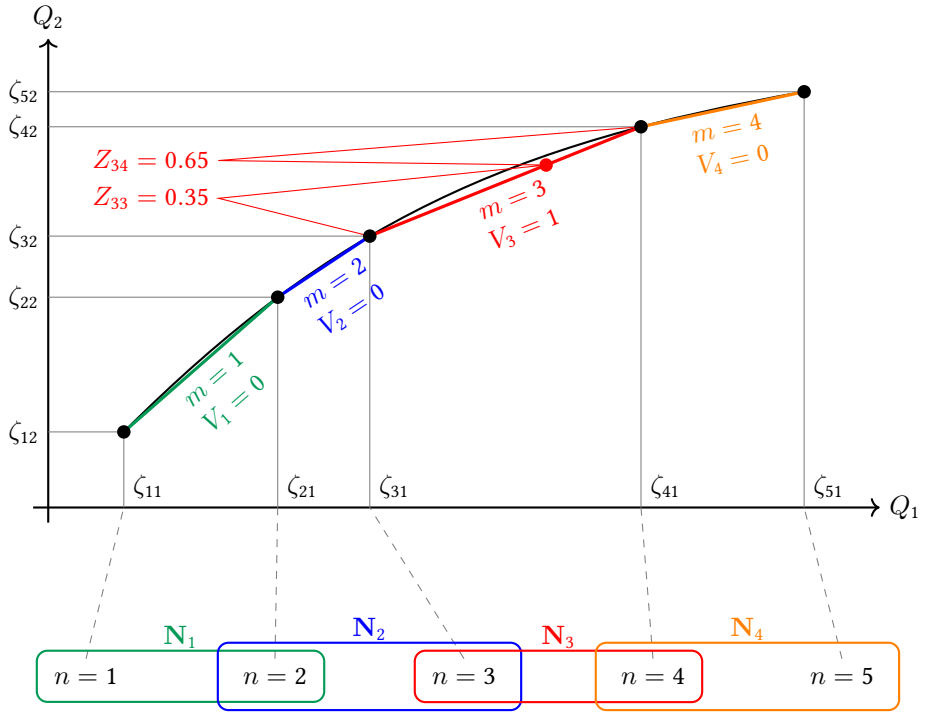


FIGURE 5.5: Example of one-dimensional piecewise-linear function. Sets are $\mathbf{N}_m$, coefficients are ζ_{mn}, and variables are Q_k, V_m, and Z_{mn}. The true nonlinear function is sampled at various points, with (piecewise) linear interpolation giving approximate values in between, as determined by the nonzero values of z_{mn}. Notice that the indicated point approximately satisfies the model.

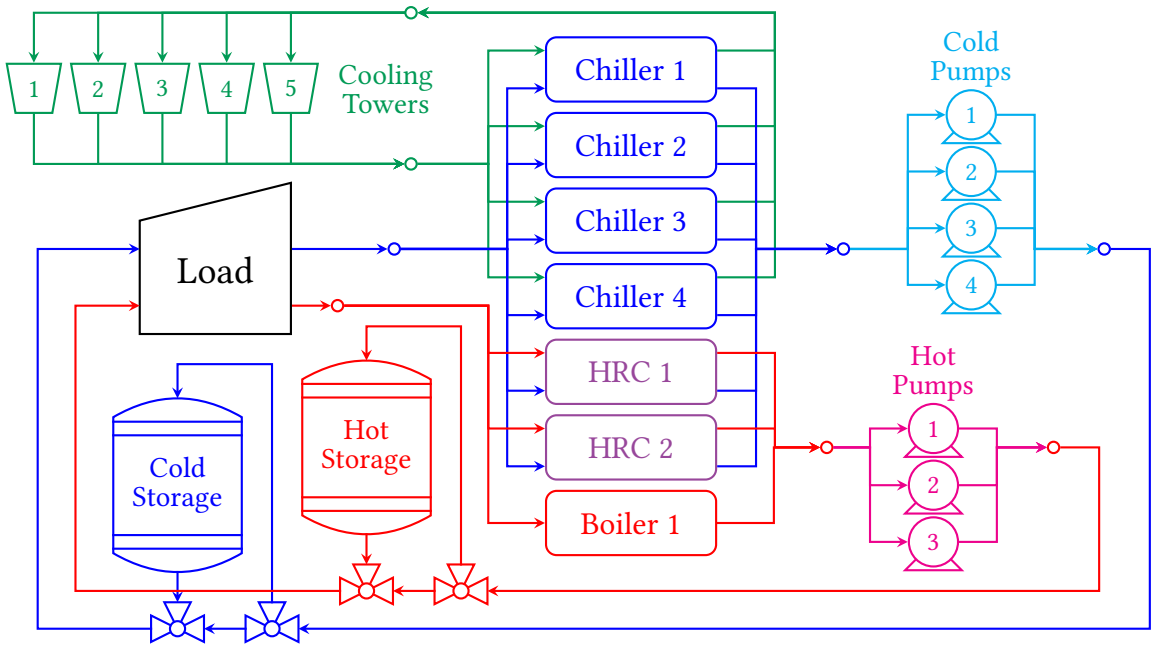


FIGURE 5.6: Diagram of combined heating and cooling central plant.

TABLE 5.2: Electricity consumption and cost breakdown by equipment type for heating and cooling example.

Equipment	Usage (MWh)		Cost (1000$)	
Chillers	331.4	(45.61%)	19.51	(42.98%)
HRCs	320.0	(44.04%)	21.11	(46.49%)
Cold Pumps	54.1	(7.45%)	3.42	(7.53%)
Hot Pumps	17.7	(2.43%)	1.16	(2.56%)
Cooling Towers	3.4	(0.46%)	0.19	(0.43%)
Total	726.5	(100.00%)	45.40	(100.00%)

horizon is 5 days (120 h). Utility prices and and heating/cooling demand are taken from real data provided by Johnson Controls. After 10 s, the optimizer finds a solution with a 0.98% optimality gap, while after 60 s, the gap is 0.75%. The optimization is stopped after 5 min with a final gap of 0.52%.

The ending solution for the combined heating and cooling problem is shown in Figure 5.7. Due to the large overlap between heating and cooling demand, the optimal solution is to run the HRCs constantly. The hot storage tank is then used to modulate consumption to meet demand. Chillers are run at higher levels during the evening hours when electricity is cheap. However, during peak hours when electricity is more expensive, the chillers are not completely shut down, as otherwise peak demand charges would be increased by the correspondingly higher chiller utilization at night. All heating and cooling demand is met on time. Costs are broken down in Table 5.2. From these values, we see that the most significant components of cost are the main chillers and HRCs. Auxiliary equipment constitutes roughly 10% of total cost, which is still significant enough to benefit from optimization.

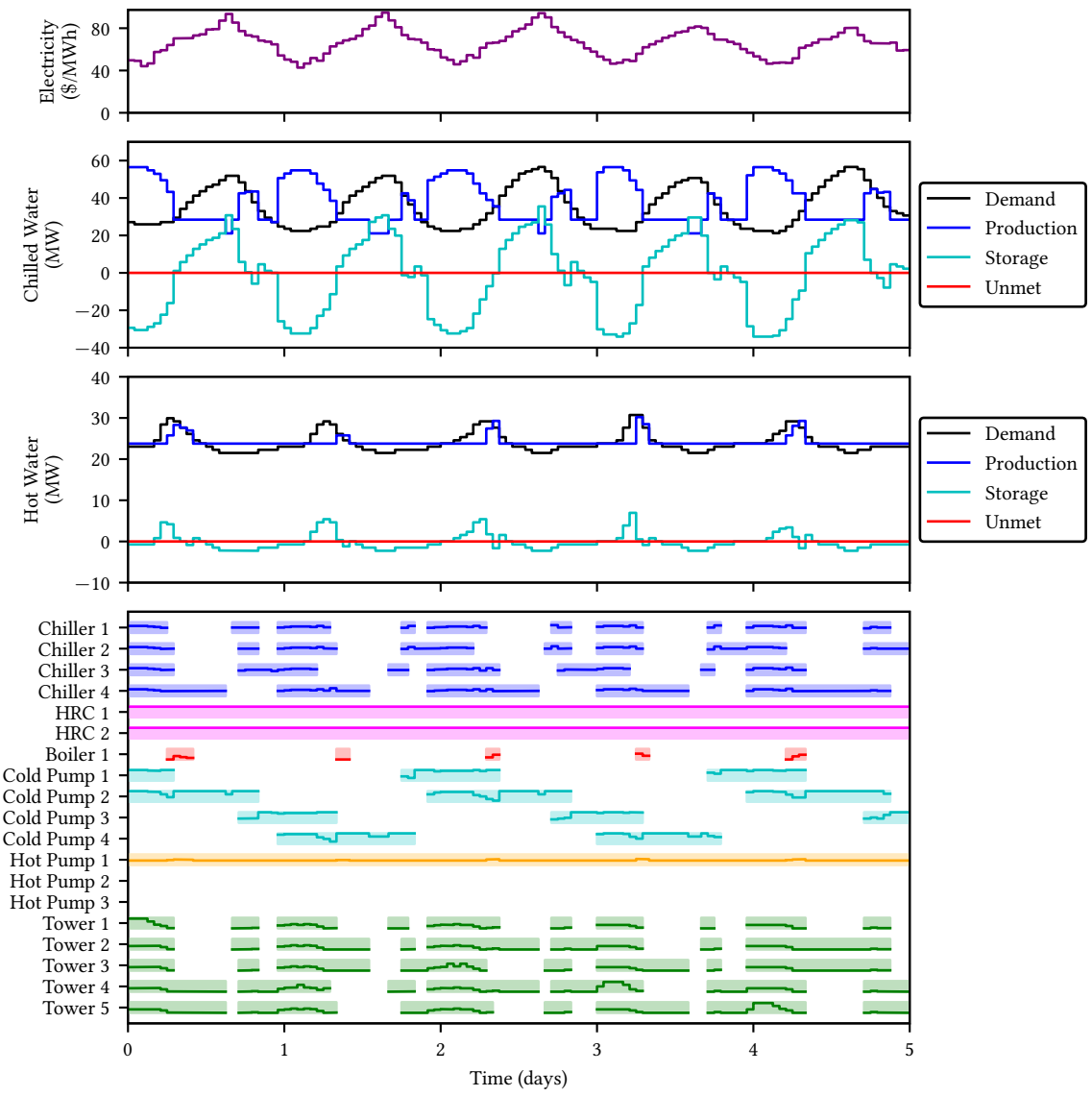


FIGURE 5.7: Optimal solution for combined heating and cooling example. Negative values of "Storage" indicate the tank is being charged, while positive values indicate discharge. In the Gantt chart, colored boxes indicate that a unit is active, while solid lines indicate part load between minimum and maximum capacity.

Table 5.3: Cost comparison between optimization and heuristic solutions.

Cost	Optimal	Heuristic	Change
Electricity Use Charges	45.40	45.95	+1.21%
Electricity Demand Charges	4.05	8.45	+108.93%
Natural Gas Charges	1.08	11.32	+948.62%
Total Charges	50.53	65.73	+30.08%

To compare the performance of the optimization model compared to existing techniques, we apply a heuristic strategy for the same problem. Because a complete heuristic is difficult to develop, we simulate a heursitic schedule by solving a modified optimization problem with the following additional constraints:

- All chillers and HRCs must run at full capacity between 1am and 7am each morning
- All chillers and HRCs must be completely off between 2pm and 6pm each afternoon

These rules are meant to represent a reasonable rule of thumb for central plant operation based on the prevailing electricity price dynamics. All other equipment decisions are made by the optimizer.

The solution obtained via the simulated heuristic is shown in Figure 5.8. Compared to the optimal solution, we see higher peak electricity usage and more use of the boiler to make up for reduced HRC utilization. Costs for the two solutions are shown in Table 5.3. From this table, we see that the heuristic solution incurs a significantly higher demand charge due to running chillers at full capacity. These rules lead to a 30% increase in total costs for the heuristic solution. Note that since most of the decisions are still made via optimization, this benchmark is an optimistic estimate of the actual performance were this heuristic implemented in practice; thus, although it seems reasonable, the chosen heuristic is actually quite poor due to the cost structure of the problem. In a real central plant at Stanford University, application of a similar (but less detailed) optimization strategy led to as much as 15% to 30% lower costs compared to manual operators (Blair, 2016; Wenzel et al., 2016). Thus, optimization strategies like the one proposed in this chapter have a significant potential to reduce utility costs and thus increase overall primary energy efficiency of the electricity grid.

5.4.2 Variable Supply Temperature

In this section, we wish to examine the effect of allowing chilled water temperature to vary. By lowering the chilled water supply temperature T_{CHWS}, the same total cooling duty Q_{CH} can be supplied using a lower volumetric flow V_{CHW}. Thus, electricity consumption in pumps is reduced at the cost of increased electricity consumption in chillers, and the optimal T_{CHWS} must balance these effects. However, as discussed in Section 5.3.1, considering variable T_{CHWS} in the MILP model requires adding an extra independent variable to the model (and thus numerous extra discrete and continuous decision variables to the problem). Thus, if there is a constant T_{CHWS} that is nearly optimal throughout the

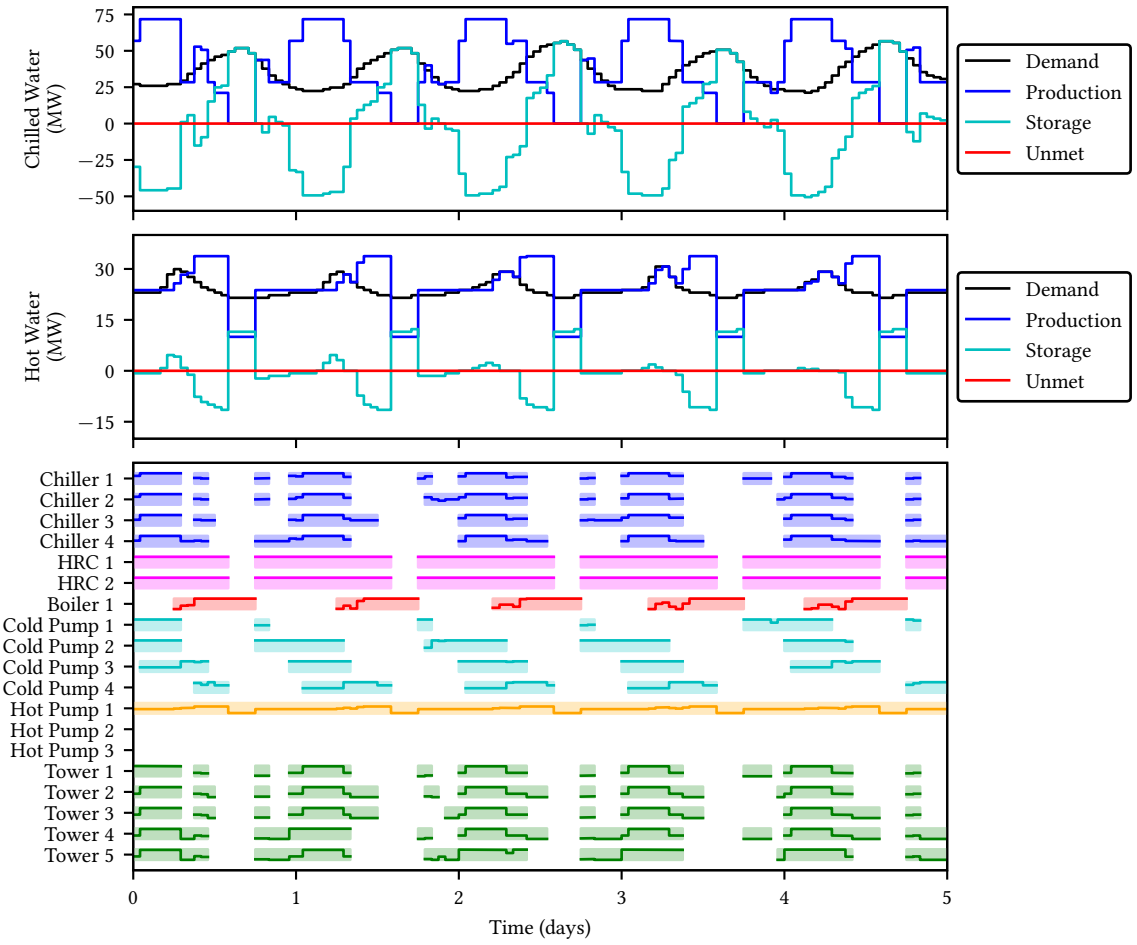


FIGURE 5.8: Simulated heuristic solution for the combined heating and cooling example. All plots are as in Figure 5.7.

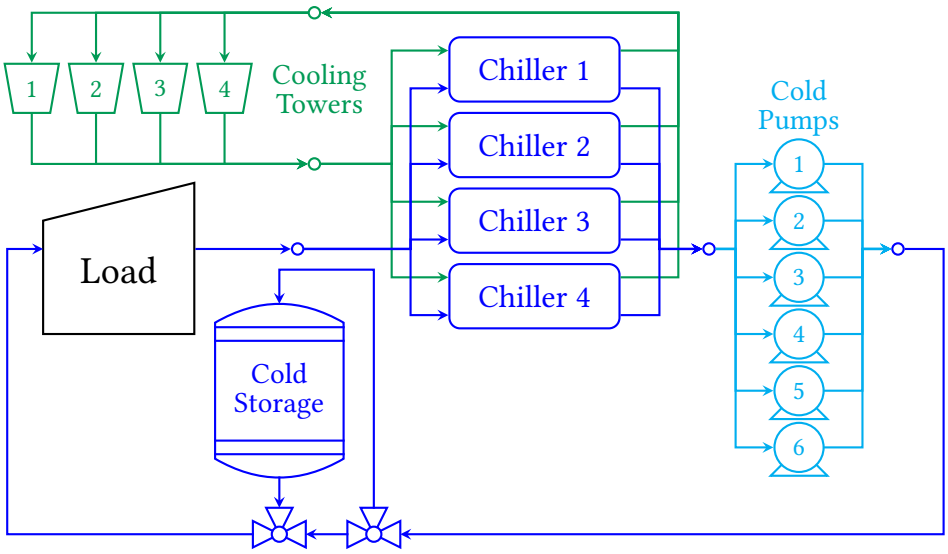


FIGURE 5.9: Diagram of cooling system for variable T_{CHWS} simulation.

entire operating range of the central plant, then the extra computational burden may not be worth minor electricity changes.

To answer these questions, we consider the cooling-only central plant shown in Figure 5.9. For various total cooling loads, we solve a one-period optimization problem to choose the optimal equipment configuration to meet that load. The only external resource input is electricity, which means the objective function is to minimize electricity use. Note that the presence of the storage tank is irrelevant for these subproblems. Due to the piecewise-linear approximation used for equipment models, material energy balances do not exactly close. Thus, to check whether the MILP model is sufficiently accurate, we also perform this comparison using the true nonlinear equipment models with explicit energy balances included.

For variable chilled water supply temperature, we allow T_{CHWS} to vary between 0 °C and 10 °C. This leads to the piecewise-linear model shown in Figure 5.10. Note that, as discussed in Section 5.3.1, the model is transformed from (Q, T) to (Q, V) space before being used in the MILP formulation. By contrast, the MINLP formulation uses T_{CHWS} directly as a decision variable. Using this and models for pumps and cooling towers, four sets of optimization problems are solved:

- MILP model with fixed $T_{\text{CHWS}} = 8$ °C
- MILP model with variable $0\ ^\circ\text{C} \leq T_{\text{CHWS}} \leq 10\ ^\circ\text{C}$
- MINLP model with fixed $T_{\text{CHWS}} = 8$ °C
- MINLP model with variable $0\ ^\circ\text{C} \leq T_{\text{CHWS}} \leq 10\ ^\circ\text{C}$

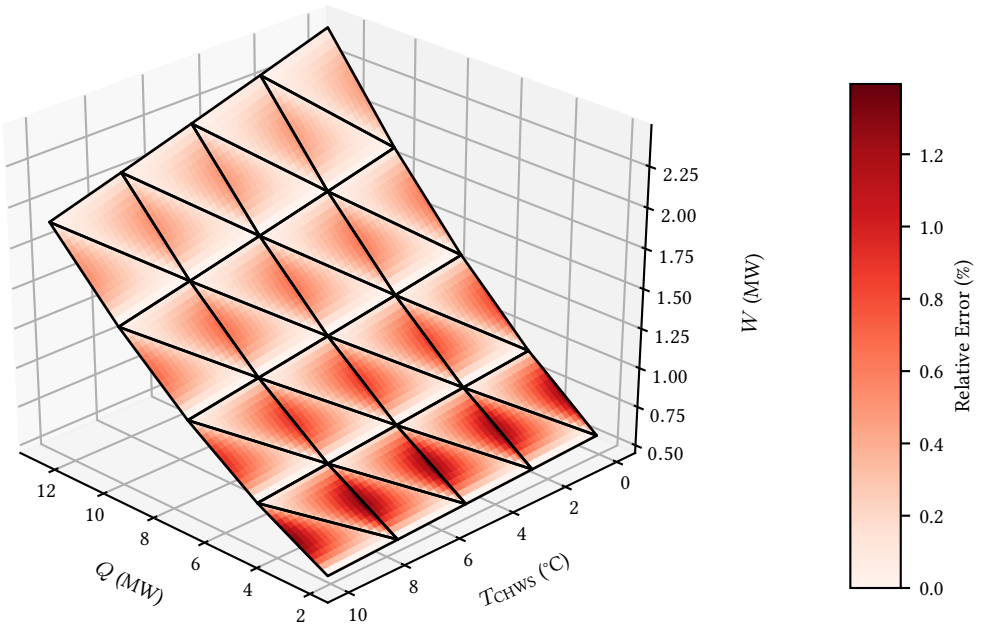


FIGURE 5.10: Piecewise-linear model for chillers with variable T_{CHWS} using five grid points along each axis. Maximum relative error is below 1.5% compared to the nonlinear model.

To provide a fair comparison, electricity consumption rates for the MILP solutions are recalculated by evaluating the nonlinear models at the chosen setpoints. Thus, the MILP models cannot unfairly benefit if they slightly underestimate electricity use.

REMARK 5.4: In general, the optimal cost of the constant T_{CHWS} optimization should be no better than that of the variable T_{CHWS} objective. Indeed, this property holds for the MINLP model, as the variable T_{CHWS} problem is a relaxation of the constant. However, because of how the piecewise-linear models are constructed in the MILP, the constant T_{CHWS} operating surface is not necessarily a subset of the variable operating surface. Thus, the constant T_{CHWS} MILP could possibly obtain a lower objective function, if it has access to slightly cheaper operating points. To minimize any such effects, we choose the constant T_{CHWS} so that it is one of the grid points in the variable T_{CHWS} model; that is, the operating surface for the constant T_{CHWS} model is exactly the $T_{\text{CHWS}} = 8$ °C slice of the surface shown in Figure 5.10. Therefore, the variable T_{CHWS} problem *is* a relaxation of the variable problem. However, because electricity usage is recalculated using the nonlinear models, it is still possible that the cost of the constant model could be less than the variable model due to the nonuniform approximation error of the piecewise-linear model.

