1-1-2012

Green Scheduling of Control Systems

Truong Xuan Nghiem
University of Pennsylvania, truong.nghiem@gmail.com

Green Scheduling of Control Systems

Abstract
Electricity usage under peak load conditions can cause issues such as reduced power quality and power outages. For this reason, commercial electricity customers are often subject to demand-based pricing, which charges very high prices for peak electricity demand. Consequently, reducing peaks in electricity demand is desirable for both economic and reliability reasons. In this thesis, we investigate the peak demand reduction problem from the perspective of safe scheduling of control systems under resource constraint. To this end, we propose Green Scheduling as an approach to schedule multiple interacting control systems within a constrained peak demand envelope while ensuring that safety and operational conditions are facilitated. The peak demand envelope is formulated as a constraint on the number of binary control inputs that can be activated simultaneously. Using two different approaches, we establish a range of sufficient and necessary schedulability conditions for various classes of affine dynamical systems. The schedulability analysis methods are shown to be scalable for large-scale systems consisting of up to 1000 subsystems. We then develop several scheduling algorithms for the Green Scheduling problem. First, we develop a periodic scheduling synthesis method, which is simple and scalable in computation but does not take into account the influence of disturbances. We then improve the method to be robust to small disturbances while preserving the simplicity and scalability of periodic scheduling. However the improved algorithm usually result in fast switching of the control inputs. Therefore, event-triggered and self-triggered techniques are used to alleviate this issue. Next, using a feedback control approach based on attracting sets and robust control Lyapunov functions, we develop event-triggered and self-triggered scheduling algorithms that can handle large disturbances affecting the system. These algorithms can also exploit prediction of the disturbances to improve their performance. Finally, a scheduling method for discrete-time systems is developed based on backward reachability analysis. The effectiveness of the proposed approach is demonstrated by an application to scheduling of radiant heating and cooling systems in buildings. Green Scheduling is able to significantly reduce the peak electricity demand and the total electricity consumption of the radiant systems, while maintaining thermal comfort for occupants.

Degree Type
Dissertation

Degree Name
Doctor of Philosophy (PhD)

Graduate Group
Electrical & Systems Engineering

First Advisor
George J. Pappas

Second Advisor
Rahul Mangharam

Keywords
control system, demand management, energy efficient, green scheduling, resource constraint

Subject Categories
Electrical and Electronics

GREEN SCHEDULING OF CONTROL SYSTEMS

Truong X. Nghiem

A DISSERTATION

in

Electrical and Systems Engineering

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2012

Supervisor of Dissertation

George J. Pappas

Professor, Electrical and Systems Engineering

Co-Supervisor of Dissertation

Rahul Mangharam

Assistant Professor, Electrical and Systems Engineering

Graduate Group Chairperson

Saswati Sarkar, Professor, Electrical and Systems Engineering

Dissertation Committee:

Rajeev Alur, Professor, Computer and Information Science

George J. Pappas, Professor, Electrical and Systems Engineering

Rahul Mangharam, Assistant Professor, Electrical and Systems Engineering

Francesco Borrelli, Associate Professor, Mechanical Engineering, University of California at Berkeley

GREEN SCHEDULING OF CONTROL SYSTEMS

Dedicated to my parents and my wife

ACKNOWLEDGEMENT

I owe my deepest gratitude to two important persons without whom I would not have finished this dissertation. The guidance, support and encouragement which I received from my advisor, George J. Pappas, is the most important reason that made the work presented here possible. He brought me the opportunity to participate in the DoE Energy Efficient Buildings project, from which I was motivated to start this thesis. Not only that, his foundational work on approximate simulation and bisimulation relations of dynamical systems inspired many results developed in this work. I am also indebted to my co-advisor, Rahul Mangharam, for his support and fruitful research collaboration. Without the help I received from him and his laboratory, mLab, I would not have produced the amount of work presented here in less than two years.

My gratitude extends to my thesis committee members Rajeev Alur and Francesco Borrelli for their comments and suggestions. My discussion with Francesco Borrelli at the 2011 IEEE Conference on Decision and Control, though very short, guided me in the right direction to find an application for the theory developed in my thesis work.

I would like to acknowledge my collaborators and friends Madhur Behl and Willy Bernal for their contributions and help in the publications that form a large portion of this thesis. Over the years at Penn, I had the opportunity to co-author several papers which do not appear in this thesis. I would like to express my appreciation to my co-authors Antoine Girard, Georgios Fainekos, Sriram Sankaranarayanan, Franjo Ivancic, Aarti Gupta, Zheng Li, Pei-Chi Huang and Aloysius Mok. My thanks also go to Bin Yan, who taught me a lot and answered many of my questions about HVAC systems in buildings.