Figure 5.11 shows electricity consumption as a function of total cooling load for these problems. From these results, there are two main takeaways. First, there is good agreement between the MILP and corresponding MINLP results. Above 25 MW total load, the two models are within 1% agreement (comparing variable MILP to variable MINLP and constant MILP to constant MILP). Below 25 MW, disagreement is as large as 2%, but this impact is minimized

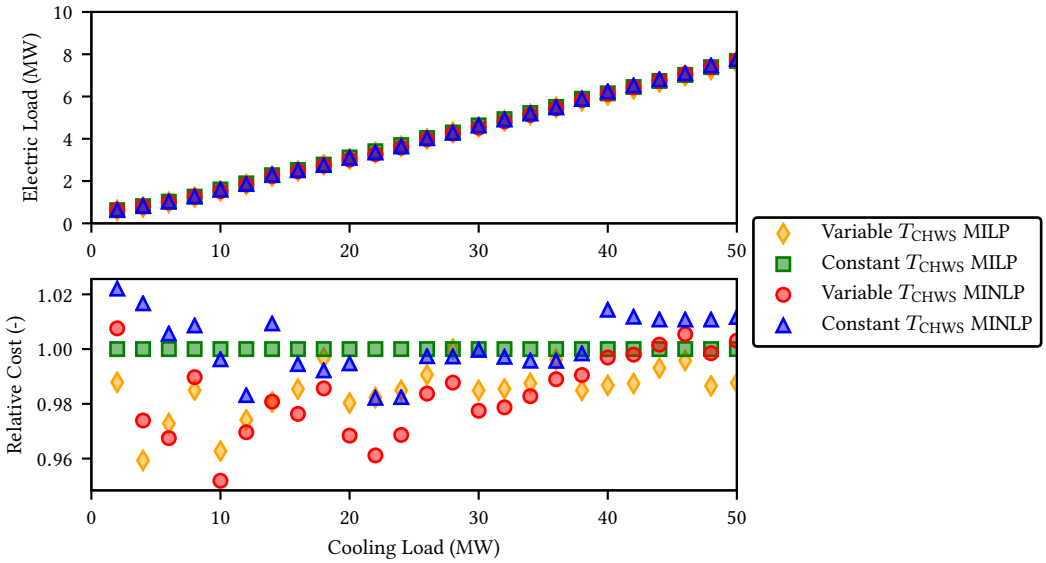


FIGURE 5.11: Optimal electricity use for varying total cooling load as predicted by the MILP and MINLP formulations for variable and constant T_{CHWS}.

because these operating points consume less electricity and thus constitute a smaller fraction of total cost than higher cooling loads. Thus, the piecewise-linear models used in the MILP formulation are sufficiently accurate. Second, for these equipment models, there does not seem to be a significant benefit to varying T_{CHWS}. Although savings are as high as 4% for low total cooling loads, the weighted average (using cooling load as weights) savings are 1.2% for the MILP model. Thus, if the computational burden is significantly higher for the variable T_{CHWS} model, then it may be worthwhile to spend those extra computational resources elsewhere (e.g., on using a longer prediction horizon).

To further analyze the effective accuracy of the piecewise-linear models, we compare the optimal values of T_{CHWS} as predicted by the MILP and MINLP models. These values, along with the number of active units, are shown in Figure 5.12. In general there is good agreement, but larger differences in optimal T_{CHWS} appear when there are jumps in the number of active pumps. Due to the slightly different equipment models used in the MILP and MINLP models, additional pumps are activated at slightly different total loads. Thus, the tradeoff between extra pumping (higher T_{CHWS}) and extra chilling (lower T_{CHWS}) are different at these points, leading to a larger difference in the optimal value of T_{CHWS}. There is a more significant difference in the number of active cooling towers predicted by each model, but these discrepancies are due to the cooling tower models being nearly linear throughout much of the operating region, leading to nearly degenerate solutions in the MILP model. By contrast, the two formulations agree exactly on the number of active chillers at each cooling load, which is the most important decision with respect to total cost. Thus, the two models are in very good agreement overall, indicating that the MILP model is sufficiently accurate.

Finally, to illustrate the significantly reduced computational burden of the MILP model versus the full MINLP model, we examine solution statistics for the subproblems. For both constant and variable T_{CHWS}, each MILP instance

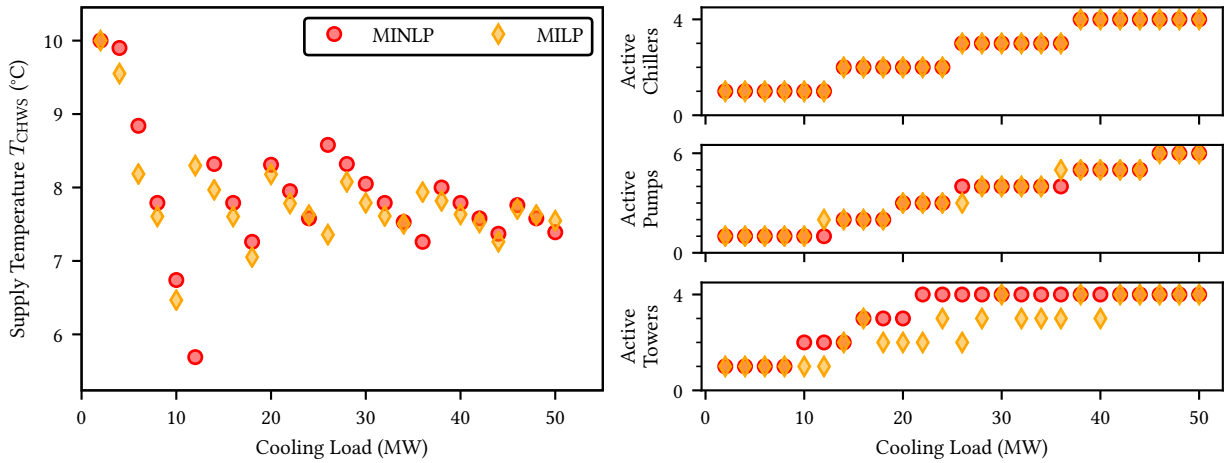


FIGURE 5.12: Optimal T_{CHWS} and number of active generators for variable-temperature models as a function of total cooling load.

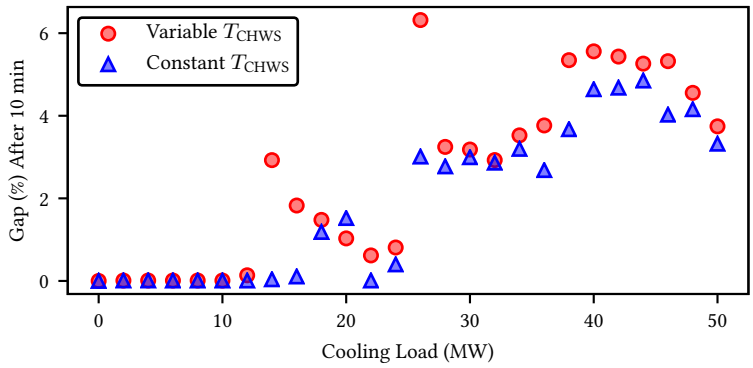


FIGURE 5.13: Optimality gaps after 10 min solution time for single-period MINLP equipment selection problems using BARON.

can be solved to optimality within 10 s using Gurobi (Gurobi Optimization, LLC, 2018). By contrast, for the full nonlinear models, each instance is a nonconvex MINLP for which it can be very difficult to prove optimality. Thus, we solve each subproblem using BARON (Sahinidis, 2018) with a time limit of 10 minutes. Figure 5.13 shows the final optimality gaps for each subproblem. We speculate that the true optimality gaps for these solutions are much smaller (i.e., the incumbent solution is near-optimal but the lower bound is not tight), but it is difficult to be sure. Indeed, these results suggest that it is difficult to sufficiently solve even a single-period MINLP model within the necessary solution time. Thus, the approximate MILP brings a significant speed advantage compared to the full MINLP model.

5.4.3 Symmetry Removal

To illustrate the effectiveness of the symmetry-free model, we solve large optimization problems two different strategies for identical pieces of equipment:

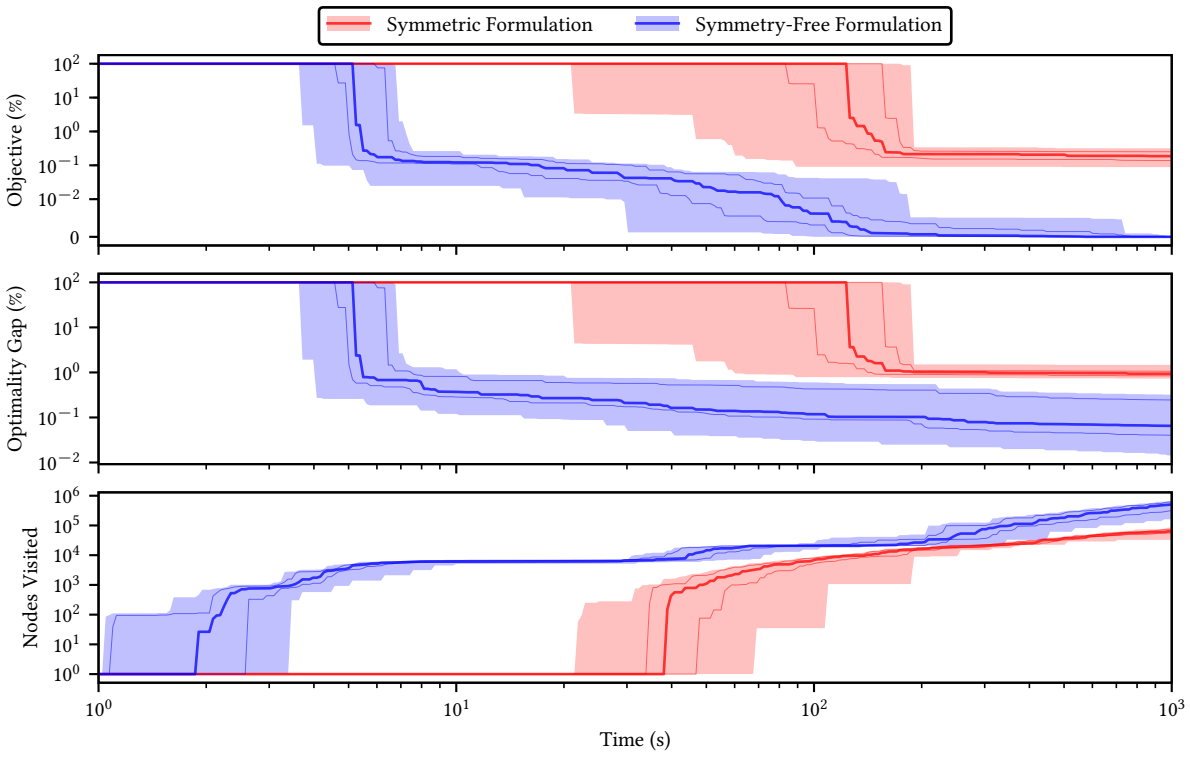


FIGURE 5.14: Average progress for solution, optimality gap, and nodes visited for the Symmetric and Symmetry-Free formulations. Solid lines show quartiles across 22 instances, while shaded regions show span.

- Duplicate the piecewise-linear models for identical equipment and make U_{jt} and V_{jmt} binary variables (the "Symmetric" formulation).
- Use one piecewise-linear model per type of equipment and make U_{jt} and V_{jmt} integer-valued as in 5.3.2 (the "Symmetry-Free" formulation).

We use the cooling-only system shown in Figure 5.9 except that there are eight identical chillers and ten identical cooling towers. The prediction horizon is 1 week (168 h). Using 22 different weeks of electricity price and cooling demand data, optimization problems are solved using the two different formulations with a time limit of 1000 s.

Figure 5.14 shows solution progress across these instances. Note that the bounds are normalized so that 0% corresponds to the best incumbent solution, and values are restricted to $[-100\%, 100\%]$. From this figure, we see that the symmetry-free formulation brings significant improvement to solution speed. In particular, the symmetry-free formulation finds a 1% suboptimal incumbent solution after 20 s; to find the same quality of solution, the symmetric formulation takes almost 400 s. This behavior is because the symmetry-free formulation can process nodes more quickly, and it also makes more progress per node on average. Thus, the symmetry-free formulation is strictly better than the symmetric formulation, and is should be used in all problem instances.

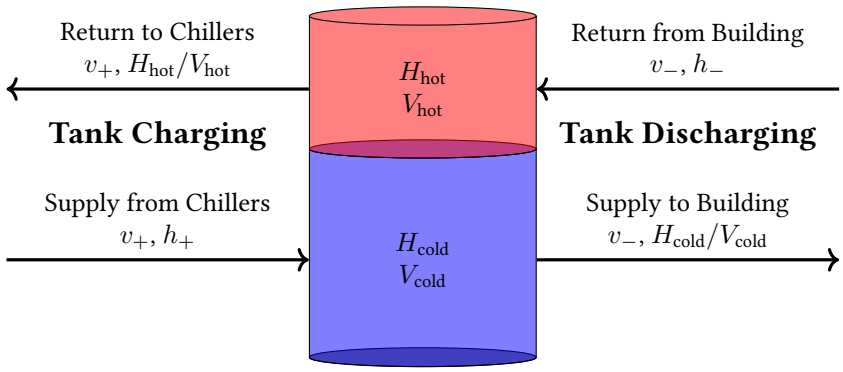


FIGURE 5.15: Diagram of stratified tank model.

5.4.4 TES Model Validation

As the final example in this section, we demonstrate the closed-loop performance of the model. As already illustrated, the piecewise-linear equipment models are sufficiently accurate for cost prediction. However, the simple linear storage tank model in (5.11) may not provide sufficient accuracy for long-term prediction. To answer this question, we simulate the storage tank using a more accurate (but nonlinear) stratified tank model similar to Ma et al. (2012b). As diagrammed in Figure 5.15, the tank is modeled as consisting to two hot and cold layers, each assumed to be uniform in temperature. The tank is well-insulated, but heat exchange (proportional to the temperature difference) takes place between the two layers. With state vector $(V_{\text{cold}}, V_{\text{hot}}, H_{\text{cold}}, H_{\text{hot}})$, the differential equations are as follows:

$$\begin{aligned}
\frac{dV_{\text{hot}}}{dt} &= -v_+ + v_-, \\
\frac{dV_{\text{cold}}}{dt} &= v_+ - v_-, \\
\frac{dH_{\text{hot}}}{dt} &= -\frac{H_{\text{hot}}}{V_{\text{hot}}} v_+ + h_- v_- - K\left(\frac{H_{\text{hot}}}{V_{\text{hot}}} - \frac{H_{\text{cold}}}{V_{\text{cold}}}\right), \\
\frac{dH_{\text{cold}}}{dt} &= h_+ v_+ - \frac{H_{\text{cold}}}{V_{\text{cold}}} v_- + K\left(\frac{H_{\text{hot}}}{V_{\text{hot}}} - \frac{H_{\text{cold}}}{V_{\text{cold}}}\right),
\end{aligned}$$

in which inputs v_+ and v_- are the charge and discharge volumetric flows, and parameters h_+ and h_- are the (per-volume) supply and return enthalpies, with heat transfer coefficient K between the two layers. In chilled water tanks, the state of interest is the enthalpy of the cold section H_{cold}, while for hot water tanks, it is H_{hot}.

As suggested in Risbeck et al. (2017), the nonlinear storage tank model can be linearized at a given operating point to determine a linear model. However, an alternative strategy that achieve greater global accuracy is to use system identification to choose the appropriate value of σ_k. Figure 5.16 shows data obtained from the nonlinear model, along with forward simulations of the obtained linear model. Even over the full 10-day horizon, the linear model fits very well, and thus is sufficient for optimization.

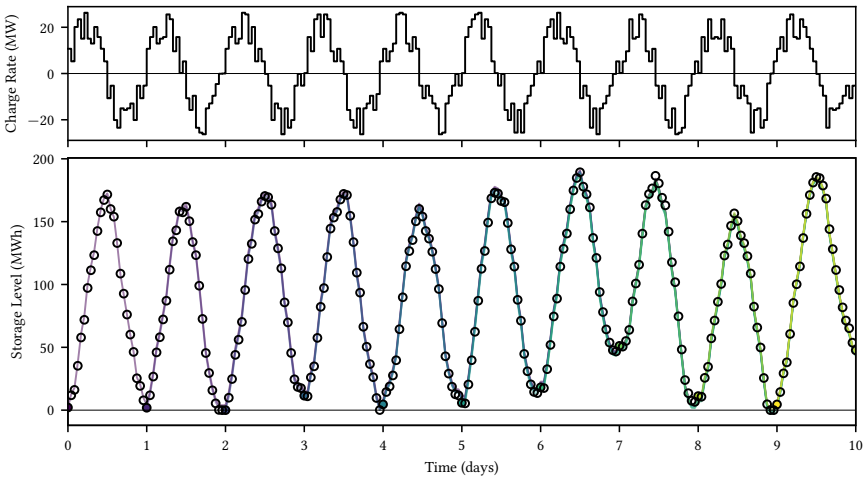


FIGURE 5.16: Linear fit of nonlinear stratified tank model. Open circles show data from the nonlinear model, while solid lines show forward simulations of the linear model starting from each day's initial condition.

To verify this accuracy in the optimization context, we simulate closed-loop optimization using the nonlinear tank model as the "true" storage tank. At each timestep, the optimizer is given the initial cold storage level H_{cold}. After solving the open-loop optimization, the first-period storage discharge rate is injected into the nonlinear model. Optimization is repeated at the next timestep using the updated value of H_{cold}, which could be different from the level predicted by the linear model. For simplicity, the only equipment in the central plant is two conventional chillers. To avoid nominal revisions to the open-loop solution, we remove the peak demand charge, and we use the optimal 13-day solution as a terminal constraint for the storage tank. This constraint prevents the optimizer from emptying the storage tank at the end of each closed-loop optimization. Note that this reference trajectory is generated using the linear storage tank model; thus, if the liner model is not accurate, then the terminal constraint will be poor, and the optimizer will need to make significant revisions to the open-loop solution. The simulation length is 10 days, and each optimization has a 3-day horizon.

The closed-loop production trajectory is shown in Figure 5.17. Because there is no peak demand charge, the optimal solution is to run the chillers near full capacity in the evening when electricity is less expensive. This strategy is observed, and there is no obvious transient misbehavior. The states of the nonlinear model are shown in Figure 5.18. While the temperature of the cold section does vary sharply when nearly empty, when full, the temperature is relatively constant. Thus, it is not necessary to model volume and enthalpy independently in the optimization model. To verify that the linear storage model is indeed sufficiently accurate, we plot the closed-loop storage tank trajectory along with the open-loop predictions in Figure 5.19. As observed in this plot, there is only minor open-loop/closed-loop discrepancy in the storage tank level. In addition, the storage level remains close to the reference trajectory throughout the simulation. Thus, feedback is sufficient to correct the minor inaccuracy of the storage tank model without significant open-loop/closed-loop mismatch.

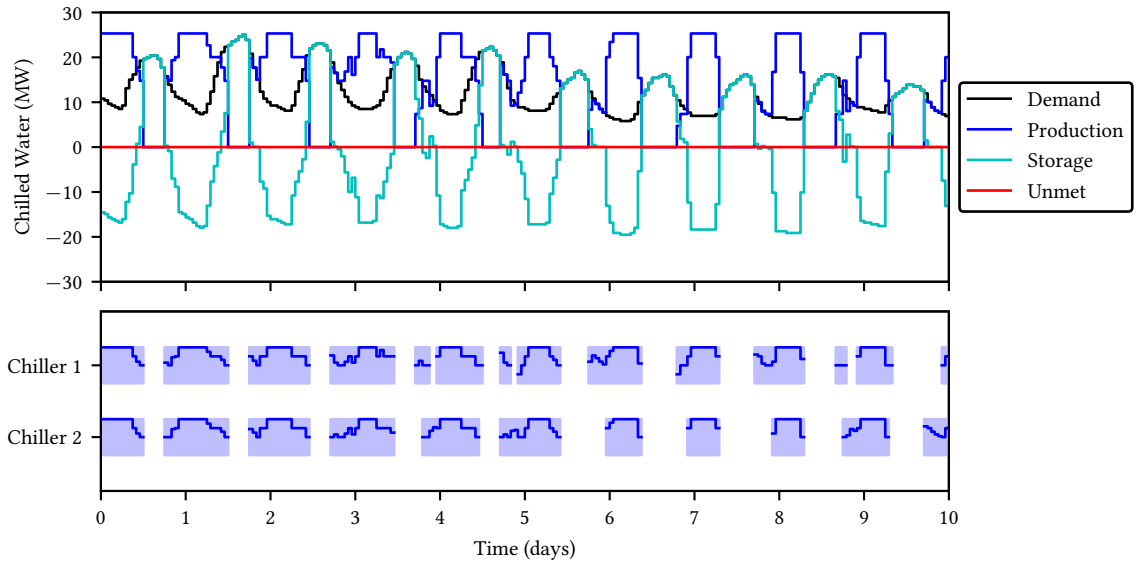


FIGURE 5.17: Closed-loop chilled water production and chiller utilization for a nonlinear storage tank.

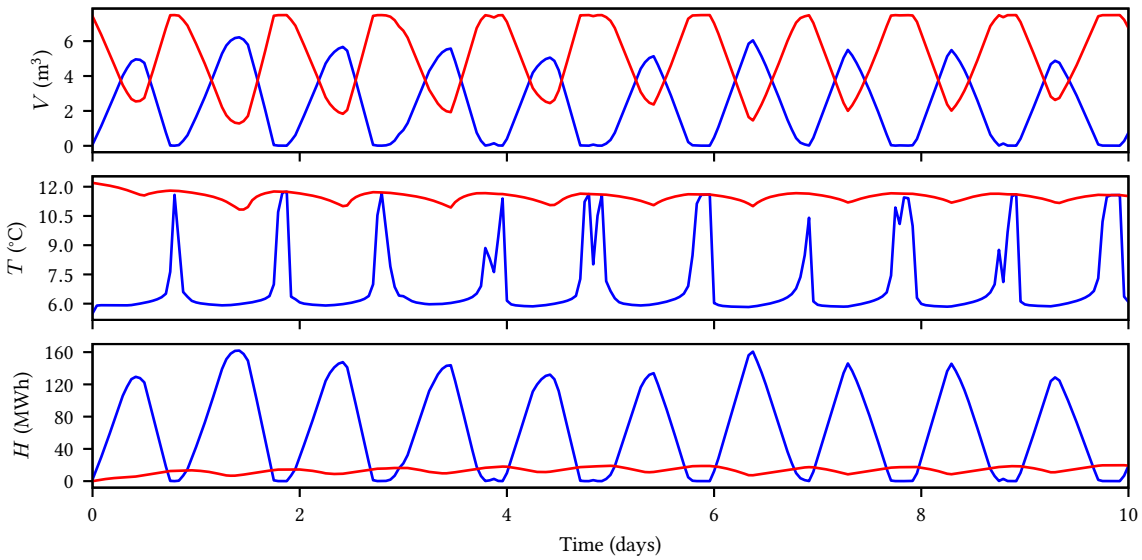


FIGURE 5.18: States of nonlinear tank model throughout the simulation. The cold section is in blue, and the hot section is in red.

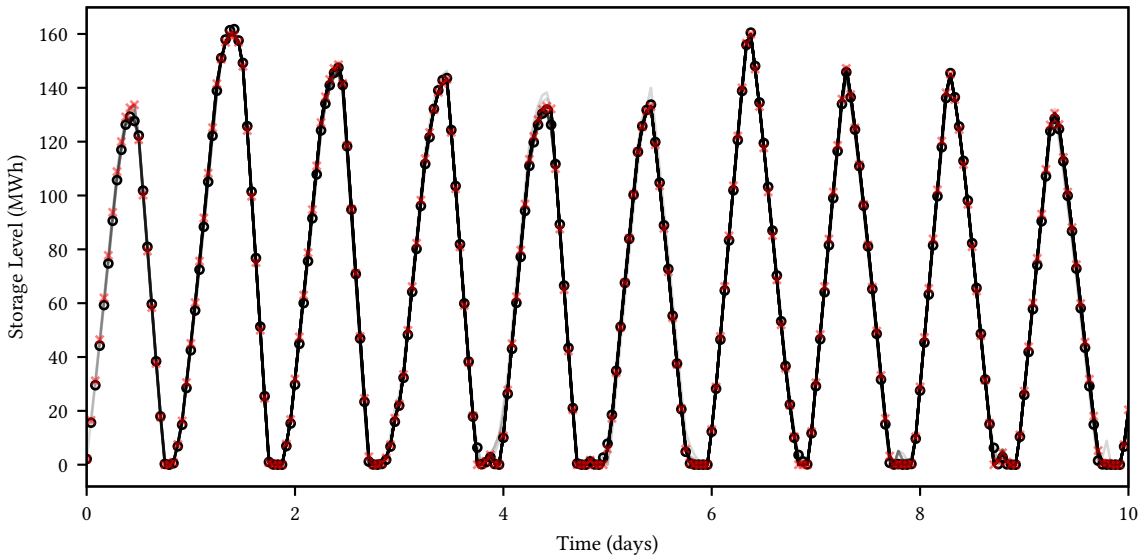


FIGURE 5.19: Predicted storage trajectories for the closed-loop simulation of a nonlinear storage tank. Open circles show true storage tank trajectory, while transparent lines show predicted trajectories for each open-loop solution. Reference trajectory used as a terminal constraint is shown in red x markers.

Identification of linear model parameters is formulated using CasADi (Andersson et al., 2018) via MPCTools (Risbeck and Rawlings, 2018a), with optimization using IPOPT (Wächter and Biegler, 2006). The nonlinear tank model (with minor modifications to avoid singularities at $V = 0$) is simulated in closed loop using integrators provided in CasADi.

5.5 SUMMARY

In this section, we have presented an MILP formulation for real-time cost optimization of central energy plants. This model considers each piece of equipment individually using piecewise-linear models that include nonzero minimum capacity, and it optimizes chilled and hot water production simultaneously. Using forecasts of time-varying utility prices and resource demands, the optimization problem determines equipment and storage utilization over a finite prediction horizon. With off-the-shelf MILP solvers, near-optimal solutions (within 1% optimality) can be found within 1 min for realistically sized problems. Via examples, we have demonstrated improved performance compared to a simulated heuristic, good agreement with an MINLP formulation using the true nonlinear equipment models, and high accuracy of predictions made by the linear storage tank model compared to a more detailed nonlinear simulation. Thus, the proposed optimization formulation meets all of the requirements for real-time optimization of central energy plants, and it has the potential to reduce costs and improve operational consistency in real facilities.

Additional computational examples of closed-loop performance, including the effect of horizon length, can be found in Risbeck et al. (2016). More details about symmetry removal and other formulation considerations can be found in Risbeck et al. (2018b).

5.6 Appendix: Piecewise-Linear Modeling

In this section, we consider the general process of making a piecewise-linear approximation of a given (vector-valued) function $f : \mathbb{X} \subset \mathbb{R}^n \to \mathbb{R}^m$. That is, we wish to determine a function $\tilde{f}(\,\cdot\,)$ such that $\tilde{f}(x) \approx f(x)$ for all $x \in \mathbb{X}$ with some suitable definition of $\approx$.

Remark 5.5: The concept of *approximating* $f(\,\cdot\,)$ is related to but conceptually different from *relaxing* $f(\,\cdot\,)$. When constructing a relaxation of a function, one seeks a set-valued map $\tilde{F}(\,\cdot\,)$ such that $f(x) \in \tilde{F}(x)$ and $|y - f(x)| \leq \epsilon$ for all $y \in \tilde{F}(x)$. For example, consider the well-known McCormick relaxation (or "envelope") for $f(x, y) = xy$, given by

$$\tilde{F}(x, y) = \{z \in \mathbb{R} : \max(0, x + y - 1) \leq z \leq \min(x, y)\}$$

on $(x, y) \in [0, 1]^2$. Such relaxations are used extensively in global mixed-integer nonlinear optimization (Belotti et al., 2013) and may be useful to preserve feasibility when nonlinear constraints are very tight. However, for the problem of interest in this chapter, approximations are sufficiently accurate and easier to solve than relaxations. Thus, we focus on approximations in this section.

5.6.1 Piecewise-Linear Functions

We begin with the following definition:

Definition 5.6 (piecewise-linear function): A function $f : \mathbb{X} \subseteq \mathbb{R}^n \to \mathbb{R}^m$ is a *piecewise linear map* if it can be expressed as

$$f(x) = f_{\rho(x)}(x), \quad \rho(x) \coloneqq \min\{p : x \in \mathbb{X}_p\}$$

for *finite* sequences $(f_p(\,\cdot\,))_{p=1}^P$ of affine functions and $(\mathbb{X}_p)_{p=1}^P$ of bounded polyhedral sets such that $\cup_{p=1}^P \mathbb{X}_p = \mathbb{X}$.

That is, the domain $\mathbb{X}$ can be partitioned into P polytopes $\mathbb{X}_p$ such that $f(\,\cdot\,)$ is given by the linear (more precisely, affine) expression $f_p(x)$ on each polytope. An example of such domain partitioning is shown in Figure 5.20. One desirable property of piecewise-linear functions is continuity. Because continuity in the interior of each subdomain is immediate, a piecewise-linear function is continuous if it is continuous on all overlapping regions, i.e., $f_p(x) = f_q(x)$ for all p and q such that $x \in \mathbb{X}_p \cap \mathbb{X}_q$.

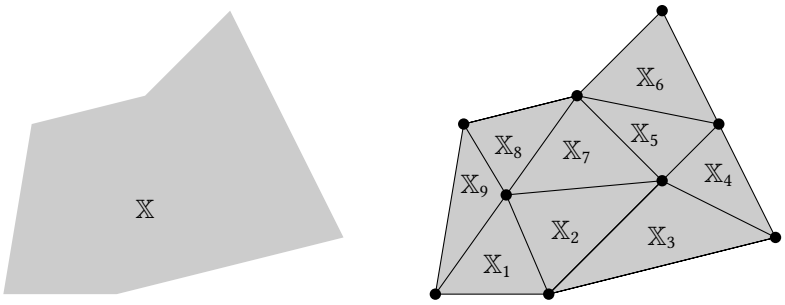


FIGURE 5.20: Example triangular partitioning of domain $\mathbb{X}$. Dots indicate extreme points of polyhedral subdomains $\mathbb{X}_p$.

There are two ways to represent the polyhedral subdomains $\mathbb{X}_i$: halfspace and extreme point. In the halfspace representation, one chooses matrices A_p and vectors b_p such that each $\mathbb{X}_p \coloneqq \{x \in \mathbb{R}^n : A_p x \leq b_p\}$. The linear functions are then given by $f_p(x) \coloneqq C_p x + d_p$ for suitably chosen matrices C_p and vectors d_p. This formulation is generally most convenient for evaluating a given piecewise-linear function, but it can be difficult to define the function this way. Alternatively, in the extreme point representation, each polytope $\mathbb{X}_p$ is associated with its set of vertices $X_p \coloneqq \{x_p^1, x_p^2, \ldots, x_p^{T_p}\}$. Note that a point is a vertex of polytope if it cannot be written as a nontrivial linear combination of other points in the polytope. (More precisely, a point v is a vertex of polytope $\mathbb{P}$ if $v \in \mathbb{P}$, and, for any two points $x, y \in \mathbb{P}$ and scalar $\lambda \in [0, 1]$ such that $v = \lambda x + (1 - \lambda)y$, it holds that $\lambda \in \{0, 1\}$, i.e., $v = x$ or $v = y$.) We then have $x \in \mathbb{X}_p$ if there exist a set of nonnegative coefficients $\lambda_p^1, \ldots, \lambda_p^{T_p}$ such that

$$x = \sum_{t=1}^{T_p} \lambda_p^t x_p^t \quad \text{and} \quad \sum_{t=1}^{T_p} \lambda_p^t = 1.$$

The individual functions are then given by

$$f_p(x) = \sum_{t=1}^{T_p} \lambda_p^t f(x_p^t),$$

i.e., as a convex combination of $f(\,\cdot\,)$ evaluated at the extreme points of $\mathbb{X}_p$. In general, this representation is more convenient to define, but less convenient to directly evaluate.

5.6.2 Approximation

With a suitable definition and representation of piecewise-linear functions, we now wish to choose a strategy for constructing approximations. For univariate scalar functions (i.e., $n = m = 1$), there exist algorithms to bound the number of segments needed to achieve a given accuracy and choose their locations (Frenzen et al., 2010; Phillips, 1968). When the domain is 2D but the function value is still scalar, Rippa (1992) presents a greedy algorithm to construct a secant approximation by iteratively adding the point of maximum error to the triangulation. For cases where the polyhedral partition has already been determined, choosing the optimal linear functions can be cast as a

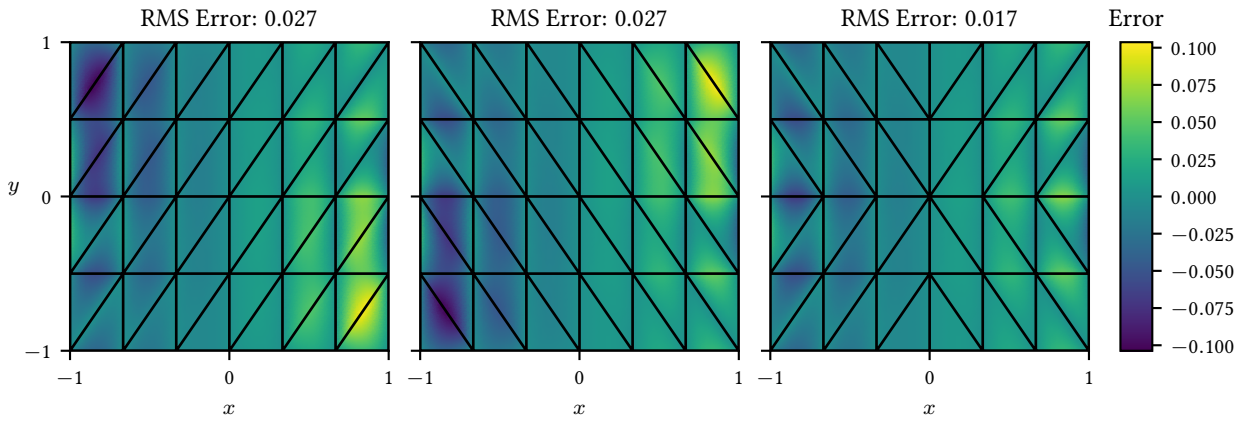
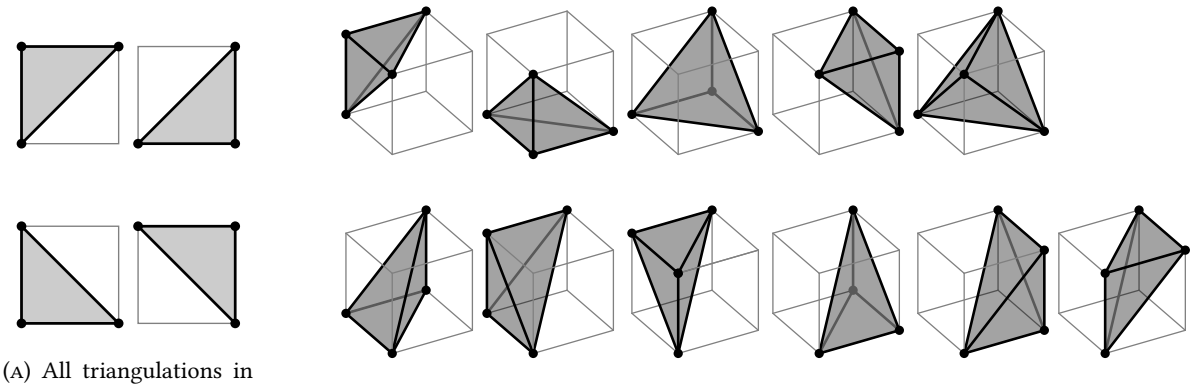


FIGURE 5.21: Example triangulations for function $f(x, y) = x/(1 + y^2)$. The third triangulation chooses between the two possible triangulations for each rectangular region, taking the smallest absolute error.

convex optimization problem (Toriello and Vielma, 2012). These methods can be extended to multiple dimensions in the range (i.e., $m > 1$) by applying them separately for each output dimension, but for additional dimensions in the domain (i.e, $n > 1$), generalization is difficult.

For the desired use case, we propose a simple three-step approach for constructing piecewise-linear approximations. First, the domain is split into a *rectangular* partitioning. Typically, we choose a uniform grid, but for certain functions, a nonuniform grid is advantageous. Note that, as illustrated in Figure 5.4, the actual variable space may not be rectangular, but we assume that the domain is rectangular under suitable transformation. Second, the nonlinear function is evaluated at each of the lattice points. The piecewise-linear approximation will take on these values at the lattice points. Third, each rectangular subdomain is triangulated using the lattice points as vertices. As illustrated in Figure 5.21, the triangulation could be the same for each rectangle, or the minimum-error triangulation could be chosen for each subdomain. In one dimension, this step is unnecessary, as each subdomain is already a one-dimensional simplex. Figure 5.22 shows triangulations for two and three dimensions. As illustrated in Figure 5.22b, different triangulations in three or more dimensions may require a different number of subdomains in each rectangular region. In any case, these triangles become the polyhedral subdomains for the piecewise-linear approximation, and the affine function used on each subdomain is uniquely defined by the function values from the previous step.

Although the proposed approximation method does not guarantee optimal piecewise-linear approximation, it does have some desirable properties. By choosing each subdomain $\mathbb{X}_p$ as a simplex (i.e., the convex hull of $n + 1$ extreme vertices in $\mathbb{R}^n$) and using common vertices for adjacent regions, the approximation is always continuous. In addition, due to the structure of the MILP model, optimal solutions are more likely to lie on subdomain vertices; thus, by using a secant approximation, the effective accuracy (i.e., as evaluated at optimal operating points) is likely to be higher compared to tangent or other approximations. The proposed technique is also agnostic to the dimensions of



(A) All triangulations in two dimensions.

(B) Some triangulations in three dimensions.

FIGURE 5.22: Example triangulations for rectangular regions. Each row shows a different triangulation.

the domain and range (although the triangulation step is more difficult in higher dimensions), which means it can be applied to all possible equipment models. Finally, by using the same domain partitioning for each component of $f(\cdot)$, only a single interpolation grid needs to be modeled in the optimization problem.

5.6.3 MILP FORMULATION

In order to use the piecewise-linear functions in an MILP optimization, it is necessary to add constraints and auxiliary variables to the formulation. Defining

$$\mathcal{F} \coloneqq \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m : y = f(x)\},$$

in which $f(\cdot)$ is the piecewise-linear function, we wish to model the set $\mathcal{F}$ using only linear equality and inequality constraints. As discussed in Vielma et al. (2010), there are a number of ways to formulate this set depending on dimensionality, configuration of the subdomains, etc. For brevity, we present two formulations based on the halfspace and extreme-point representations respectively of a piecewise-linear function.

In the halfspace formulation, we use the fact that each $\mathbb{X}_p$ is represented via the linear inequalities $A_p x \leq b_p$ with affine functions $f_p(x) = C_p x + d_p$. In Vielma et al. (2010), this formulation is called the "multiple choice" model. For each polyhedral subdomain, we add auxiliary variables $x_p \in \mathbb{R}^n$ and $v_p \in \{0, 1\}$. The constraints are as follows:

$$\sum_{p=1}^{P} v_p = 1, \tag{5.17a}$$

$$A_p x_p \leq b_p v_p, \quad p \in \{1, \ldots, P\}, \tag{5.17b}$$

$$x = \sum_{p=1}^{P} x_p, \tag{5.17c}$$

$$y = \sum_{p=1}^{P} \left(C_p x_p + d_p v_p \right). \tag{5.17d}$$

In (5.17a), the optimizer chooses which subdomain to use by setting exactly one of the v_p to one, with all others zero. The second constraint (5.17b) then enforces $A_p x_p \leq b_p$ for the active subdomain, while forcing $A_p x_p \leq 0$ for all other inactive regions. Note that it is important that each $\mathbb{X}_p$ is bounded so that $A_p x_p \leq 0 \implies x_p = 0$. Finally, (5.17c) and (5.17d) calculate the desired x and y variables by summing the individual subdomain values; in both cases, only one term in the summation will be nonzero. Satisfaction of (5.17) thus ensures that $(x, y) \in \mathcal{F}$. Assuming the triangulation procedure from the previous subsection was followed, each A_p will have exactly $n + 1$ rows. Therefore, the halfspace formulation requires nP continuous auxiliary variables, P discrete auxiliary variables, and $(n + 1)P + n + m + 1$ constraints.

For the extreme-point formulation, x and y are still split into constituent parts for each subdomain, but this time, they are calculated as convex combinations of the extreme points. As in the halfspace representation, discrete auxiliary variables $v_p \in \{0, 1\}$ are added. Recalling that each subdomain $\mathbb{X}_p$ has extreme points $\{x_p^1, \ldots, x_p^{T_p}\}$, we also add auxiliary variables $\lambda_p^t \in [0, 1]$. Note that although some extreme points may be present in multiple subdomains (i.e., $x_{p_1}^{t_1} = x_{p_2}^{t_2}$ for some $p_1 \neq p_2$), there is a separate copy λ_p^t for each point *and* subdomain. Note that there are formulations that avoid these copies, they are not tight and generally lead to worse performance (Vielma et al., 2010). With these variables, the set $\mathcal{F}$ is modeled as

$$\sum_{p=1}^{P} v_p = 1, \tag{5.18a}$$

$$\sum_{t=1}^{T_p} \lambda_p^t = v_p, \quad p \in \{1, \ldots, P\}, \tag{5.18b}$$

$$x = \sum_{p=1}^{P} \sum_{t=1}^{T_p} \lambda_p^t x_p^t, \tag{5.18c}$$

$$y = \sum_{p=1}^{P} \sum_{t=1}^{T_p} \lambda_p^t f(x_p^t). \tag{5.18d}$$

As before, (5.18a) chooses which subdomain to activate via $v_p = 1$. However, instead of considering x_p directly as a decision variable in Cartesian coordinates, each x_p is calculated in Barycentric coordinates, using the weights λ_p^t as decision variables to determine x_p as a convex combination of extreme points x_p^t. Equation (5.18b) enforces valid weights (i.e., summing to 1 for the active subdomain and all zero for inactive subdomains), while (5.18c) and (5.18d) calculate x and y by summing the individual values. Satisfaction of (5.18) thus also gives $(x, y) \in \mathcal{F}$. In Vielma et al.

(2010), this formulation is called the "disaggregated convex combination" model. Following the proposed triangulation procedure, each $T_p = n + 1$, and thus this formulation requires P discrete auxiliary variables, $(n + 1)P$ continuous auxiliary variables, and $P + n + m + 1$ constraints with $(n + 1)P$ lower bounds $\lambda_p^t \geq 0$.

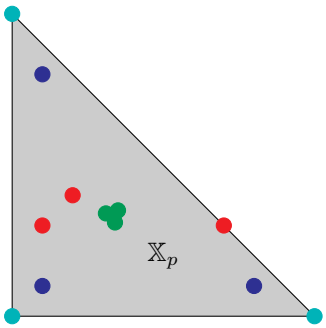
One potential advantage of the extreme point formulation over the halfspace formulation is that it can be made agnostic to which variables are independent and dependent. Specifically, we use the variable $z \coloneqq (x, y)$ directly and replace (5.18c) and (5.18d) with the single

$$z = \sum_{p=1}^{P} \sum_{t=1}^{T_p} \lambda_p^t z_p^t,$$

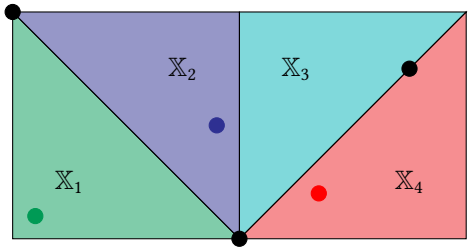
with $z_p^t \coloneqq (x_p^t, f(x_p^t))$. This distinction (or lack thereof) has no impact on solution times, but it potentially allows the algebraic representation of the model to be more compact. In particular for the central energy plant model, the generality is nice so that different generators can use different resources as the independent and dependent variables. By contrast, the halfspace formulation must specifically differentiate between x and y or else convert the $\mathbb{X}_p$ to some dummy space and add the appropriate extra rows to C_p and d_p. Besides this flexibility, there is little difference between the two formulations. Computational simulations from (Vielma et al., 2010) suggest that the halfspace formulation performs better for a specific transportation problem using CPLEX 11. However, performance is likely to vary for different problem types, especially as MILP solvers continue to improve. Note that both formulations require $O(P)$ auxiliary binary variables and constraints. In Vielma and Nemhauser (2009), alternative formulations are presented that require only $O(\log P)$ auxiliary binary variables and in specific cases only $O(\log P)$ constraints. While these formulations can be effective for very large P, simulations of the central plant optimization suggest that benefits are small at best for modest values of P (on the order of 2 to 20). As with the difference between the halfspace and extreme point representations, performance for these methods ie likely to vary between problem and solve, and thus, evaluation must be on a case-by-case basis.