I would like to thank my friends Linh Phan and Nha Nguyen for their support and encouragement when I felt depressed or bored. The financial and legal support from the Vietnam Education Foundation (VEF) gave me the opportunity to come and study in the United States. My grateful appreciation goes to the VEF and its staff.

Last but not least, I would like to thank my parents and my younger sister for their unconditional love and support for me during the time I was studying abroad. Finally, my greatest thankfulness goes to my wonderful wife whose unconditional love, support, understanding and positive attitude gave me the strength and determination to complete this work.

ABSTRACT

GREEN SCHEDULING OF CONTROL SYSTEMS

Truong X. Nghiem

George J. Pappas

Rahul Mangharam

Electricity usage under peak load conditions can cause issues such as reduced power quality and power outages. For this reason, commercial electricity customers are often subject to demand-based pricing, which charges very high prices for peak electricity demand. Consequently, reducing peaks in electricity demand is desirable for both economic and reliability reasons. In this thesis, we investigate the peak demand reduction problem from the perspective of safe scheduling of control systems under resource constraint. To this end, we propose Green Scheduling as an approach to schedule multiple interacting control systems within a constrained peak demand envelope while ensuring that safety and operational conditions are facilitated. The peak demand envelope is formulated as a constraint on the number of binary control inputs that can be activated simultaneously. Using two different approaches, we establish a range of sufficient and necessary schedulability conditions for various classes of affine dynamical systems. The schedulability analysis methods are shown to be scalable for large-scale systems consisting of up to 1000 subsystems. We then develop several scheduling algorithms for the Green Scheduling problem. First, we develop a periodic scheduling synthesis method, which is simple and scalable in computation but does not take into account the influence of disturbances. We then improve the method to be robust to small disturbances while preserving the simplicity and scalability of periodic scheduling. However the improved algorithm usually result in fast switching of the control inputs. Therefore, event-triggered and self-triggered techniques are used to alleviate this issue. Next, using a feedback control approach based on attracting sets and robust control Lyapunov functions,

we develop event-triggered and self-triggered scheduling algorithms that can handle large disturbances affecting the system. These algorithms can also exploit prediction of the disturbances to improve their performance. Finally, a scheduling method for discrete-time systems is developed based on backward reachability analysis. The effectiveness of the proposed approach is demonstrated by an application to scheduling of radiant heating and cooling systems in buildings. Green Scheduling is able to significantly reduce the peak electricity demand and the total electricity consumption of the radiant systems, while maintaining thermal comfort for occupants.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

Chapter 1

Introduction

1.1. Motivation

Balancing the energy generation and utilization in an electric grid is essential for its efficient operation. A fundamental law of electric grids is the balance between the power generated (the *supply*) and the power consumed (the *demand*) modulo power loss in transmission. Ideally, if the demand were constant, it would have been easy for the utilities to satisfy this law by generating the exact amount of electricity needed. However, in reality, electricity demand always fluctuates throughout the day, with multiple high peaks during the *on-peak periods* and low valleys during the *off-peak periods*. Take the campus of the University of Pennsylvania for example. Figure 1.1 on the following page displays the hourly electricity demand of the campus during the week from Sunday, July 10^{th} to Saturday, July 16^{th} in 2011. Evidently, its electricity demand fluctuated significantly between the weekdays and the weekends, and during a day between the on-peak hours and the off-peak hours. In any day, the demand attained its peak values during the late morning and afternoon hours (on-peak) and its bottom values at night (off-peak). In that week, the maximum demand was 97.15 MW while the minimum demand was only 66.49 MW.

Failing to adjust the electricity generation to the demand can result in power outages or reduced power quality, which cost businesses in the United States an estimated average of about $100 billion every year (U.S. Department of Energy, 2008). On the other hand, accommodating peak demands is very expensive for the utilities because inefficient, high-cost peaker plants (also known as backup plants) must be brought online, not to mention the cost to build those sporadically used plants. For example, Braithwait and Eakin (2002) estimated that by reducing the peak demand by only 5%, the electricity price during the California