5.6.4 Multiple Points

In certain cases, it may be desirable to choose multiple points from the same piecewise-linear function. For example, when a central plant contains multiple identical generators, the optimizer can choose to activate and operate the two generators independently, and thus the optimizer can pick between zero and two points from the same piecewise-linear set $\mathcal{F}$. If the individual x_i and y_i variables are required by the formulation (e.g., for rate-of-change constraints or for further use in piecewise-linear functions), then the auxiliary variables and constraints must be duplicated for each separate value of x_i and y_i. However, in problems where only the *sums* $\sum_i x_i$ and $\sum_i y_i$ are needed, there is a trivial modification to the halfspace and extreme point representations that allows this extension without adding additional variables or constraints.



(A) All four colored groups of operating points satisfy $v = 3$ and $\lambda_p^t = 1$ for $t \in \{1, 2, 3\}$.



(B) Points on shared boundaries (black) could be assigned to either region, which could affect value chosen for other points in that region (colored).

FIGURE 5.23: Nonuniqueness for mapping from aggregate x and y to individual x_i and y_i for multiple-point piecewise-linear formulations.

For the multiple-point extension, let $u \in \{0, \ldots, U\}$ be the integer-valued decision variable that chooses how many points $x_i = f(y_i)$ to select. Equations (5.17a) and (5.18a) are modified to

$$\sum_{p=1}^{P} v_p = u,$$

with the $v_p \in \{0, \ldots, U\}$ now integer-valued. The remaining constraints for each formulation are unchanged, except that now the variables x and y refer to the summations of the u values x_i and $f(x_i)$. After solving the optimization problem, it may be desirable to determine the individual values of x_i and y_i for later analysis. Unfortunately, when multiple points are selected from the same or adjacent regions, this assignment is nonunique as illustrated in Figure 5.23. Thus, different strategies need to be developed for this mapping depending on the particular application. Nevertheless, it is always possible to find a feasible combination of individual operating points, and thus the significant computational benefits of this formulation come at very little cost.

> Use the active [TES]. [It] is usually more direct and vigorous than the passive. … This rule does not, of course, mean that the [optimizer] should entirely discard the passive [TES], which is frequently convenient and sometimes necessary.
>
> — William Strunk, Jr.
> *The Elements of Style*

Chapter 6

Large-Scale HVAC Optimization

6.1 Introduction

In the previous chapter, a math programming model has been developed for the real-time cost optimization of central energy plants in campuses or large buildings. The main pieces of equipment in these plants are primarily for supplying chilled and hot water that is consumed in building air handling units (AHUs) to modulate temperature. In the previous problem formulation, it was assumed that chilled and hot water demand was a fixed parameter, and the central plant must simply provide this demand. A primary source of cost savings was thus the ability to store energy (in the form of hot or chilled water tanks), which allows temporal utility price and equipment efficiency differences to be exploited for cost savings. Assuming some fixed operation of the building HVAC system (e.g., a constant temperature setpoint), heating and cooling needs are fully specified by ambient conditions and there is nothing left to decide.

However, because buildings are bound by the laws of physics, temperature changes do not happen instantaneously. Depending on component masses, heat capacities, etc., certain parts of the building heat and cool much more slowly than others. In particular, solid parts of the building like floors and walls change temperature on different timescales than the air within the building, both due to physical properties (brick is more dense than air) and the nature of the energy transfer (air is cooled convectively by forcing colder air into rooms, while mass is cooled conductively by being contact with the air). These dynamic differences give an additional source of energy storage: by cooling solid building components, future ambient heat can be absorbed directly by the solid materials, thus providing a passive source of cooling to the building. For example, pre-cooling a 3-inch concrete floor by 6 °C can provide a quarter of cooling requirements for a typical 12 h occupation cycle (Morris et al., 1994). To differentiate between the two

types of storage, we refer to direct thermal energy storage (TES) in fluids as "active TES," with the use of building components as "passive TES." Utilizing both active and passive TES can provide cost savings relative to applying either technique individually (Henze et al., 2004; Pavlak et al., 2015).

6.1.1 Problem Statement

In order to exploit additional passive storage possibilities, we augment the problem statement from Chapter 5 to also consider optimization of all aspects of the heating, ventilation, and air conditioning (HVAC) equipment is buildings. This includes the central plant as discussed before, and in addition the AHUs. Based on the primary heat transfer medium, we refer to the central plant and its equipment as the "waterside" subsystem, while the building temperature zones and AHUs are the "airside" subsystem. In particular, the airside optimization must make the following decisions:

- What is the predicted temperature trajectory of each temperature zone?
- How much hot or chilled water is sent to AHUs?

The former implicitly decides passive storage utilization, while the latter determine how much hot or chilled water must be produced in the central plant. As in the waterside problem, the following constraints must be respected:

- Comfortable conditions must be maintained in occupied temperature zones.
- AHU capacities cannot be violated.

We would like to address large-scale instances, e.g., a tall high-rise building or a campus of multiple individual buildings. Thus, the problem formulation must be able to address hundreds of zones at a time. Note that all of the decisions and constraints from the waterside problem still apply and must be considered as well.

6.1.2 Literature Review

Because of its ability to utilize forecasts of ambient and economic conditions, MPC has been proposed for control of HVAC systems (Mendoza-Serrano and Chmielewski, 2012; Oldewurtel et al., 2012). Indeed, MPC has demonstrated benefit over conventional PID or other control strategies (Bengea et al., 2014), although obtaining sufficiently accurate models can be a challenge (Killian and Kozek, 2016; Sturzenegger et al., 2016). Common methods to obtain dynamic models include first-principles derivations (Kircher and Max Zhang, 2015; Mendoza-Serrano and Chmielewski, 2012) or applying system identification techniques to operating data (Avci et al., 2013). Although nonlinearities can arise, e.g., from products of mass flows and temperatures (Ma et al., 2012b, 2015), linear models like (2.3) are often sufficient for control purposes (Ma et al., 2012a).

Due to the inherently unsteady nature of storage utilization, economic MPC is a useful tool for optimizing airside energy use (Ma et al., 2014). These methods use temperature dynamic models to predict optimal building temperature

trajectories, which implicitly accounts for passive TES. It is important that these strategies ensure occupant comfort, which can generally be based on temperature and possibly humidity (Fang et al., 1998; Frontczak and Wargocki, 2011), although more complicated metrics can be used if directly optimizing or constraining comfort (Freire et al., 2008). Most existing buildings include PI or on/off controllers to operate AHUs (Afram and Janabi-Sharifi, 2014). If retrofit is not possible, the dynamic limitations imposed by any local actuation or control layer must be included in the dynamic models (Rawlings et al., 2017a). Systems with more exotic heating and cooling equipment (e.g., radient slabs (Feng et al., 2015) or phase-change materials (Touretzky and Baldea, 2016)) provide additional opportunities for energy storage, but they also bring additional control challenges for which MPC is appropriate.

To address large-scale instances, distributed MPC strategies have been proposed, which split the control problem into multiple interacting subproblems (Cai et al., 2016; Morosan et al., 2010; Scherer et al., 2014). Stochastic programming approaches have been applied to address uncertainty in ambient and occupancy conditions (Ma et al., 2015; Zavala et al., 2009). However, most existing approaches do not consider the significant complexities of the central energy plant. Thus, to provide the greatest opportunity for large-scale savings, we wish to propose a formulation that can account for both passive TES use on the airside and active TES use on the waterside, all while optimizing equipment utilization in the central plant.

6.2 Problem Formulation

To consider optimization of the airside and waterside subsystems simultaneously, we augment the central plant model presented in Section 5.2 with additional variables and constraints as follows.

6.2.1 Sets

The airside model requires a single additional set:

- Airside temperature zones $i \in \mathbf{I}$

Note that a temperature zone can be used to describe a single room, multiple rooms served by a single AHU, or even an entire building (albeit at reduced accuracy). The only restriction is that the temperature of a single zone is assumed to be uniform.

6.2.2 Parameters

The main parameters for the airside problem formulation define the dynamic evolution of temperature zones. These parameters are as follows

- Temperature interaction coefficients $\alpha_{ii'}$

- Resource use coefficients $\beta_{ii'k}$
- Temperature disturbance forecast θ_{it}

These coefficients define a linear state-space model in the form (2.3). Note however, that the core dynamic model $\alpha_{ii'}$, $\beta_{ii'k}$ is time-invariant; this restriction is not a limitation of the formulation, but rather a reflection of the fact that determining a quality time-varying model for a real building is extremely difficult.

One of the requirements of the airside optimization is that the temperatures of occupied zones must be within established limits. To ensure that a feasible solution exists for each model, we soften these constraints with linear penalties. In addition, AHU capacities limit the amount of chilled or hot water that can be delivered to a zone. These considerations lead to the following additional parameters:

- Zone temperature bounds Θ_{it}^{+}, Θ_{it}^{-}
- Temperature voilation penalties ξ_{it}^{+}, ξ_{it}^{-}
- Resource consumption bounds Γ_{ikt}

In general, ξ_{it}^{+} and ξ_{it}^{-} are chosen sufficiently large so that temperature bounds are respected whenever feasible.

6.2.3 Variables

For airside optimization, we add the following variables:

- Zone temperatures $T_{it} \in (-\infty, \infty)$
- Resource consumption $G_{ikt} \in [0, \Gamma_{ikt}]$ by zones
- Slack variables $T_{it}^{+}, T_{it}^{-} \in [0, \infty)$ for temperature bounds

Note of course that only a small subset of resources (chilled and hot water) are consumed in the airside; thus, most of the G_{ikt} variables are removed from the model by setting the bound $\Gamma_{ikt} = 0$.

6.2.4 Constraints

The main additional constraint added to the formulation is the dynamic model for temperature evolution:

$$T_{it} = \sum_{i' \in \mathbf{I}} \alpha_{ii'} T_{i'(t-1)} + \sum_{k \in \mathbf{K}} \sum_{i' \in \mathbf{I}} G_{i'kt} + \theta_{it}, \quad i \in \mathbf{I}, t \in \mathbf{T}. \tag{6.1}$$

Note that as in the case of S_{k0} and U_{k0} from the waterside model, the quantities T_{i0} represent initial conditions to the model and are thus parameters in the optimization problem.

To enforce soft constraints on temperature bounds, we add the following two constraints:

$$T_{it} \geq \Theta^{-}_{it} - T^{-}_{it}, \quad i \in \mathbf{I}, t \in \mathbf{T} \tag{6.2}$$
$$T_{it} \leq \Theta^{+}_{it} + T^{+}_{it}, \quad i \in \mathbf{I}, t \in \mathbf{T}. \tag{6.3}$$

When temperature bounds $\Theta^{+}_{it}, \Theta^{-}_{it}$ are satisfied, the optimizer sets the slack variables T^{+}_{it}, T^{-}_{it} to zero; when bounds are violated, the slacks are set to nonzero values to ensure feasibility of (6.2) or (6.3).

In addition to adding the previous constraints to the problem formulation, the main demand satisfaction constraint (5.4) is modified to include a new consumption term G_{ikt} as follows:

$$\sum_{j \in \mathbf{J}} Q_{jkt} + Y_{kt} + P_{kt} + H_{kt} \geq \phi_{kt} + \sum_{i \in \mathbf{I}} G_{ikt}, \quad k \in \mathbf{K}, t \in \mathbf{T}. \tag{6.4}$$

Thus, the central energy facility must meet the demands of the airside system. Note that the parameter ϕ_{kt} is retained in this constraint to represent resource demand not under direct control of the HVAC system. For example, even in the summer, buildings require domestic hot water, and thus the central plant must generate hot water regardless of the fact that temperature zones do not need heating. In this case, $\phi_{kt} > 0$ for $k =$ Hot Water. We refer to such sources as "secondary demand," with the resource consumption in G_{ikt} as "primary demand." Note in addition that the restriction $H_{kt} \leq \phi_{kt}$ is also still included to ensure that H_{kt} is only used for missed secondary demand. Missed primary demand is explicitly not allowed, although the slacks T^{+}_{it} and T^{-}_{it} serve a similar purpose.

For the objective function, we split into a separate waterside term F^{WS} (from (5.1)) and an airside term F^{AS}:

$$F^{\text{WS}} = \sum_{k \in \mathbf{K}} C_k + \sum_{k \in \mathbf{K}} \sum_{t \in \mathbf{T}} \omega_{kt} B_{kt}, \tag{6.5}$$
$$F^{\text{AS}} = \sum_{t \in \mathbf{T}} \sum_{i \in \mathbf{I}} \left(\chi^{+}_{it} T^{+}_{it} + \chi^{-}_{it} T^{-}_{it} \right). \tag{6.6}$$

The overall objective function is thus

$$\min \quad F^{\text{WS}} + F^{\text{AS}}, \tag{6.7}$$

which considers the real utility costs C_k of operating the central plant, as well as the soft constraint violations T^{+}_{it} and T^{-}_{it} for primary demand, and B_{kt} for secondary demand.

The combined optimization problem consists of constraints (5.2), (5.3) and (5.5) to (5.12) from the central plant formulation with modified demand constraint (6.4), as well as additional constraints (6.1) to (6.3), (6.5) and (6.6). The objective function is (6.7). This problem is also a mixed-integer linear programming (MILP) formulation. We refer to this formulation as the combined problem.

6.2.5 Remarks

Remark 6.1: Although the dynamic model in (6.1) treats every zone identically, not every zone needs to represent air space within the building. Because the comfort bounds Θ_{it}^{+} and Θ_{it}^{-} as well as consumption limits Γ_{ikt} can vary by zone, we can use certain temperature zones to model the slower-timescale evolution of the building mass. Such zones cannot receive any direct heating or cooling via G_{ikt}, but their temperature trajectory T_{it} need not satisfy comfort bounds. Thus, by clever choice of zones and coupling coefficients, (6.1) can accommodate arbitrarily higher-order linear models for the temperature of "true" zones.

More generally, the model in (6.1) can be replaced with any arbitrary continuous-variable linear dynamic model provided that the zone temperatures T_{it} and resource consumptions G_{ikt} are outputs of the model. For example, Patel et al. (2016a) suggests including local regulatory controllers in the dynamic temperature model. With this choice, the model inputs are the signals sent to regulatory controllers, while resource consumption is calculated based on the dynamics imposed by these controllers. More complicated models possibly including discrete variables could also be included, although the resulting optimization problem is likely to be difficult to solve, and the decomposition strategies developed in this chapter may not be applicable.

Remark 6.2: As written, the combined problem formulation does not consider direct utility consumption by airside equipment. However, in real systems, AHUs, local building pumps, and other airside units consume electricity to operate. The justification for neglecting these consumption terms is that they are approximately constant. However, if the optimizer makes significant use of passive TES, then there may be periods where airside equipment is operated at higher than usual levels, as well as periods where the equipment is essentially shut off. If such differences are significant, then they could be approximately accounted for by adding extra terms to the demand balance (6.4). For example, adding terms $\sum_{k' \in \mathbf{K}} \pi_{kk'} G_{ikt}$ would allow the modeling of electricity consumption that is proportional to the amount of chilled water used by the airside system.

6.3 Decomposition Strategies

The main difficulty of the combined formulation is the combination of the discrete variables from the waterside model with the large number of continuous variables from the airside model. Compared to the waterside-only model, a similar number of nodes need to be visited to find a comparable solution, but due to the significantly larger number of variables, each node requires more time and memory to process. The end result is that it may not be possible to even find a feasible solution in a reasonable amount of time for large instances. Therefore, we propose decomposition strategies to separate these two main sources of difficulty.

6.3.1 Aggregate Central Plant Models

To start, we describe a method for constructing approximate aggregate models for the central energy facility. Although these models ignore many of the relevant effects of discrete variables (e.g., minimum capacities and dwell times), they provide a reasonable estimate of necessary resource consumption to produce a given amount of chilled or hot water in waterside equipment. Thus, they allow approximate optimization of the central plant without the major source of complexity.

The premise of the aggregate models is that is that in any single time period, the optimizer will choose a configuration of equipment that is optimal or near-optimal for the current total load. Thus, by finding the optimal static equipment allocation for various total production levels, the complexity of piecewise-linear models, auxiliary equipment, intermediate resources, etc., can be summarized in terms of aggregate production and consumption. In general, the central plant will consume a small subset of resources purchased from utility markets and deliver another small subset to the airside. We define these sets as follows:

$$\mathbf{K}^- \coloneqq \{k \in \mathbf{K} : \Pi_k > 0\},$$
$$\mathbf{K}^+ \coloneqq \{k \in \mathbf{K} : \phi_{kt} > 0 \text{ or } \Gamma_{ikt} > 0 \text{ for some } t \in \mathbf{T} \text{ or } i \in \mathbf{I}\}.$$

Here, $\mathbf{K}^-$ are the resources purchased externally, while $\mathbf{K}^+$ are the resources consumed by the airside (or used to meet secondary demand ϕ_{kt}). The goal of the aggregate models is to determine how much of each resource in $\mathbf{K}^-$ must be purchased to produce a given amount of resources in $\mathbf{K}^+$. The remaining resources in $\mathbf{K} \setminus (\mathbf{K}^- \cup \mathbf{K}^+)$ are completely internal to the central plant and thus are used only to determine auxiliary equipment utilization.

To treat the entire central plant with a single aggregate model, one would need a function mapping from $\mathbb{R}^{|\mathbf{K}^-|} \to \mathbb{R}^{|\mathbf{K}^+|}$. For the examples we consider here, $\mathbf{K}^- \coloneqq \{\text{Electricity}, \text{Gas}\}$, and $\mathbf{K}^+ \coloneqq \{\text{Chilled Water}, \text{Hot Water}\}$. Thus, the required model would be two-dimensional. However, for each major type of equipment (chillers, HRCs, and boilers), only one resource $k \in \mathbf{K}^+$ can be independently varied. Therefore, similar to Wenzel et al. (2014), we produce a separate aggregate model for each major type of equipment. For each group, a representative resource $k^* \in \mathbf{K}^+$ is chosen, and the following subproblem is solved with ϕ_{k^*} varying from zero to the maximum total capacity of that type of unit:

$$\min \quad \sum_{k \in \mathbf{K}} \rho_k P_k \tag{6.8a}$$

$$\text{s.t.} \quad \sum_{j \in \mathbf{J}} Q_{jk} + P_k \geq \phi_k, \quad k \in \mathbf{K} \tag{6.8b}$$

$$Q_{jk} = \sum_{m \in \mathbf{M}j} \sum_{n \in \mathbf{N}jm} \zeta_{jkn} Z_{jmn}, \quad j \in \mathbf{J}, k \in \mathbf{K} \tag{6.8c}$$

$$\sum_{m \in \mathbf{M}j} V_{jm} = U_j, \quad j \in \mathbf{J} \tag{6.8d}$$

$$\sum_{n \in \mathbf{N}jm} Z_{jmn} = V_{jm}, \quad j \in \mathbf{J}, m \in \mathbf{M}_j \tag{6.8e}$$

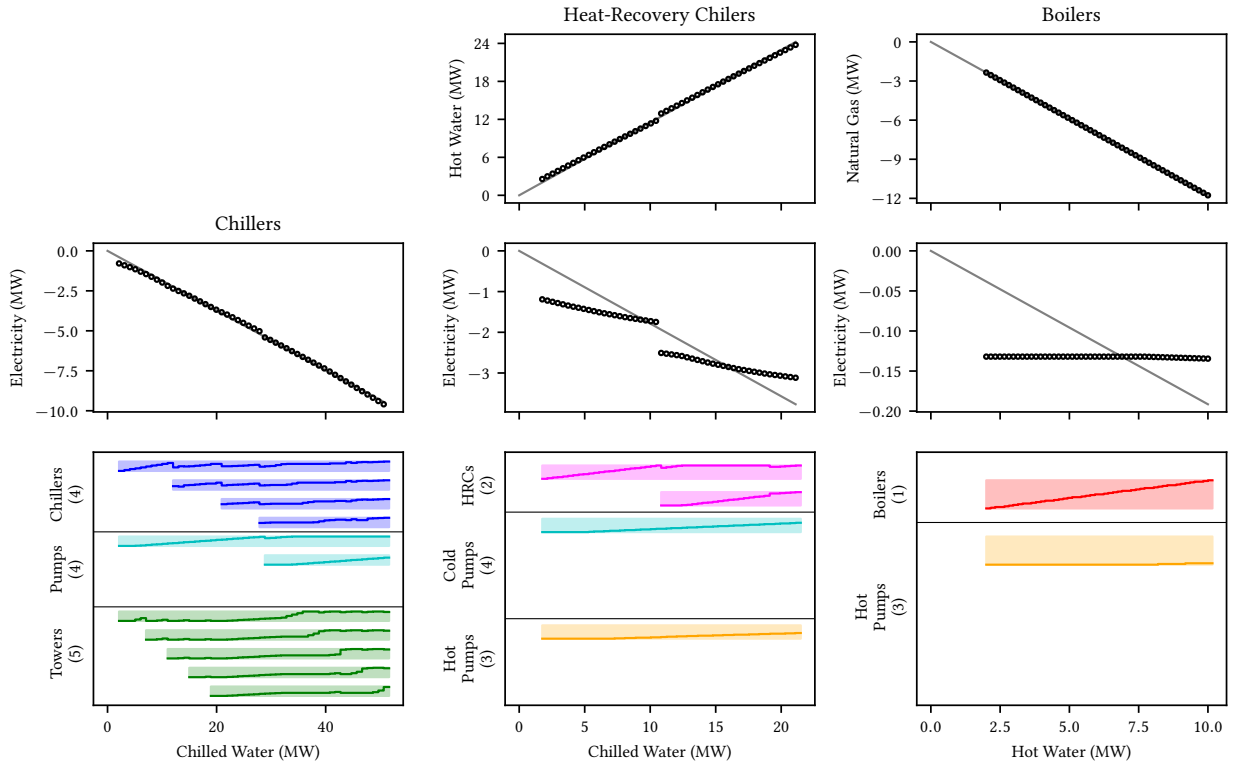


FIGURE 6.1: Example aggregate system curves. Circular markers are solutions to (6.8), while gray lines are linear fits for $\Xi_{jkt}(\,\cdot\,)$. Central plant equipment is as shown in Figure 5.6.

Note that other types of major equipment are excluded, but all other auxiliary equipment (i.e., that does not produce resources in $\mathbf{K}^+$) are retained. Using the optimal values of P_k from each solution to (6.8), an aggregate function $\Xi_{jkt} : [0,1] \to \mathbb{R}^{|\mathbf{K}^+ \cup \mathbf{K}^-|}$ with the independent variable $Z \in [0,1]$ choosing the load between 0% and 100% for that type of equipment. Note that the index j refers to the major type of equipment, rather than any particular piece of equipment. The goal of these aggregate models is that for a given level Z, $\Xi_{jkt}(Z) \approx \phi_k$ for $k \in \mathbf{K}^+$ and $\Xi_{jkt}(Z) \approx P_k$ for $K \in \mathbf{K}^-$; that is, the aggregate function $\Xi_{jkt}(\,\cdot\,)$ is an approximation of the optimal operating points determine from (6.8). We provide an example of an aggregate operating curve in Figure 6.1. Although relative error is somewhat high for certain resources, we will demonstrate via simulation, that quality solutions can still be found using these approximations.

To account for possible time variability in equipment efficiency, we include a time index t on the aggregate models $\Xi_{jkt}(\,\cdot\,)$. For the best accuracy, a different aggregate model could be constructed for each different set of ambient conditions. However, if a representative sampling can be constructed offline, then no instances of (6.8) need be solved online. It is also possible that different values for utility prices ρ_k could affect optimal configurations. However, since each major piece of equipment consumes only one major resource, the choice of ρ_k in (6.8) is largely irrelevant. Finally, because we want to use these models in other optimization problems, it is advantageous to choose the functions

$\Xi_{jkt}(\,\cdot\,)$ to be convex. In particular, if they are *linear*, then the aggregate models can be used in other LP problems, which increases applicability. We will use these models for two separate cases as discussed in the following sections.

6.3.2 Hierarchical Decomposition

As alluded to in the introduction, we can improve solution speed by separating the large continuous-variable airside model from the modestly sized but mixed-integer waterside model. In particular if the airside resource consumption variables G_{ikt} are *fixed*, then the resulting optimization problem is equivalent to an instance of the original central plant problem that can be solved in a reasonable abount of time. To achieve this goal, we use a hierarchical decomposition that splits the problem into an upper-level centralized subproblem without any discrete variables and a lower level waterside subproblem without the large airside model.

For the centralized subproblem, we replace the discrete on/off variables and piecewise-linear generator models with the aggregate equipment models developed in the previous section. This choice leads to the simplified demand constraint

$$\sum_{j\in\mathbf{J}} \Xi_{jkt}(Z_{jt}) + Y_{kt} + P_{kt} + H_{kt} \geq \phi_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}. \tag{6.9}$$

Note that the index $j \in \mathbf{J}$ refers to the major types of equipment, not individual generators. In addition, the set of resources need only contain $\mathbf{K}^+ \cup \mathbf{K}^-$, as the remaining resources are implicitly accounted for in the aggregate models. The centralized subproblem thus consists of constraints (5.2), (5.3) and (5.11) to (5.12) from the central plant model, (6.1) to (6.3), (6.5) and (6.6) from the airside model, and the simplified (6.9). With objective function (6.7), the centralized subproblem is a linear programming problem, assuming the aggregate models $\Xi_{jkt}(\,\cdot\,)$ are linear.

After solving the centralized subproblem to determine near-optimal values for G_{ikt}, the total resource demand $\phi_{kt} + \sum_{i\in\mathbf{I}} G_{ikt}$ can be calculated as the sum of primary and secondary demands. This value of total demand is used as the parameter ϕ_{kt} in the original central plant optimization from Chapter 5, which is solved as the waterside subproblem. Note that the storage tank model is included in both subproblems; it is part of the centralized subproblem so that active and passive TES are considered simultaneously, while it is included in the waterside subproblem so that the dynamics can be considered using the higher accuracy of the piecewise-linear equipment models. By combining the values of T_{it} and G_{ikt} from the centralized subproblem with the optimal solution to the waterside subproblem, a feasible solution to the combined problem is obtained, giving an upper bound on the optimal combined objective. This process is illustrated in Figure 6.2. Note that it is also possible to find a lower bound for the combined problem, which will be discussed as part of the Lagrangian decomposition strategy.

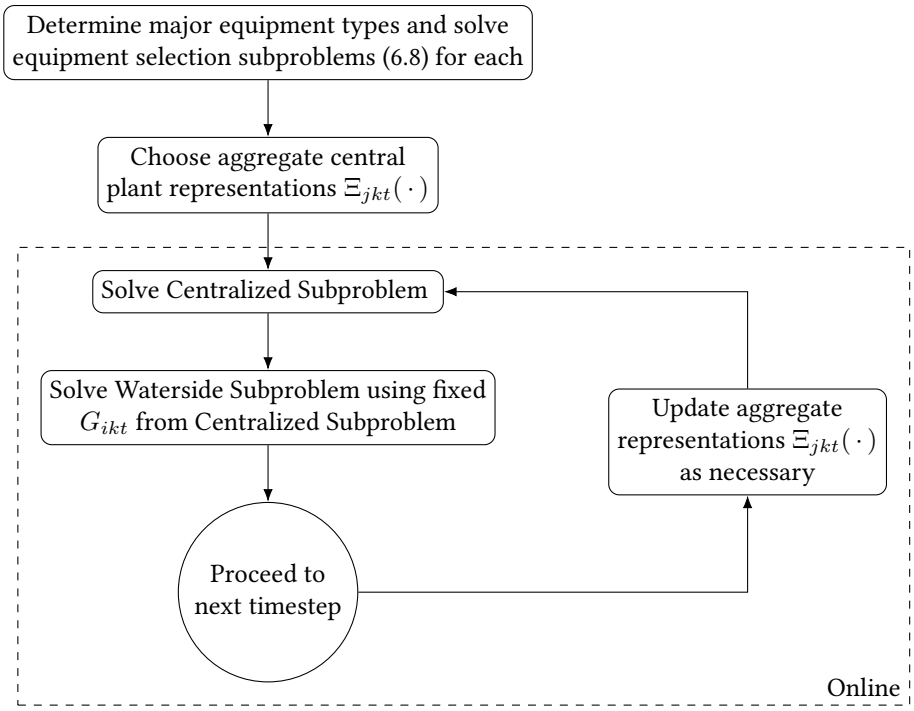


FIGURE 6.2: Algorithm for hierarchical decomposition.

6.3.3 LAGRANGIAN DECOMPOSITION

While the hierarchical decomposition from the previous section effectively decouples the two challenging aspects of the combined problem, it is predicated on the ability to construct accurate aggregate curves $\Xi_{jkt}(\cdot)$. In addition, in the centralized subproblem, every single temperature zone has to be included, even zones in completely separate buildings that do not interact dynamically. To avoid these potential issues, we propose an alternative decomposition strategy based on Lagrangian relaxation.

As mentioned in the previous case, the only source of coupling in the combined formulation is the demand constraint (6.4). To start, we slightly reformulate the problem as follows: constraint (6.4) is split into two pieces

$$\sum_{j \in \mathbf{J}} Q_{jkt} + Y_{kt} + P_{kt} + H_{kt} \geq \phi_{kt} + \bar{G}_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}, \tag{6.10}$$

$$\bar{G}_{kt} = \sum_{i \in \mathbf{I}} G_{ikt}, \quad k \in \mathbf{K}, t \in \mathbf{T}, \tag{6.11}$$

in which $\bar{G}_{kt}$ are new variables to represent the total primary demand across all temperature zones. With this reformulation, the only source of airside/waterside coupling is (6.11). To split the problem, we decompose this constraint in the Lagrangian sense; that is, we remove the constraint, choose multipliers λ_{kt}, and any deviation in the objective function.

With the relaxation of (6.11), we end up with two completely separate subproblems: the airside dual subproblem consisting of constraints (6.1) to (6.3) with objective function

$$\min \quad \sum_{t\in\mathbf{T}}\left(\sum_{i\in\mathbf{I}}\left(\chi_{it}^{+}T_{it}^{+}+\chi_{it}^{-}T_{it}^{-}+\sum_{k\in\mathbf{K}}\lambda_{kt}G_{ikt}\right)\right), \tag{6.12}$$

and the waterside dual subproblem with constraints (5.2), (5.3) and (5.5) to (5.12) and (6.10), using the objective function

$$\sum_{k\in\mathbf{K}}\left(C_k+\sum_{t\in\mathbf{T}}\left(\omega_k B_{kt}-\lambda_{kt}\bar{G}_{kt}\right)\right). \tag{6.13}$$

It can be shown that for any value of the multipliers λ_{kt}, the optimal objective value for the dual subproblems is a lower bound for the optimal combined objective function. Note that better bounds could potentially be obtained by copying all of the G_{ikt} variables and decomposing the individual equality constraints, rather than decomposing (6.11) directly (Guignard and Kim, 1987). However, this technique did not lead to improved performance in the examples tested.

Using the values of G_{ikt} from the airside dual subproblem, we can find a feasible solution to the combined problem (and thus an upper bound) by solving an instance of the waterside (primal) subproblem just as in the hierarchical decomposition. It is also possible to update λ_{kt}, and the process can be repeated to potentially find better lower or upper bounds. For example, using subgradient methods (Wolsey, 1998), a possible update formula is

$$\lambda_{ikt} \coloneqq \lambda_{kt}+\mu\left(\sum_{i\in\mathbf{I}}G_{ikt}-\bar{G}_{kt}\right),$$

with various selection rules available for the stepsize μ. Once a suitable solution has been obtained, the process is stopped, and the multipliers are shifted for use in the next timestep. We illustrate this process in Figure 6.3.

The main benefits of this method over the hierarchical decomposition is that no approximations are necessary, and this method can theoretically find proven optimal solutions to the combined problem. Unfortunately, due to nonconvexity of the waterside model, there is likely a nonzero duality gap for this problem, which means the gap between lower and upper bounds never closes. However, as in the central plant optimization, it is sufficient to find solutions with a gap of roughly 1%, which is attainable in a single iteration if a quality guess for λ_{kt} is used. Of course, the downside of this approach is that iteration may be necessary if the multipliers are highly suboptimal, whereas the hierarchical decomposition does not require iteration. A possible benefit of this approach is that the airside dual subproblem can be decomposed on a per-building basis. Because zones in one building do not interact with zones in another building, the dynamic model coefficients $\alpha_{ii'}$ and $\beta_{ii'k}$ are block-diagonal, which means the decisions made in one building are completely independent from others. This property can be very useful if there a very large number of separate buildings.

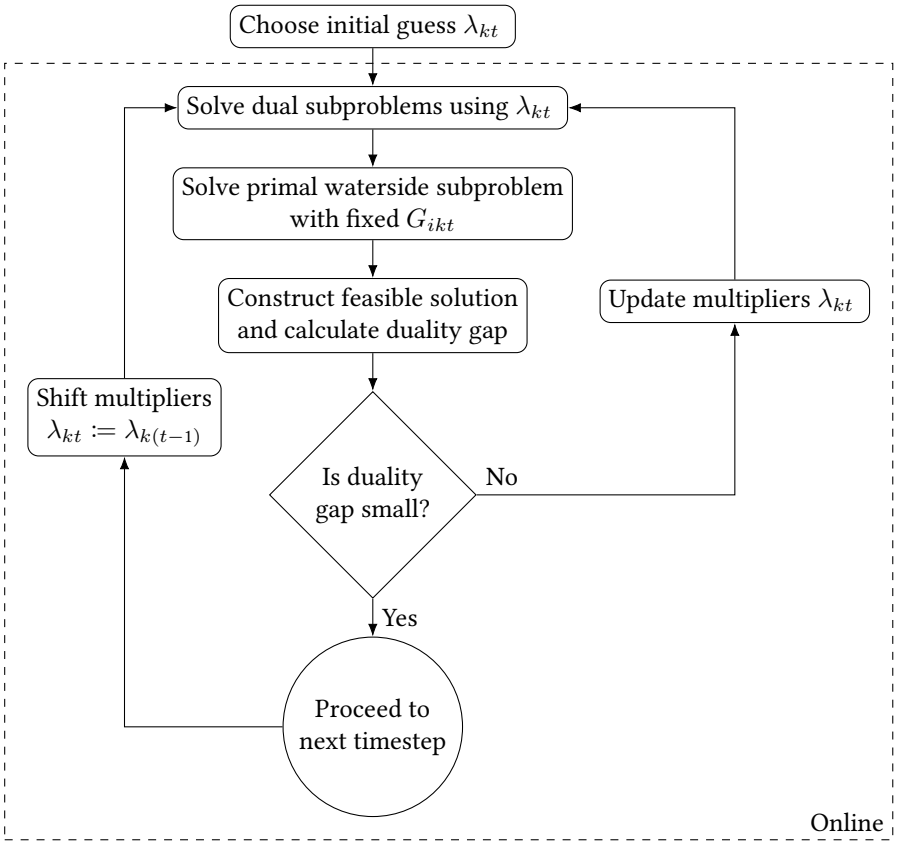


FIGURE 6.3: Algorithm for Lagrangian decomposition.

Remark 6.3: The Lagrangian decomposition can be viewed as a price negotiation between the airside and waterside systems. The multipliers λ_{kt} give the (time-varying) price of chilled and hot water to the airside subsystem; in the airside dual subproblem, the optimizer minimizes its cost, possibly pre-cooling to take advantage temporal differences in chilled water costs. The multipliers also give the revenue received by the waterside subsystem for providing those resources; thus, in the waterside dual subproblem, the optimizer decides how much of each resource to deliver, in order to maximize its profit. By choosing a price where the airside consumes the same amount as the waterside would like to produce, a near-optimal solution can be found. If the two sides are far apart, we can directly find a feasible combined solution by forcing the waterside to meet the airside's current resource demand.

Remark 6.4: Since the combined solution to the airside and waterside dual subproblems provides a lower bound on the optimal objective of the combined problem, they can be used to estimate a lower bound for the hierarchical decomposition. Of course, the hierarchical decomposition does not directly use the multipliers λ_{kt}, but a reasonable guess can sometimes be obtained from the dual multipliers on constraint (6.9) in the centralized problem. This bound is likely to be weaker than the bound obtained from optimizing λ_{kt} directly, but it still may be sufficient for online closed-loop implementation.

6.3.4 Stochastic Demand Charge Optimization

As suggested in Section 3.4, the best way to address peak charges is to include the term in the objective function without any adjusted weighting, but also to include a lower bound that represents the past peak value *and* a forecast for the future remaining peak. While inclusion of the past peak is trivial, it is difficult to determine a good value for the future peak without considering the entire peak charge window, which is typically one month or longer. In addition, even if a reduced problem can be formulated, forecasting price and heating/cooling demand is likely to suffer significant uncertainty, and thus deterministic optimization is likely to provide poor estimates.

To address these issues, we propose a simple stochastic programming formulation to determine peak charge targets. Stochastic programming allows for optimization under uncertainty, and the technique has been proposed for optimizing electricity market participation using batteries (Dowling et al., 2017; Kumar et al., 2018), which contains many of the same elements as the current problem. To start, the peak charge window is split into multiple stages $s \in \mathbf{S}$. Each stage is given multiple realizations $r \in \mathbf{R}_s$ of uncertain parameters ρ_{kt} and ϕ_{kt}. For each stage and realization, the following problem formulation is as follows (we omit indices r and s for clarity).

$$\sum_{j \in \mathbf{J}} \Xi_{jkt}(Z_{jt}) + Y_{kt} + P_{kt} + H_{kt} \geq \phi_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}, \tag{6.14a}$$

$$S_{kt} = \sigma_k S_{k(t-1)} - Y_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}, \tag{6.14b}$$

$$P_k^{\max} \geq P_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}, \tag{6.14c}$$

$$P_k^{\max} \geq P_{k0}^{\max}, \quad k \in \mathbf{K}. \tag{6.14d}$$

Here, the indices $j \in \mathbf{J}$ refer to the aggregate equipment groups as discussed in Section 6.3.1, and $\Xi_{jt}(\,\cdot\,)$ are the corresponding aggregate models (with variable $Z_{jt} \in [0,1]$ choosing how much to utilize each group). Note that the initial conditions S_{k0} and $P_{k0}^{\max}$ are inherited from previous stages (or the current plant conditions in the case of the first stage).

Denoting the values S_{k0} and $P_{k0}^{\max}$ as the system state x and taking all other variables as inputs u, the subproblem (6.14) essentially gives a model of the form $x_s = f(x_{s-1}, u_s)$. Using these variables, we can thus construct a scenario tree of possible parameter realizations and optimize. Specifically, this tree structure imposes constraints of the form

$$x_{rs} = f_{rs}(x_{\mathfrak{r}_{rs}(s-1)}, u_{rs}), \quad s \in \mathbf{S}, r \in \mathbf{R}_s,$$

in which $\mathfrak{r}_{rs} \in \mathbf{R}_{s-1}$ gives the ancestor realization from stage $s-1$. For the case of a monthly demand charge, each stage could be a week in length, with three unique sets of parameters for each (e.g., parameters for colder temperatures, average temperatures, and hot temperatures). These choices give rise to the scenario tree in Figure 6.4, in which there is 1 realization in Stage 1, 3 realizations in Stage 2, 9 realizations in Stage 3, and 27 realizations in Stage 4. Note that without loss of generality, we can assume there is only a single realization in the first stage, as multiple first-stage realizations can be addressed by solving completely separate problem instances.

With a scenario tree established, the objective function is

$$\min \quad \sum_{s\in\mathbf{S}} \sum_{r\in\mathbf{R}} \left(\pi_{rs} \sum_{t\in\mathbf{T}_s} (\rho_{kt} P_{kt})_{rs} \right) + \sum_{r\in\mathbf{R}_{s^*}} \left(\pi_{rs^*} \sum_{k\in\mathbf{K}} (\rho_k^{\max} P_k^{\max})_{rs^*} \right).$$

in which π_{rs} is the probability of realization r in stage s, and s^* is the index of the final stage. Thus, the objective function calculates the average total monthly charges, including use charges throughout all stages, and the monthly demand charges as determined by the value of $P_k^{\max}$ in the final stage. In the case of linear aggregate equipment functions $\Xi_{ijt}(\,\cdot\,)$, the entire problem is a large but manageable linear programming problem. As long as the $\Xi_{ijt}(\,\cdot\,)$ are *convex*, a variety of stochastic programming decomposition strategies can be employed (e.g., Rockafellar and Wets (1991); Shapiro (2011)). After finding an optimal solution, the values of $P_k^{\max}$ from each stage can be used as peak utility targets (e.g., mean or median values across realizations in each stage).

6.4 Simulations

We conclude this chapter with some example simulations of the proposed formulations. All optimizations (both MILP and LP) are solved using Gurobi (Gurobi Optimization, LLC, 2018). All timesteps are 1 h.

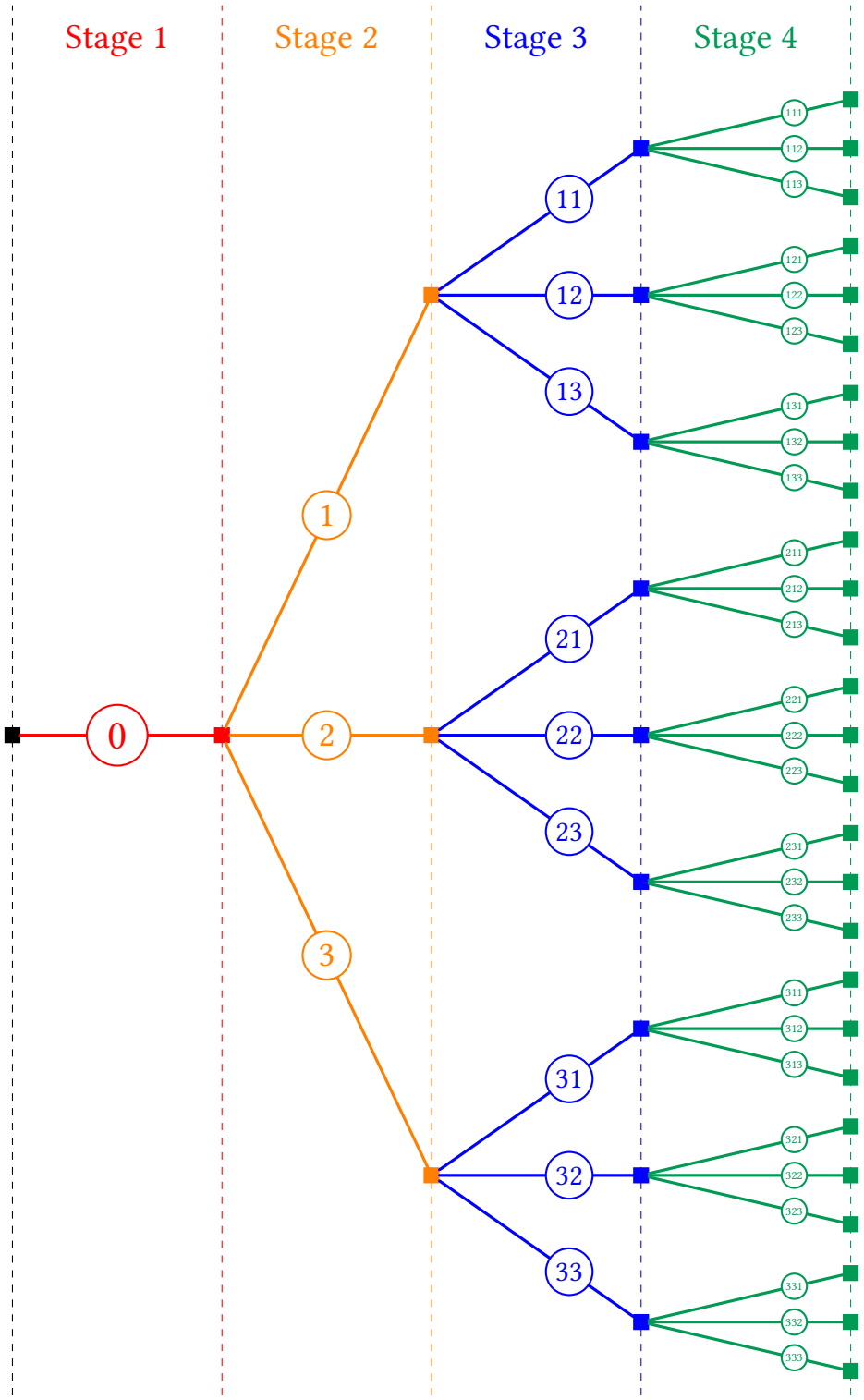


FIGURE 6.4: Example scenario tree. Numbers in circles are realizations r. Ancestor realization $\mathfrak{r}_{rs}$ is the value of r for the parent arc. The final stage is $s^* = 4$.

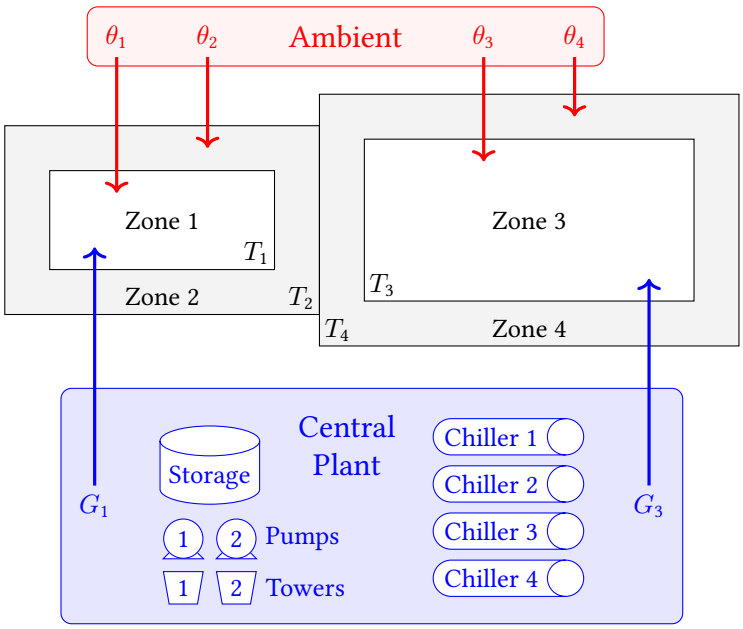


FIGURE 6.5: Example combined airside/waterside system. Zones 1 and 3 are air zones, while Zones 2 and 4 are mass zones. The ambient disturbance parameter θ_i includes ambient conduction, ambient radiation, and internal heat generation for each zone.

6.4.1 COMBINED EXAMPLE

To start, we consider an instance of the combined airside/waterside problem. The central plant consists of only cooling equipment (conventional chillers, cooling towers, and pumps). Following the advice of Remark 6.1, we model the air and solid mass of each temperature region as a separate region. The air zones receive cooling via the G_{ikt} variables that must offset direct heat generation via θ_{ikt}, and conductive interaction with the corresponding mass zones. Mass zones do not receive any direct cooling, but their temperatures need not satisfy the temperature bounds; these zones couple conductively with adjacent mass zones as well as the ambient. A diagram of the system is shown in Figure 6.5.

Solving the combined problem gives the optimal solution shown in Figure 6.6. From this solution, we see that the temperatures of the air zones (1 and 3) remain at their upper bounds throughout the horizon. Thus, the active TES capacity is large enough to provide optimal load shifting, and thus passive TES is not utilized. Note that the mass zones (2 and 4) follow the ambient temperature trajectory more closely, since they are not directly heated or cooled.

To assess the scalability of the combined formulation, we vary the number of zones and repeat the optimization problem. Problem size and solution information are shown in Table 6.1. From these results, we see that the optimization problem does become significantly harder as the number of zones is increased. Although the number of integer variables is constant, the growing number of continuous variables and constraints leads to much slower solution

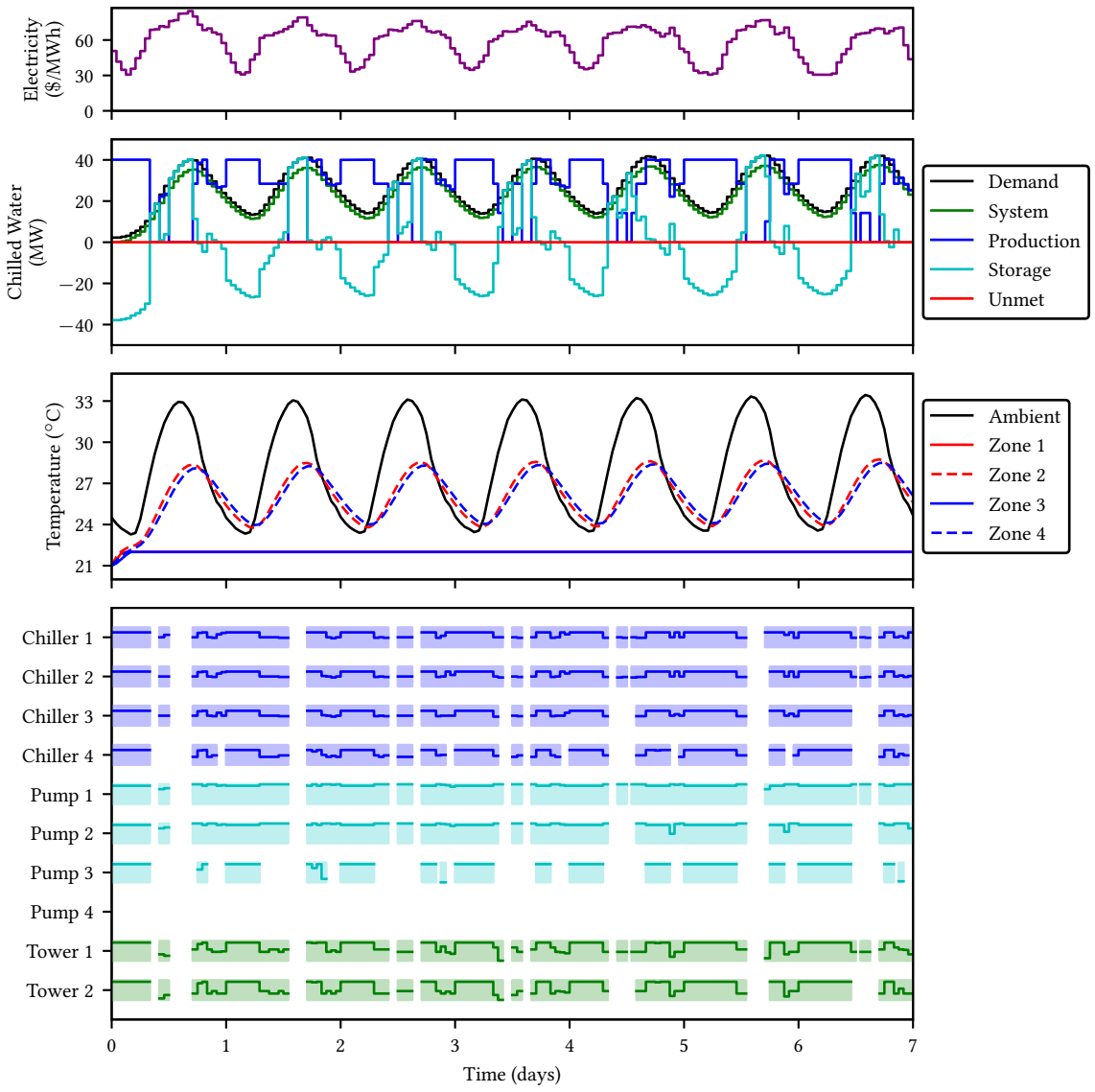


FIGURE 6.6: Optimal solution for the combined airside/waterside system from Figure 6.5. The "System" curve refers to primary demand by the airside system, while "Demand" gives total (i.e., primary and secondary) demand for the central plant.

Table 6.1: Problem size and solution time information for combined formulation. Sizes reported are after Gurobi presolve reductions. The "No Building" line is a waterside-only optimization that meets the optimal demand from the 25-zone example.

Size	Variables ($\times 10^3$)	Integers ($\times 10^3$)	Constraints ($\times 10^3$)	1% Gap Time (s)	Final Gap (after 1000 s)
4 Zones	8.9	2.8	5.7	93.1	0.06%
16 Zones	13.9	2.8	9.7	134.8	0.13%
30 Zones	19.8	2.8	14.4	164.0	0.45%
50 Zones	28.1	2.8	21.1	281.0	0.53%
Waterside	7.2	2.8	4.3	16.9	0.05%

progress. Thus, while these times may be sufficiently fast, for large buildings or campuses with many more zones, solving the combined problem is likely not a viable option. Therefore, decomposition strategies need to be applied.

6.4.2 Hierarchical vs. Lagrangian Decomposition

To examine the effectiveness of the decomposition strategies, we apply the techniques to a large airside system. The airside system consists of 20 separate buildings, each with 25 air and 25 mass zones, for a total of 1000 temperature zones. The central plant consists of 5 chillers, 3 HRCs, 2 boilers, 6 chilled water pumps, 5 hot water pumps, and 5 cooling towers. In addition to the primary chilled water demand required by the airside, there is high secondary demand for hot water and low secondary demand for chilled water. Thus, the central plant must provide chilled and hot water simultaneously. The horizon is 7 days. Due to the large size of this problem, the optimizer is unable to find a feasible solution to the combined formulation within 2 h, and thus decomposition is necessary.

Both the hierarchical and Lagrangian decomposition strategies are able to find a feasible solution to the problem. Figure 6.7 shows the solution obtained by the hierarchical decomposition; a similar solution is found by the Lagrangian decomposition. In this figure, we note that the solution does not make use of passive TES on the airside, as there there is a high-capacity active TES for chilled water. Problem size information is shown in Table 6.2 and Table 6.3. We note from the problem sizes that the centralized and airside subproblems are large continuous-variable models, while the waterside primal and dual problems are moderately sized mixed-integer models. From the performance table, we note that the two decomposition strategies find roughly the same solution, but the duality gap is smaller for the Lagrangian decomposition. Solution times are roughly the same, although the bulk of the time is spent in the waterside subproblems, which were both given a time limit of 10 min. Finally, we note that in this case, running an additional iteration of the Lagrangian decomposition does not lead to an improved solution.

To assess the viability when no active TES is available, we repeat the calculation for a central plant without hot or chilled water storage tanks. To make up for this effective loss of hot water capacity, the system is given two additional boilers. We show the hierarchical solution in Figure 6.8. The system does make use of passive TES, and the building

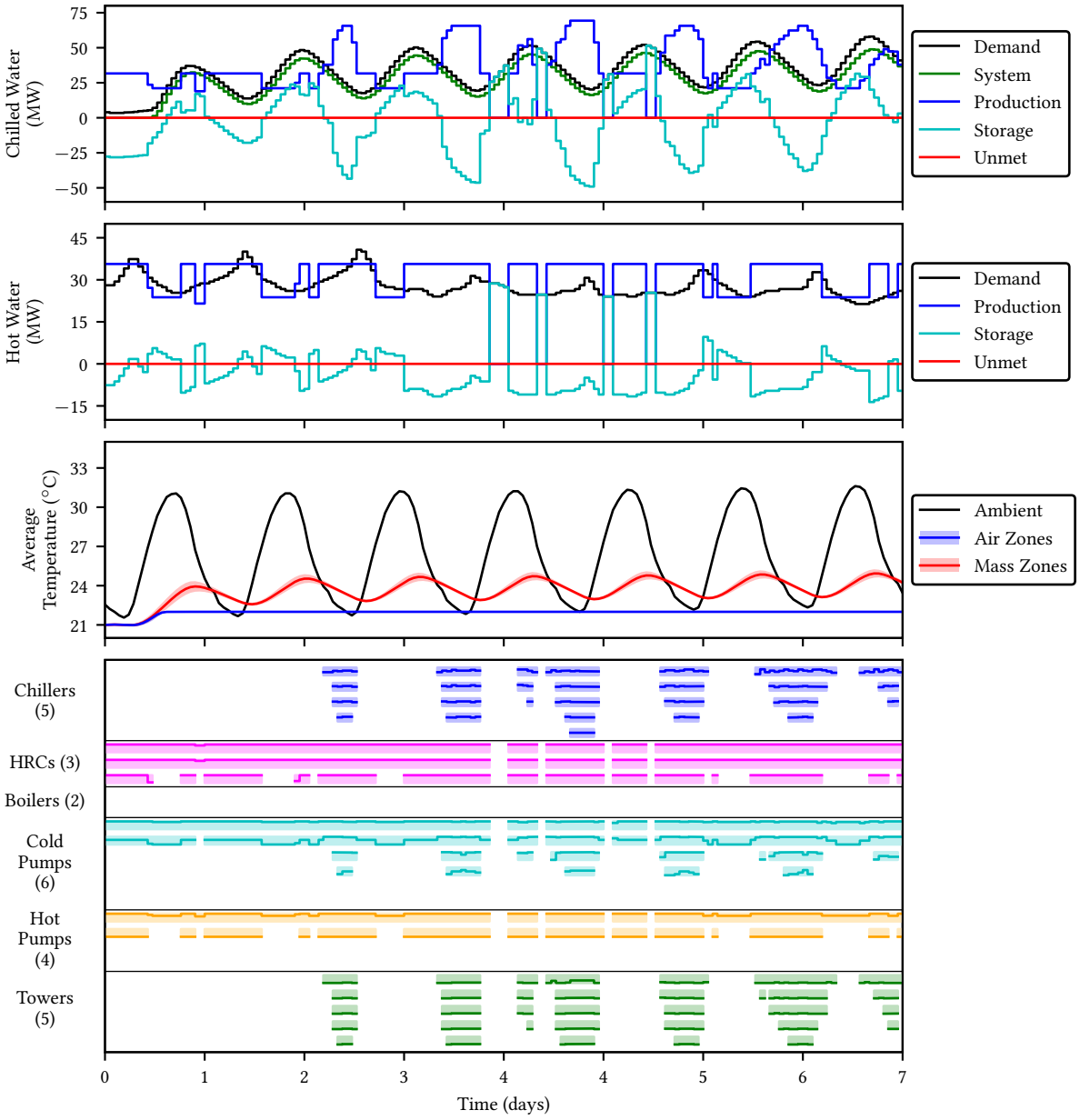


FIGURE 6.7: Solution to the large heating/cooling problem using the hierarchical decomposition with active TES. The solution obtained from the Lagrangian decomposition is similar.

Table 6.2: Subproblem sizes for the hierarchical decomposition strategy applied to the large heating/cooling example. Values are as reported by Gurobi after presolve reductions. Sizes for the Lagrangian decomposition strategy are nominally the same, although presolve reductions may be slightly different due to non-identical problem data.

Subproblem	Variables ($\times 10^3$)	Integers ($\times 10^3$)	Constraints ($\times 10^3$)	Nonzeros ($\times 10^6$)
Centralized	252.90	0	168.22	2.05
Airside Dual	250.72	0	166.72	1.96
Waterside Primal	13.57	5.35	8.18	0.05
Waterside Dual	13.73	5.35	8.17	0.05

Table 6.3: Solution time and gap information for the large heating/cooling example. Objective functions are normalized to the hierarchical decomposition solution. The bottom block gives solution times in minutes.

	Hierarchical	Lagrangian (Iteration 1)	Lagrangian (Iteration 2)
Objective	100.00	99.99	99.99
Duality Gap	1.82	0.43	0.43
Centralized	0.31	—	—
Waterside Primal	10.02	10.00	10.00
Waterside Dual	10.02	10.02	10.02
Airside Dual	0.10	0.09	0.09
Total	20.45	20.12	20.12

is pre-cooled slightly overnight. Unfortunately, the Lagrangian decomposition does not work as well for this problem, finding a solution that is 10% worse than that of the hierarchical solution. A guess for the dual multipliers is obtained by solving the LP relaxation of the waterside problem; due to the presence of HRCs, these multipliers can be negative, leading to overly aggressive pre-cooling for the airside The duality gaps for both strategies are above 10%. Thus, while the hierarchical decomposition can still find quality primal solutions, the resulting optimality gaps are poor unless better multipliers can be determine.

To assess the impact of active TES on solution quality and optimizer performance, we repeat the previous calculations for varying chilled water active TES capacities. To avoid the complications of HRCs, the central plant does not consider hot water production; thus, the waterside equipment consists of 8 chillers, 8 cooling towers, and 6 pumps. The airside system is as in the previous example. Figure 6.9 shows solution information for this simulation. Note that the Lagrangian decomposition uses only a single iteration for these results. From the top axes, we see that total cost decreases monotonically as the capacity of active TES increases. In particular, cost is reduced by more than 20% when the full chilled water storage tank is present compared to not having a storage tank at all. For the performance of the two decomposition strategies, we see that the Hierarchical decomposition generally finds better upper bounds, while the Lagrangian decomposition finds better lower bounds. For the upper bounds, the two strategies become closer as the TES capacity is increased (and the Lagrangian actually wins slightly at 100% TES), but the solutions are always within 1% of each other. For the lower bounds, the Lagrangian decomposition is significantly

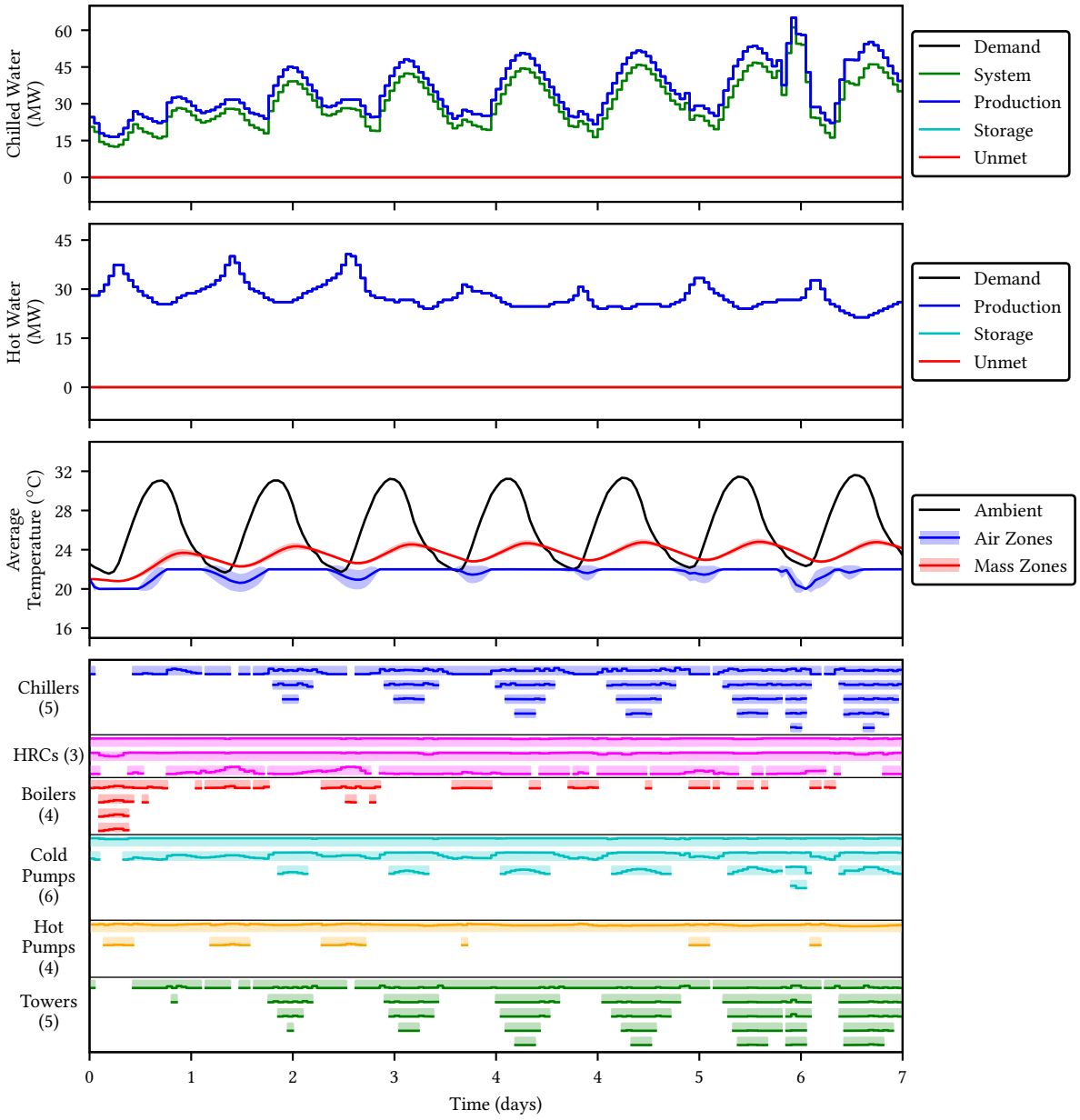


FIGURE 6.8: Solution to the large heating/cooling problem using the hierarchical decomposition without active TES. Temperature plot shows average ± standard deviation for zone temperatures.

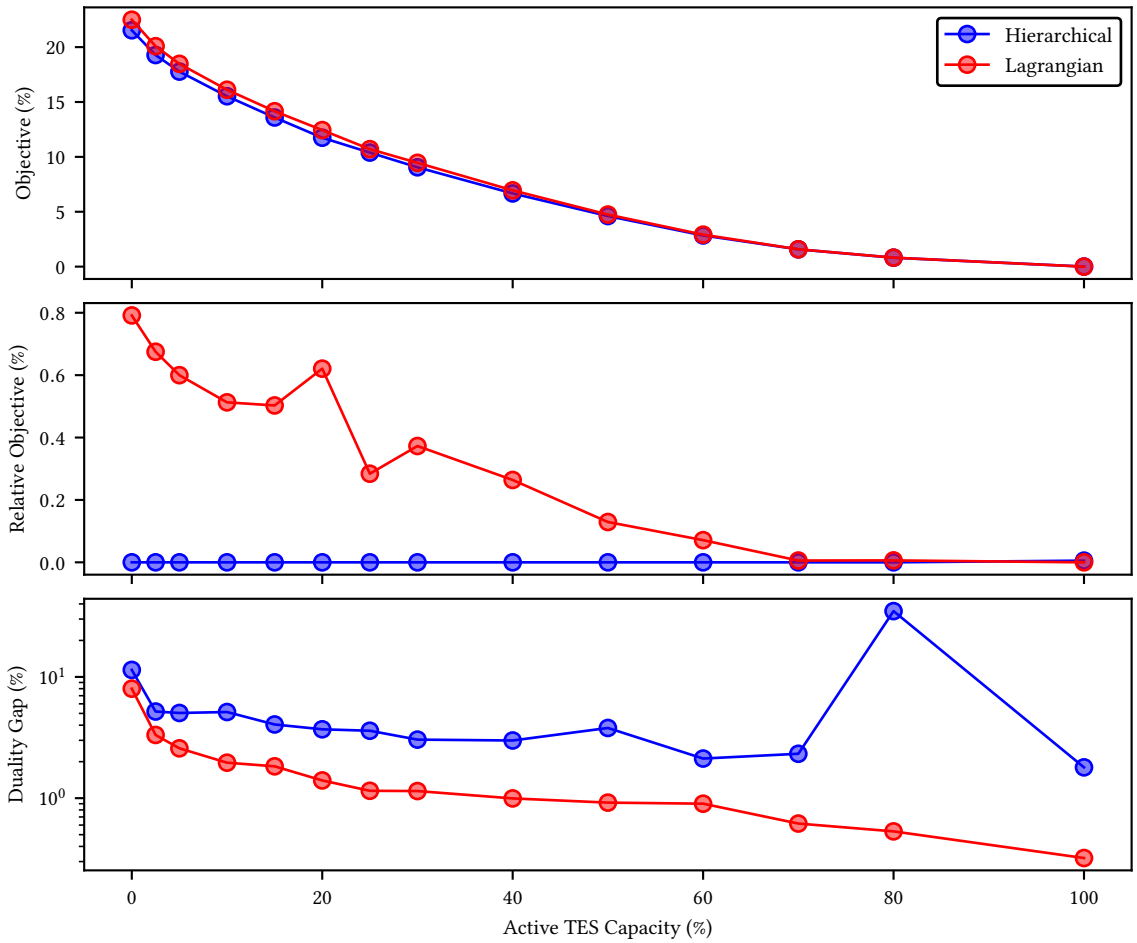


FIGURE 6.9: Objective function and bounds information for the Hierarchical and Lagrangian decomposition strategies applied to a large system. Relative objective uses Hierarchical solution as a baseline.

better above roughly 20%. Overall, neither strategy is significantly better than the other. In general, if lower bounds are unnecessary, then the Hierarchical decomposition may be the better choice, but the Lagrangian decomposition does provide more flexibility for implementation.

6.5 SUMMARY

In this chapter, we have extended the central plant optimization formulation from Chapter 5 to include optimization of airside temperature trajectories. By embedding a dynamic model for building temperature dynamics within the problem formulation, the optimizer can take advantage of passive TES by pre-cooling temperature zones to reduce cooling requirements later. For small to medium-sized airside systems, the combined problem can be solved directly to optimally trade off use of active and passive TES. However, for large airside systems, the combined model cannot be solved within the given time requirements. Thus, we have presented two strategies to decompose into more

manageable subproblems. In the first strategy, the central plant is first optimized using approximate aggregate equipment models without discrete variables; using that airside demand, the waterside system can be optimized without including an airside model. In the second strategy, the coupling constraints between the airside and waterside models are dualized in the Lagrangian sense; using a good guess for dual multipliers, quality solutions can be found by optimizing the airside and waterside subsystems independently. Finally, we have presented a stochastic programming formulation for long-term peak demand charge calculation with uncertain forecasts. Solving this problem gives a target for peak demand that can be used with the ideas of Section 3.4 for closed-loop demand charge optimization.

For additional information about the airside problem formulation, including the building dynamic model, see Risbeck et al. (2017). For more solution timing statistics, see Risbeck et al. (2018b).

Summer's heat or winter's cold
The seasons pass; the [horizon] will roll

— Fred Cornell
Carmen Ohio

Chapter 7

Conclusions

7.1 Outlook

The world is getting faster. What used to take a day in the library can now be accomplished by a single internet search. Utility markets move drastically in seconds, due to both the unpredictability of the weather and the caprice of energy speculators. The price of steel can jump 25% overnight. Conditions change, priorities realign, and assumptions may no longer hold. Fortunately, as the speed of life is increasing, so too is the speed of computers and optimization algorithms. Even within the past five years, problems that started out intractable are now considered trivial. Thus, by relying increasingly on automated decision-making, we can now respond optimally and in real time to external changes or disturbances, however big or small.

The work presented in this document serves to expand the applicability of and builds momentum behind the increasing use of online optimization. In the HVAC application, we suggest replacing fragile heuristics with resilient mathematical programming methods. For scheduling problems, we advocate a control-oriented perspective to provide recursive feasibility and eliminate pathological behavior. Even in something as esoteric as the definition of asymptotic stability, we use a generalized norm rather than a true norm to allow greater flexibility in problem formulation. We hope that these ideas can spur some additional creativity on the part of the greater research community and lead to more innovative and effective solutions to our ever-growing list of problems. In that vein, we conclude this work by summarizing the main contributions and providing some suggestions for further investigation.

7.2 Summary

In Chapter 2, we have presented a formulation of mixed-integer MPC for tracking problems. By avoiding the historically common assumption of the setpoint lying on the interior of the input set $\mathbb{U}$, the extension of stability theory to discrete actuators is immediate. To provide full generality, we present both optimal and suboptimal MIMPC in time-varying formulations. Via examples, we illustrate that suboptimal MPC is adequate for stabilization, although better performance is usually experienced when optimal solution are found.

In Chapter 3, we extend MIMPC theory to demonstrate inherent robustness to state disturbances and to derive closed-loop properties for economic MPC. With inherent robustness, we show that stability of suboptimal MPC cannot be destroyed by arbitrarily small disturbances. By formulating stability in terms of a *generalized norm* on the extended state $(x, \mathbf{u})$, we avoid the restriction that $\mathbf{u} \to 0$ as $x \to 0$, allowing discrete actuators to vary within arbitrarily small neighborhoods of the origin; of course, this property may or may not be desirable depending on the particular application, but the flexibility is available. For economic MPC, we extend asymptotic performance and dissipativity-based stability theorems to cover the time-varying mixed-integer case. We then present an extended-state formulation and terminal cost to allow peak-charge-style terms in the objective functions. The robustness and asymptotic performance results are illustrated using simplified energy system optimization examples.

In Chapter 4, we have demonstrated how standard production scheduling problems can be formulated as state-space control problems. We first illustrate, via a simple batch production example, how catastrophic open-loop/closed-loop mismatch can occur when closed-loop scheduling is implemented without terminal constraints. We then discuss a state-space formulation for integrated scheduling and control, i.e., when the scheduling units are defined by underlying dynamic models. To allow feasible solutions to be obtained without the need to solve the full integrated problem, we present a dynamic-aware extended scheduling model that models a subset of the feasible dynamic space while remaining computationally tractable. Finally, we demonstrate closed-loop properties for scheduling and integrated problems using the example of multi-product reactors with shared resource constraints.

In Chapter 5, we have developed a general MILP model for real-time cost optimization in central energy facilities. A key feature of the model is that it determines both discrete on/off decisions for equipment as well as continuous operating point selection (subject to nonzero minimum capacities). Using an abstract representation of the plant in terms of resources and generators, the formulation is flexible and can consider a wide variety of equipment type. To calculate generator resource consumption, we use piecewise-linear approximations of nonlinear equipment models. These approximations can be tuned to provide the desired level of accuracy. Additional decision variables such as supply temperature can be considered by adding extra dimensions to the piecewise-linear model. Using various example systems, we demonstrate that quality solutions to this model can be found within minutes for realistically

sized systems. In addition, we illustrate good agreement between the optimal equipment configurations predicted by the piecewise-linear models compared to the true nonlinear models subject to closed energy balances. These properties indicate that this formulation is suitable for online use and can lead to cost reduction compared to existing heuristic strategies.

In Chapter 6, we have augmented the central energy facility model to include optimization of building temperature trajectories. By including dynamic models for building temperature evolution, building cooling loads are now variables (determined by airside operational decisions) rather than fixed parameters. Utilization of passive storage techniques like pre-cooling are implicit in the optimization formulation. To address problems with a large number of zones, we have developed two decompositions to separate the airside and waterside into subproblems: a hierarchical decomposition that uses an approximate model for the central plant, and a Lagrangian decomposition that dualizes the airside/waterside coupling constraints. Finally, we present a simple stochastic programming formulation for month-long peak demand optimization. Through example simulations, we demonstrate the capability of these techniques.

Overall, we have developed control theory to expand the scope of application of model predictive control to systems with discrete-valued inputs, and then we have applied these techniques to the specific case of building energy systems. These ideas can serve as the basis for a wide variety of cost-reducing and efficiency-improving online optimization strategies, leading to greater operational flexibility and faster response to changing economic or other external conditions.

7.3 Future Directions

Due to the finite-horizon nature of a thesis, not all of the issues relevant to the topics of study have been addressed in this document. Indeed, there are many important questions left to be answered. To this end, we close by providing some suggestions for future avenues of research. Readers may view this section as the terminal control law of this thesis: it provides a feasible path forward, but careful optimization may be able to find something better.

7.3.1 Mixed-Integer Model Predictive Control

Although we have presented a set of conditions under which continuous-actuator MPC results can be extended to cover discrete actuators, some practical and theoretical challenges remain. In particular, the solution to MIMPC optimization problems could benefit from tailored optimization strategies, the effect of changes to the reference trajectory on tracking MPC could be investigated, and inherent robustness results should be extended to the economic MPC case.

While the suboptimal MPC strategy introduced in Chapter 2 has only trivial optimization requirements to provide closed-loop stability, process performance is likely to improve from better open-loop solutions. Advances in general mixed-integer optimization techniques are helpful, but it is likely that the specific structure of MPC problems allows the application of more specialized strategies. For example, (approximate) dynamic programming is attractive because the computational effort is linear in the horizon N, but application can be complicated by discontinuities in the objective function, which are more likely in the discrete-actuator setting. However, even simple tweaks to existing optimization methods could lead to improvements. For example, the values of discrete inputs early in the horizon are likely to have greater effect on the objective function than later inputs, and thus setting a branching priority for early times could lead to faster bound improvement. Thus, these and other specialized techniques should be explored to improve MIMPC solution times.

As presented in Chapter 2, tracking MPC is formulated in terms of stabilizing the origin, which is equivalent to stabilizing an arbitrary (but fixed) feasible reference trajectory. The chief limitation here is that any changes to the reference trajectory (i.e., setpoint changes) essentially begin a new transient period. Depending both the frequency of updates and how much the cost changes with each update, stability may no longer hold. This presents a challenge for hierarchical implementations of MPC, for example using a top-level economic MPC problem to generate a trajectory that is tracked by a lower-level tracking MPC problem. Even in the nominal case, economic MPC may decide to revise its solution at each timestep, and thus the stability theory of Chapter 2 does not apply to the nominal lower-level system. To overcome this limitation, tracking MPC analysis could be reformulated to explicitly account for changes to the reference trajectory. This case is slightly different than robust MPC approaches because the reference trajectory is generated by a known process that can be modified, in contrast to random disturbances which generally cannot be modified. With suitable modifications to constrain reference trajectory changes, nominal asymptotic stability may still be achievable, which can help guide the development of hierarchical decomposition strategies.

Finally, one of the chief benefits of applying economic MPC is that the system optimizes economics dynamically. In particular, this means the system can respond asymmetrically to disturbances whether they are economically favorable or unfavorable. Unfortunately, there is little theory to describe the effects of these disturbances. While there has been some work on robust and stochastic economic MPC, it is desirable to obtain a similar *inherent* robustness result for economic MPC. The chief difficulty os that discrete actuators coupled with economic optimization often leads to discontinuous optimal value function. As discussed in Allan et al. (2017), not all discontinuities are detrimental to robustness, but continuity is generally necessary near the origin. Because the economic MPC reference trajectory can be essentially anywhere in the state space, it is therefore possible that any discontinuity could end up being in a bad location. Thus, additional analysis is needed to determine a sufficient set of assumptions under which robust equivalents of Theorem 3.15 and Theorem 3.20 hold.

7.3.2 Closed-Loop Scheduling

For the closed-loop scheduling problem, more work is needed to increase the flexibility of the constraints added by economic MPC as well as to provides methods to control the amount of closed-loop revision to the schedule. In addition, the benefits of integrated scheduling and control need to investigated, along with improvements to the integrated formulation.

The use of an exact terminal equality constraint provides an asymptotic bound on closed-loop cost as discussed in Theorem 3.15. However, because the optimizer must terminate exactly on the reference trajectory, the decisions available to the optimizer, especially towards the end of the horizon, can become severely restricted. In addition, if disturbances push the system far from the reference trajectory (perhaps knocking it out of phase), the optimization problem can become infeasible, which requires constraint softening for practical implementation. A better strategy would be to use a larger terminal region coupled with a terminal cost. Thus, the development of more general terminal constraints for scheduling systems is a useful avenue for future research effort. For example, it might be possible to analyze a given scheduling formulation directly and determine constraints based on on model parameters like batch sizes and product demands. Another possibility would be to apply dynamic programming (likely in approximate form) to determine conservative upper bounds on the cost to go. Finally, in the case of periodic references with non-fundamental periods (e.g., a 48 h periodic solution for a system whose parameters are 24 h periodic), it could be helpful to add an option for the optimizer to choose the phase with which it meets the reference trajectory. This flexibility would help in the face of disturbances or when the system's initial condition is far from the reference, but care is needed to ensure that the closed-loop properties from Section 3.3 still hold.

An additional area for exploration would be the extension of rate-of-change penalties within an optimization to revision penalties between sequential optimization problems. Even in the nominal case, MPC can choose to change its predicted input trajectory from one timestep to the next. For traditional control applications, these revisions do not matter, as inputs can be implemented quickly. By contrast, in scheduling problems, the decisions to, e.g., start a particular batch in a particular unit requires a certain amount of preparation and advanced notice. Therefore, if the schedule is revised just before a task is started, then there is a hidden cost that is not known to the optimizer. Of course, if the modification leads to significant cost reduction, then it is worthwhile, but in many cases, these revisions are due to the optimizer chasing very small improvements to the objective function). A possible solution is to consider the extended state $(x, \mathbf{u})$ of the system's current state x and predicted input sequence $\mathbf{u}$ and add a penalty to the optimization problem for choosing $\mathbf{u}^* \neq \mathbf{u}$. An example of such a penalty is $|\mathbf{u} - \mathbf{u}^*|$, but a quick investigation reveals that this term inappropriately penalizes various revisions. For example, delaying a task by one timestep is penalized the same amount as moving a task forward by multiple timestep, which is two times the penalty of simply canceling the batch all together. Thus, additional effort is needed to determine a penalty $\delta(\mathbf{u}, \mathbf{u}^*)$ that correctly

penalizes schedule deviations. As long as $\delta(\,\cdot\,)$ is positive-definite, the closed-loop cost bound from Theorem 3.15 still holds, and the other results likely generalize as well. Adding these penalties to the online closed-loop scheduling problem can reduce the so-called "plant nervousness" associated with making constant revisions to the incumbent schedule.

Finally, for the problem of integrated scheduling and control, costs and benefits need to be investigated in greater detail. In many systems, it is likely that the traditional hierarchy is sufficient based on timescale separation and lack of significant interaction between units. However, for units in continuous operation, explicit consideration of the underlying system dynamics may make certain transitions faster, or it could render particular combinations of transitions infeasible. Thus, for situations with a tangible benefit from integration, more work is needed to determine a tractable integrated formulation. In particular, general methods to identify the key transitions and operating points should be developed so that modeling effort can be focused on those specific items in order to maximize cost improvement. Such techniques would, for example, guide the creation of transition and operating point catalogs for the dynamic-aware scheduling formulation and would prevent unnecessary development and optimization effort on unimportant cases.

7.3.3 Optimization of Building Energy Systems

For the building energy optimization, the key remaining challenges concern practical implementation and additional extensions. Specifically, strategies for obtaining and maintaining models should be developed, and extensions of the airside and waterside decision space should be considered.

For waterside equipment models, steady-state operating models can be obtained from fits of historical or manufacturer's data. For units like cooling towers whose operating depends on ambient conditions, the model parameters can be time-varying as necessary. On a slower timescale, the models can be updated based on recent operating data. However, updating each chiller's model individually leads to multiple different units with only slightly different equipment models. This change creates a nearly-symmetric formulation, which could adversely affect solution times. Thus, when updating equipment models from online operation data, it may be necessary to determine a similarity threshold below which units are treated as identical. In addition, for units with long startup and shutdown periods, it may be necessary to extend the formulation to include these transient dynamic effects. Constraints on ramp rates can be added to prevent rapid changes, but if resource consumption characteristics are very different from steady operation (in particular, if there are spikes in electricity usage that could affect peak demand charges), then these effects may need to be modeled explicitly.

For airside temperature dynamic models, linear state-space models can be obtained by applying system identification techniques. To make the most effective use of passive TES, the two-timescale behavior of temperature evolution

must be captured, which has some ramifications for how much data needs to be used. As an additional complication, it may not be possible to control cooling loads directly, and instead air handlers must be sent temperature setpoints. Therefore, models should be extended so that the inputs are setpoint signals sent to the air handlers, while the outputs are temperature (to enforce comfort constraints) and resource consumption (to determine necessary heating and cooling demand). Initial efforts in this direction are discussed in Patel et al. (2016b), but industrial implementation has not yet been demonstrated.

Finally, an additional source of cost reduction comes from varying the temperatures of various water and air streams. For chilled water supply, suitable extensions have been discussed in Chapter 5, and the findings were that chilled water supply temperature variation is not a significant source of cost reduction. Similar techniques could be applied to the cooling water loop, which would relax the assumption that cooling water supply temperature is a fixed parameter. However, a significant number of additional nonlinearities would have to be addressed, as the aggregate stream temperatures would have to be calculated from the individual unit values. This difficulty combined with the results of chilled water supply temperature optimization indicate it may not be useful to include in the long-horizon scheduling model. Nevertheless, in a static single-periodic optimization, benefits could still be gained from a less computationally costly procedure. Similar temperature considerations also exist for the discharge air temperature on the airside. Unfortunately, benefits are likely to vary from building to building, as the electricity requirements of ventilation depend on the particulars of the HVAC system internals. Therefore, additional study is warranted to see if further cost improvement can be achieved.

Bibliography

T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

A. Afram and F. Janabi-Sharifi. Theory and applications of HVAC control systems—a review of model predictive control (MPC). *Building and Environment*, 72:343–355, 2014.

R. P. Aguilera and D. E. Quevedo. Stability analysis of quadratic MPC with a discrete input alphabet. *IEEE Transactions on Automatic Control*, 58(12):3190–3196, 2013.

M. H. Albadi and E. F. El-Saadany. Demand response in electricity markets: An overview. In *2007 IEEE Power Engineering Society General Meeting*, pages 1–5. IEEE, 2007.

M. Ali, V. Vukovic, M. H. Sahir, and G. Fontanella. Energy analysis of chilled water system configurations using simulation-based optimization. *Energy and Buildings*, 59:111–122, 2013.

D. A. Allan, M. J. Risbeck, and J. B. Rawlings. Stability and robustness of model predictive control with discrete actuators. In *Proceedings of the American Control Conference*, pages 32–37, Boston, MA, 2016.

D. A. Allan, C. N. Bates, M. J. Risbeck, and J. B. Rawlings. On the inherent robustness of optimal and suboptimal nonlinear MPC. *Systems & Control Letters*, 106:68–78, 2017.

R. Amrit, J. B. Rawlings, and D. Angeli. Economic optimization using model predictive control with a terminal cost. *Annual Reviews in Control*, 35(2):178–186, 2011.

J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi—a software framework for nonlinear optimization and optimal control. *Mathematical Programming*, 2018. Accepted.

D. Angeli, R. Amrit, and J. B. Rawlings. On average performance and stability of economic model predictive control. *IEEE Transactions on Automatic Control*, 57(7):1615–1626, 2012.

A. J. Ardakani, F. F. Ardakani, and S. H. Hosseinian. A novel approach for optimal chiller loading using particle swarm optimization. *Energy and Buildings*, 40(12):2177–2187, 2008.

A. Ashouri, S. S. Fux, M. J. Benz, and L. Guzzella. Optimal design and operation of building services using mixed-integer linear programming techniques. *Energy*, 59:365–376, 2013.

M. Avci, M. Erkoc, A. Rahmani, and S. Asfour. Model predictive HVAC load control in buildings using real-time electricity pricing. *Energy and Buildings*, 60:199–209, 2013.

M. Baotić, F. J. Christophersen, and M. Morari. Constrained optimal control of hybrid systems with a linear performance index. *IEEE Transactions on Automatic Control*, 51(12):1903–1919, 2006.

F. A. Bayer, M. A. Müller, and F. Allgöwer. Tube-based robust economic model predictive control. *Journal of Process Control*, 24(8):1237–1246, 2014.

F. A. Bayer, M. Lorenzen, M. A. Müller, and F. Allgöwer. Robust economic model predictive control using stochastic information. *Automatica*, 74:151–161, 2016.

M. Behl, T. X. Nghiem, and R. Mangharam. Green scheduling for energy-efficient operation of multiple chiller plants. *Real-Time Systems Symposium (RTSS), 2012 IEEE 33rd*, pages 195–204, 2012.

P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.

A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3): 407–427, 1999.

S. C. Bengea, A. D. Kelman, F. Borrelli, R. Taylor, and S. Narayanan. Implementation of model predictive control for an HVAC system in a mid-size commercial building. *HVAC&R Research*, 20(1):121–135, 2014.

S. V. Berg and A. Savvides. The theory of maximum kW demand charges for electricity. *Energy Economics*, 5(4): 258–266, 1983.

S. Blair. Editors' choice and best energy/industrial: Stanford Energy System Innovations. *Engineering News-Record*, 2016. URL `http://www.enr.com/articles/39005-editors-choice-best-energyindustrial-stanford-energy-system-innovations`.

M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1):31–45, 1998.

J. E. Braun. A near-optimal control strategy for cool storage systems with dynamic electric rates. *HVAC&R Research*, 13(4):557–580, 2007a.

J. E. Braun. A general control algorithm for cooling towers in cooling plants with electric and/or gas-driven chillers. *HVAC&R Research*, 13(4):581–598, 2007b.

J. E. Braun and G. T. Diderrich. Near-optimal control of cooling towers for chilled-water systems. *ASHRAE Transactions*, 96(CONF-9006117), 1990.

J. Cai, D. Kim, R. Jaramillo, J. E. Braun, and J. Hu. A general multi-agent control approach for building energy system optimization. *Energy and Buildings*, 127:337–351, 2016.

E. F. Camacho, D. R. Ramirez, D. Limon, D. M. de la Peña, and T. Alamo. Model predictive control techniques for hybrid systems. *Annual Reviews in Control*, 34(1):21–31, 2010.

J. A. Candanedo, V. R. Dehkordi, and M. Stylianou. Model-based predictive control of an ice storage device in a building cooling system. *Applied Energy*, 111:1032–1045, 2013.

D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 2, pages 1319–1324. IEEE, 1996.

Y. Chu and F. You. Integration of scheduling and control with online closed-loop implementation: Fast computational strategy and large-scale global optimization algorithm. *Computers & Chemical Engineering*, 47:248–268, 2012.

W. J. Cole, T. F. Edgar, and A. Novoselac. Use of model predictive control to enhance the flexibility of thermal energy storage cooling systems. In *American Control Conference (ACC), 2012*, pages 2788–2793, 2012.

H. Dagdougui, R. Minciardi, A. Ouammi, M. Robba, and R. Sacile. Modeling and optimization of a hybrid system for the energy supply of a "green" building. *Energy Conversion and Management*, 64:351–363, 2012.

S. Di Cairano, W. Heemels, M. Lazar, and A. Bemporad. Stabilizing dynamic controllers for hybrid systems: A hybrid control Lyapunov function approach. *IEEE Transactions on Automatic Control*, 59(10):2629–2643, 2014.

L. S. Dias and M. G. Ierapetritou. Integration of scheduling and control under uncertainties: Review and challenges. *Chemical Engineering Research and Design*, 116, 2016.

M. Diehl, R. Amrit, and J. B. Rawlings. A lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control*, 56(3):703–707, 2011.

A. W. Dowling, R. Kumar, and V. M. Zavala. A multi-scale optimization framework for electricity market participation. *Applied Energy*, 190, 2017.

J. Du, J. Park, I. Harjunkoski, and M. Baldea. A time scale-bridging approach for integrating production scheduling and process control. *Computers & Chemical Engineering*, 79:59–69, 2015.

M. N. ElBsat and M. J. Wenzel. Load and electricity rates prediction for building wide optimization applications. In *4th International High Performance Buildings Conference at Purdue*, West Lafayette, IN, 2016.

M. Ellis, H. Durand, and P. D. Christofides. A tutorial review of economic model predictive control methods. *Journal of Process Control*, 24(8):1156–1178, 2014.

S. Engell and I. Harjunkoski. Optimal operation: Scheduling, advanced control and their integration. *Computers & Chemical Engineering*, 47:121–133, 2012.

G. D. Eppen and R. K. Martin. Determining safety stock in the presence of stochastic lead time and demand. *Management Science*, 34(11):1380–1390, 1988.

P. Falugi and D. Q. Mayne. Tube-based model predictive control for nonlinear systems with unstructured uncertainty. In *Proceedings of 50th IEEE Conference on Decision and Control*, pages 2656–2661, Orlando, Florida, USA, 2011.

L. Fang, G. Clausen, and P. O. Fanger. Impact of temperature and humidity on the perception of indoor air quality. *Indoor Air*, 8(2):80–90, 1998.

J. D. Feng, F. Chuang, F. Borrelli, and F. Bauman. Model predictive control of radiant slab systems with evaporative cooling sources. *Energy and Buildings*, 87:199–210, 2015.

A. Flores-Tlacuahuac and I. E. Grossmann. Simultaneous cyclic scheduling and control of a multiproduct CSTR. *Industrial and Engineering Chemistry Research*, 2006.

R. Z. Freire, G. H. Oliveira, and N. Mendes. Predictive controllers for thermal comfort optimization and energy savings. *Energy and Buildings*, 40(7):1353–1365, 2008.

C. Frenzen, T. Sasao, and J. T. Butler. On the number of segments needed in a piecewise linear approximation. *Journal of Computational and Applied Mathematics*, 234(2):437–446, 2010.

M. Frontczak and P. Wargocki. Literature survey on how different factors influence human comfort in indoor environments. *Building and Environment*, 46(4):922–937, 2011.

C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3): 335–348, 1989.

Z. W. Geem. Solution quality improvement in chiller loading optimization. *Applied Thermal Engineering*, 31(10): 1848–1851, 2011.

R. Goebel, R. G. Sanfelice, and A. R. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, 2009.

R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems*. Princeton University Press, Princeton and Oxford, 2012.

P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, 2006.

L. Grüne and M. Stieler. Asymptotic stability and transient optimality of economic MPC without terminal conditions. *Journal of Process Control*, 24(8):1187–1196, 2014.

D. W. Griffith, V. M. Zavala, and L. T. Biegler. Robustly stable economic NMPC for non-dissipative stage costs. *Journal of Process Control*, 57:116–126, 2017.

G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel. Examples when nonlinear model predictive control is nonrobust. *Automatica*, 40:1729–1738, 2004.

G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel. Nominally robust model predictive control with state constraints. *IEEE Transactions on Automatic Control*, 52(10):1856–1870, 2007.

M. Guignard and S. Kim. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical Programming*, 39(2):215–228, 1987.

D. Gupta and C. T. Maravelias. On deterministic online scheduling: Major considerations, paradoxes and remedies. *Computers & Chemical Engineering*, 94:312–330, 2016.

D. Gupta, C. T. Maravelias, and J. M. Wassick. From rescheduling to online scheduling. *Chemical Engineering Research and Design*, 116, 2016.

Gurobi Optimization, LLC. Gurobi 8.0. 2018. URL `http://www.gurobi.com/`.

S.-I. Gustafsson. Mixed integer linear programming and building retrofits. *Energy and Buildings*, 28(2):191–196, 1998.

M. A. Gutiérrez-Limón, A. Flores-Tlacuahuac, and I. E. Grossmann. Minlp formulation for simultaneous planning, scheduling, and control of short-period single-unit processing systems. *Industrial and Engineering Chemistry Research*, 53(38):14679–14694, 2014.

S. Gyamfi, S. Krumdieck, and T. Urmee. Residential peak electricity demand response—highlights of some behavioural issues. *Renewable and Sustainable Energy Reviews*, 25:71–77, 2013.

A. Hajiah and M. Krarti. Optimal control of building storage systems using both ice storage and thermal mass—part I: Simulation environment. *Energy Conversion and Management*, 64:499–508, 2012.

I. Harjunkoski, R. Nyström, and A. Horch. Integration of scheduling and control—theory or practice? *Computers & Chemical Engineering*, 33(12):1909–1918, 2009.

I. Harjunkoski, C. T. Maravelias, P. Bongers, P. M. Castro, S. Engell, I. E. Grossmann, J. Hooker, C. Méndez, G. Sand, and J. Wassick. Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, 62:161–193, 2014.

M. Heidarinejad, J. Liu, and P. D. Christofides. Economic model predictive control of switched nonlinear systems. *Systems & Control Letters*, 62(1):77–84, 2013.

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

G. P. Henze, B. Biffar, D. Kohn, and M. P. Becker. Optimal design and operation of a thermal storage system for a chilled water plant serving pharmaceutical buildings. *Energy and Buildings*, 40(6):1004–1019, 2008.

R. Huang, E. Harinath, and L. T. Biegler. Lyapunov stability of economically oriented nmpc for cyclic processes. *Journal of Process Control*, 21(4):501–509, 2011.

Z.-P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

G.-Y. Jin, W. Cai, L. Lu, E. L. Lee, and A. Chiang. A simplified modeling of mechanical cooling tower for control and optimization of HVAC systems. *Energy Conversion and Management*, 48(2):355–365, 2007.

M. Jones and M. M. Peet. Solving dynamic programming with supremum terms in the objective and application to optimal battery scheduling for electricity consumers subject to demand charges. *ArXiv e-prints*, 2017.

G. Karer, G. Mušič, I. Škrjanc, and B. Zupančič. Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model. *Nonlinear Analysis: Hybrid Systems*, 2(2):491–509, 2008.

S. S. Keerthi and E. G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2): 265–293, 1988.

C. M. Kellett. A compendium of comparison function results. *Mathematics of Control, Signals, and Systems*, 26(3): 339–374, 2014.

M. Killian and M. Kozek. Ten questions concerning model predictive control for energy efficient buildings. *Building and Environment*, 105:403–412, 2016.

K. J. Kircher and K. Max Zhang. On the lumped capacitance approximation accuracy in RC network building models. *Energy and Buildings*, 108:454–462, 2015.

K. Kobayshi, W. W. Shein, and K. Hiraishi. Large-scale MPC with continuous/discrete-valued inputs: Compensation of quantization errors, stabilization, and its application. *SICE Journal of Control, Measurement, and System Integration*, 7(3):152–158, 2014.

E. Kondili, C. Pantelides, and R. Sargent. A general algorithm for short-term scheduling of batch operations—I. MILP formulation. *Computers & Chemical Engineering*, 17(2):211–227, 1993.

G. M. Kopanos and E. N. Pistikopoulos. Reactive scheduling by a multiparametric programming rolling horizon framework: A case of a network of combined heat and power units. *Industrial and Engineering Chemistry Research*, 53(11):4366–4386, 2014.

R. Kumar, M. J. Wenzel, M. J. Ellis, M. N. ElBsat, K. H. Drees, and V. M. Zavala. A stochastic model predictive control framework for stationary battery systems. *IEEE Transactions on Power Systems*, PP(99):1, 2018.

M. Lazar and W. Heemels. Predictive control of hybrid systems: Input-to-state stability results for sub-optimal solutions. *Automatica*, 45(1):180–185, 2009.

M. Lazar, W. Heemels, S. Weiland, and A. Bemporad. Stabilizing model predictive control of hybrid systems. *IEEE Transactions on Automatic Control*, 51(11):1813–1818, 2006.

M. Lazar, D. M. de la Peña, W. P. M. H. Heemels, and T. Alamo. On input-to-state stability of min-max nonlinear model predictive control. *Systems & Control Letters*, 57:39–48, 2008.

T.-S. Lee, K.-Y. Liao, and W.-C. Lu. Evaluation of the suitability of empirically-based models for predicting energy performance of centrifugal water chillers with variable chilled water flow. *Applied Energy*, 93:583–595, 2012.

W.-S. Lee, Y.-T. Chen, and Y. Kao. Optimal chiller loading by differential evolution algorithm for reducing energy consumption. *Energy and Buildings*, 43(2–3):599–604, 2011.

Z. Li and C. A. Floudas. A comparative theoretical and computational study on robust counterpart optimization: III. improving the quality of robust solutions. *Industrial and Engineering Chemistry Research*, 53(33):13112–13124, 2014.

Z. Li and M. G. Ierapetritou. Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, 32(4-5):715–727, 2008.

L. Lu, W. Cai, Y. C. Soh, L. Xie, and S. Li. HVAC system optimization—condenser water loop. *Energy Conversion and Management*, 45(4):613–630, 2004.

C. A. Méndez, J. Cerdá, I. E. Grossmann, I. Harjunkoski, and M. Fahl. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, 30(6-7):913–946, 2006.

M. A. Müller and L. Grüne. Economic model predictive control without terminal constraints: Optimal periodic operation. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4946–4951, 2015.

M. A. Müller, D. Angeli, and F. Allgöwer. Economic model predictive control with self-tuning terminal cost. *European Journal of Control*, 19(5):408–416, 2013.

J. Ma, J. Qin, T. Salsbury, and P. Xu. Demand reduction in building energy systems based on economic model predictive control. *Chemical Engineering Science*, 67(1):92–100, 2012a.

J. Ma, S. J. Qin, and T. Salsbury. Application of economic MPC to the energy and demand minimization of a commercial building. *Journal of Process Control*, 24(8):1282–1291, 2014.

Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. C. Bengea, and P. Haves. Model predictive control for the operation of building cooling systems. *IEEE Control Systems Technology*, 20(3):796–803, 2012b.

Y. Ma, J. Matuško, and F. Borrelli. Stochastic model predictive control for building HVAC systems: Complexity and conservatism. *IEEE Transactions on Control Systems Technology*, 23(1):101–116, 2015.

C. T. Maravelias. General framework and modeling approach classification for chemical production scheduling. *AIChE Journal*, 58(6):1812–1828, 2012.

D. Mayne. Robust and stochastic model predictive control: Are we going in the right direction? *Annual Reviews in Control*, 41:184–192, 2016.

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

D. Q. Mayne, M. M. Serón, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

L. W. McKenzie. Turnpike theory. *Econometrica*, 44(5):841, 1976.

C. A. Mendez and J. Cerdá. An milp framework for batch reactive scheduling with limited discrete resources. *Computers & Chemical Engineering*, 28(6-7):1059–1068, 2004.

D. I. Mendoza-Serrano and D. J. Chmielewski. Controller and system design for HVAC with thermal energy storage. In *American Control Conference (ACC), 2012*, pages 3669–3674, 2012.

A. Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.

P.-D. Morosan, R. Bourdais, D. Dumur, and J. Buisson. Building temperature regulation using a distributed model predictive control. *Energy and Buildings*, 42(9):1445–1452, 2010.

F. B. Morris, J. E. Braun, and S. J. Treado. Experimental and simulated performance of optimal control of building thermal storage. *ASHRAE Transactions*, 100(1), 1994.

M. A. Muller, D. Angeli, and F. Allgöwer. On necessity and robustness of dissipativity in economic model predictive control. *IEEE Transactions on Automatic Control*, 60(6):1671–1676, 2015.

Y. Nie, L. T. Biegler, C. M. Villa, and J. M. Wassick. Discrete time formulation for the integration of scheduling and dynamic optimization. *Industrial and Engineering Chemistry Research*, 54(16):4303–4315, 2015.

F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45: 15–27, 2012.

G. Pannocchia, J. B. Rawlings, and S. J. Wright. Conditions under which suboptimal nonlinear MPC is inherently robust. *Systems & Control Letters*, 60(9):747–755, 2011.

C. C. Pantelides. Unified frameworks for optimal process planning and scheduling. In *Proceedings on the second conference on foundations of computer aided operations*, pages 253–274, 1994.

N. N. R. Patel, M. J. Risbeck, J. B. Rawlings, M. M. J. Wenzel, and R. D. Turney. Distributed economic model predictive control for large-scale building temperature regulation. In *American Control Conference*, pages 895–900, Boston, MA, 2016a.

N. R. Patel, J. B. Rawlings, M. J. Wenzel, and R. D. Turney. Design and application of distributed economic model predictive control for large-scale building temperature regulation. In *4th International High Performance Buildings Conference at Purdue*, West Lafayette, IN, 2016b.

G. S. Pavlak, G. P. Henze, and V. J. Cushing. Evaluating synergistic effect of optimally controlling commercial building thermal mass portfolios. *Energy*, 84:161–176, 2015.

G. M. Phillips. Algorithms for piecewise straight line approximations. *The Computer Journal*, 11(2):211–212, 1968.

B. Picasso, S. Pancanti, A. Bemporad, and A. Bicchi. Receding–horizon control of LTI systems with quantized inputs. In *Analysis and Design of Hybrid Systems 2003 (ADHS 03): A Proceedings Volume from the IFAC Conference, St. Malo, Brittany, France, 16-18 June 2003*, volume 259, 2003.

B. Picasso, D. Desiderio, and R. Scattolini. Robust stability analysis of nonlinear discrete-time systems with application to MPC. *IEEE Transactions on Automatic Control*, 57(1):185–191, 2012.

K. M. Powell, W. J. Cole, U. F. Ekarika, and T. F. Edgar. Optimal chiller loading in a district cooling system with thermal energy storage. *Energy*, 50:445–453, 2013.

B. Pregelj and S. Gerkšič. Hybrid explicit model predictive control of a nonlinear process approximated with a piecewise affine model. *Journal of Process Control*, 20(7):832–839, 2010.

S. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

D. E. Quevedo, G. C. Goodwin, and J. A. De Doná. Finite constraint set receding horizon quadratic control. *International Journal of Robust and Nonlinear Control*, 14(4):355–377, 2004.

C. V. Rao and J. B. Rawlings. Steady states and constraints in model predictive control. *AIChE Journal*, 45(6):1266–1278, 1999.

J. Rawlings and K. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, 1993.

J. Rawlings, N. Patel, M. Risbeck, C. Maravelias, M. Wenzel, and R. Turney. Economic MPC and real-time decision making with application to large-scale hvac energy systems. *Computers & Chemical Engineering*, 114:89 – 98, 2017a.

J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI, 2009.

J. B. Rawlings and M. J. Risbeck. On the equivalence between statements with epsilon-delta and K-functions, 2015. URL `http://jbrwww.che.wisc.edu/tech-reports/twccc-2015-01.pdf`.

J. B. Rawlings and M. J. Risbeck. Model predictive control with discrete actuators: Theory and application. *Automatica*, 78:258–265, 2017.

J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation and Design*. Nob Hill Publishing, Madison, WI, 2017b.

S. Rippa. Adaptive approximation by piecewise linear polynomials on triangulations of subsets of scattered data. *SIAM Journal on Scientific and Statistical Computing*, 13(5):1123–1141, 1992.

M. J. Risbeck and J. B. Rawlings. MPCTools: Nonlinear model predictive control tools for CasADi (Octave interface), 2018a. URL `https://bitbucket.org/rawlings-group/octave-mpctools`.

M. J. Risbeck and J. B. Rawlings. Economic model predictive control for time-varying cost and peak demand charge optimization. *IEEE Transactions on Automatic Control*, 2018b. Under review.

M. J. Risbeck, C. T. Maravelias, J. B. Rawlings, and R. D. Turney. Cost optimization of combined building heating/cooling equipment via mixed-integer linear programming. In *Proceedings of the American Control Conference*, pages 1689–1694, Chicago, IL, 2015.

M. J. Risbeck, C. T. Maravelias, J. B. Rawlings, and R. D. Turney. Closed-loop scheduling for cost minimization in hvac central plants. In *International High Performance Buildings Conference*, 2016.

M. J. Risbeck, C. T. Maravelias, J. B. Rawlings, and R. D. Turney. A mixed-integer linear programming model for real-time cost optimization of building heating, ventilation, and air conditioning equipment. *Energy and Buildings*, 142:220–235, 2017.

M. J. Risbeck, C. T. Maravelias, and J. B. Rawlings. Closed-loop economic model predictive control for scheduling and control problems. In *Proceedings of the 13th International Symposium on Process Systems Engineering (PSE)*, San Diego, CA, 2018a.

M. J. Risbeck, C. T. Maravelias, J. B. Rawlings, and R. D. Turney. Mixed-integer optimization methods for online scheduling in large-scale HVAC systems. *Optimization Letters*, 2018b. Accepted.

P. Rivotti and E. N. Pistikopoulos. A dynamic programming based approach for explicit model predictive control of hybrid systems. *Computers & Chemical Engineering*, 72:126–144, 2015.

R. T. Rockafellar and R. J. B. Wets. Scenarios and policy agregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

B. Roset, W. Heemels, M. Lazar, and H. Nijmeijer. On robustness of constrained discrete-time systems to state measurement errors. *Automatica*, 44(4):1161–1165, 2008.

S. Sager, H. G. Bock, and G. Reinelt. Direct methods with maximal lower bound for mixed-integer optimal control problems. *Mathematical Programming*, 118(1):109–149, 2009.

S. Sager, H. G. Bock, and M. Diehl. The integer approximation error in mixed-integer optimal control. *Mathematical Programming*, 133(1-2):1–23, 2010.

N. V. Sahinidis. BARON user manual v. 2018.5.9. 2018. URL `http://www.minlp.com/downloads/docs/baronmanual.pdf`.

G. Sand, S. Engell, A. Märkert, R. Schultz, and C. Schulz. Approximation of an ideal online scheduler for a multiproduct batch plant. *Computers & Chemical Engineering*, 24(2-7):361–367, 2000.

H. Scherer, M. Pasamontes, J. Guzmán, J. Álvarez, E. Camponogara, and J. Normey-Rico. Efficient building energy management using distributed model predictive control. *Journal of Process Control*, 24(6):740–749, 2014.

P. O. Scokaert and J. B. Rawlings. Infinite horizon linear quadratic control with constraints. *IFAC Proceedings Volumes*, 29(1):5905–5910, 1996.

P. O. M. Scokaert and J. B. Rawlings. Feasibility issues in linear model predictive control. *AIChE Journal*, 45(8): 1649–1659, 1999.

A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209 (1):63–72, 2011.

H. Shi and F. You. A computational framework and solution algorithms for two-stage adaptive robust scheduling of batch manufacturing processes under uncertainty. *AIChE Journal*, 62(3):687–703, 2016.

D. Sturzenegger, D. Gyalistras, M. Morari, and R. S. Smith. Model predictive climate control of a Swiss office building: Implementation, results, and cost-benefit analysis. *IEEE Transactions on Control Systems Technology*, 24(1):1–12, 2016.

K. Subramanian, C. T. Maravelias, and J. B. Rawlings. A state-space model for chemical production scheduling. *Computers & Chemical Engineering*, 47:97–110, 2012.

K. Subramanian, J. B. Rawlings, and C. T. Maravelias. Economic model predictive control for inventory management in supply chains. *Computers & Chemical Engineering*, 64:71–80, 2014.

M. Sznaier and M. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *26th IEEE Conference on Decision and Control*, pages 761–762. IEEE, 1987.

T. N. Taylor and P. M. Schwarz. The long-run effects of a time-of-use demand charge. *RAND Journal of Economics*, 21 (3):431, 1990.

J. Thomas. Analytical non-linear model predictive control for hybrid systems with discrete inputs only. *IET Control Theory and Applications*, 6(8):1080–1088, 2012.

A. Toriello and J. P. Vielma. Fitting piecewise linear continuous functions. *European Journal of Operational Research*, 219(1):86–95, 2012.

C. R. Touretzky and M. Baldea. Integrating scheduling and control for economic MPC of buildings with energy storage. *Journal of Process Control*, 24(8):1292–1300, 2014.

C. R. Touretzky and M. Baldea. A hierarchical scheduling and control strategy for thermal energy storage systems. *Energy and Buildings*, 110:94–107, 2016.

C. R. Touretzky, I. Harjunkoski, and M. Baldea. Dynamic models and fault diagnosis-based triggers for closed-loop scheduling. *AIChE Journal*, 63(6):1959–1973, 2017.

M. Trifkovic, W. A. Marvin, P. Daoutidis, and M. Sheikhzadeh. Dynamic real-time optimization and control of a hybrid energy system. *AIChE Journal*, 60(7):2546–2556, 2014.

US Department of Energy. Buildings energy data book, 2011. URL `http://buildingsdatabook.eren.doe.gov/`.

S. Velez and C. T. Maravelias. Reformulations and branching methods for mixed-integer programming chemical production scheduling models. *Industrial and Engineering Chemistry Research*, 52(10):3832–3841, 2013.

J. P. Vielma and G. L. Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, 128(1-2):49–72, 2009.

J. P. Vielma, S. Ahmed, and G. Nemhauser. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research*, 58(2):303–315, 2010.

J. Villadsen and M. L. Michelsen. *Solution of Differential Equation Models by Polynomial Approximation.* Prentice-Hall, Englewood Cliffs New Jersey, 1978.

J. P. Vin and M. G. Ierapetritou. A new approach for efficient rescheduling of multiproduct batch plants. *Industrial and Engineering Chemistry Research*, 39(11):4228–4238, 2000.

A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

M. J. Wenzel, R. D. Turney, and K. H. Drees. Model predictive control for central plant optimization with thermal energy storage. In *3rd International High Performance Buildings Conference at Purdue*, West Lafayette, IN, 2014.

M. J. Wenzel, R. D. Turney, and K. H. Drees. Autonomous optimization and control for central plants with energy storage. In *4th International High Performance Buildings Conference at Purdue*, West Lafayette, IN, 2016.

L. A. Wolsey. *Integer programming*. Wiley, 1998.

G. Yi and G. V. Reklaitis. Adaptive model predictive inventory controller for multiproduct batch plant. *AIChE Journal*, 61(6):1867–1880, 2015.

F. You and I. E. Grossmann. Design of responsive supply chains under demand uncertainty. *Computers & Chemical Engineering*, 32(12):3090–3111, 2008.

S. Yu, M. Reble, H. Chen, and F. Allgöwer. Inherent robustness properties of quasi-infinite horizon nonlinear model predictive control. *Automatica*, 50(9):2269–2280, 2014.

M. Zanon, S. Gros, and M. Diehl. A Lyapunov function for periodic economic optimizing model predictive control. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 5107–5112, 2013.

M. Zanon, S. Gros, and M. Diehl. A periodic tracking MPC that is locally equivalent to periodic economic MPC. *IFAC–Papers Online*, 50(1):10711–10716, 2017a.

M. Zanon, L. Grune, and M. Diehl. Periodic optimal control, dissipativity and MPC. *IEEE Transactions on Automatic Control*, 62(6), 2017b.

V. M. Zavala. A multiobjective optimization perspective on the stability of economic MPC. *IFAC–Papers Online*, 48(8): 974–980, 2015.

V. M. Zavala, E. M. Constantinescu, T. Krause, and M. Anitescu. On-line economic optimization of energy systems using weather forecast information. *Journal of Process Control*, 19(10):1725–1736, 2009.

D. Zhang, N. Shah, and L. G. Papageorgiou. Efficient energy consumption and operation management in a smart building with microgrid. *Energy Conversion and Management*, 74:209–222, 2013.

J. Zhuge and M. G. Ierapetritou. Integration of scheduling and control with closed loop implementation. *Industrial and Engineering Chemistry Research*, 51(25):8550–8565, 2012.