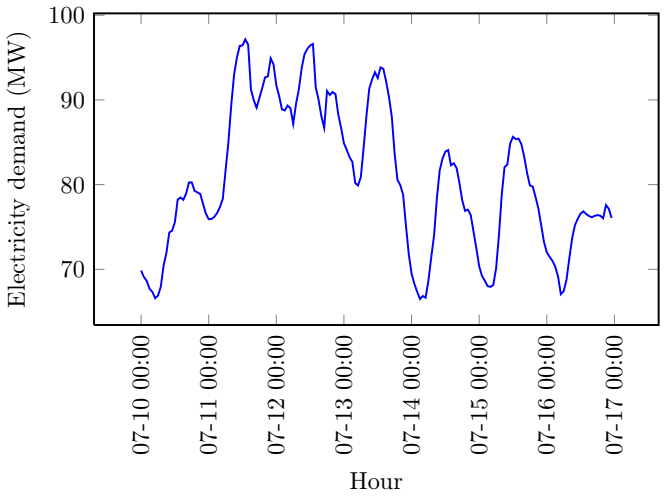


Figure 1.1: Hourly total electricity demand of the University of Pennsylvania for the week from 2011-07-10 (Sunday) to 2011-07-16 (Saturday). The demand fluctuated between the minimum of $66.49\,\mathrm{MW}$ and the maximum of $97.15\,\mathrm{MW}$. In any day, the demand attained its peak values during the late morning and afternoon hours (on-peak) and its bottom values during the night (off-peak).

electricity crisis in 2000–2001 could have been reduced by 50%. Therefore, the ability to control electricity demand so as to reduce the peak demand is always desirable for both economic and reliability reasons.

Various technical and mostly economical methods have been used to control the peak demand, with the most widely adopted being *demand-based pricing* (Albadi and El-Saadany, 2007; Motegi et al., 2007). In a demand-based pricing policy, a large commercial electricity customer is charged not only for the amount of electricity it has consumed but also for its maximum demand over the billing cycle. Specifically, the electricity cost for a billing period (often a month) in a basic demand-based tariff consists of three parts:

- *Basic charge* is the fixed meter charge;
- *Energy charge*, or *usage charge*, is the charge for the amount of energy E used by the customer; and
- *Demand charge* is the charge for the maximum electricity demand $D_{\max}$ of the customer during the on-peak hours of the billing period.

The total electricity bill is then calculated as $C_{\text{basic}} + p_{\text{usage}}E + p_{\text{demand}}D_{\max}$ where C_{basic} is the fixed basic charge, p_{usage} and p_{demand} are the unit prices of the usage charge and the demand charge respectively. This pricing policy incentivizes customers to lower their energy usage under peak load conditions by raising the unit price of the demand charge very high, up to 240 times in some cases (TRFund Study, 2007) and even more. The demand charge portion can account for about 40% of the total electricity bill, depending on how large the peak demand is (McQuiston et al., 2005). Therefore, peaks in electricity demand are inefficient and expensive for both suppliers and customers.

To reduce the peak demand, it is essential to identify their causes. Peak electricity demands are often caused by unnecessary appliances (e.g., running a washing machine during on-peak hours) and by devices consuming more energy than necessary (e.g., setting too low the temperature setpoint of an air conditioner), which can be reduced by the *demand shifting* and *demand limiting* strategies (Motegi et al., 2007). Demand limiting sheds electricity loads when a demand limit is reached, for example by changing setpoints or turning off devices. Demand shifting changes the time that electricity is used, for example by running appliances during off-peak hours or by means of energy storage.

In large-scale systems, such as campuses or a large group of residential customers, peak demands are also caused by the uncoordinated operation of electric equipment. As a simple example, a spike in the demand can be caused by a large number of air conditioners turning on at about the same time. In this case, *coordinating the operation of the devices is necessary to reduce their peak demand*. However, *the coordinated operation must also maintain certain operational specifications* such as thermal comfort for occupants inside buildings or physical constraints of the devices. In other words, this is a **safe control problem where multiple systems are coordinated to reduce their aggregated power demand (a global objective) while each of them satisfies certain safety or operational constraints (local objectives)**. Moreover, their behaviors can be coupled (i.e., the behavior of one system can influence that of another system) and can be subject to disturbances (e.g., the

weather, the number of occupants). In this dissertation, we propose *Green Scheduling* as an approach to solve this problem: **instead of trying to minimize the aggregated peak demand, we set a maximum value on it and schedule the systems within this constrained peak demand envelope while ensuring that safety and operational conditions are facilitated**.

1.1.1. Direct Load Control and Thermostatically Controlled Loads

The idea of coordinating multiple systems to reduce their peak demand is not new, especially in the context of demand response with direct load control and thermostatically controlled loads. Direct load control (DLC) refers to the practice that a utility or system operator can remotely shut down or cycle a customer's electrical equipment on short notice to curtail the peak demand (Chen et al., 1995; Albadi and El-Saadany, 2007). Thermostatically controlled loads (TCL) are equipment such as refrigerators, air conditioners, and electric heaters. Much like batteries which can store chemical energy, they are capable of storing thermal energy and are therefore particularly suitable for DLC because they can be switched off and on without any significant immediate effect on their behavior or thermal comfort.

The scheduling problem for DLC has been studied extensively in the literature and various methods have been proposed. In most cases, the on-off scheduling problem was formulated in the Dynamic Programming (DP) framework (Cohen and Wang, 1988; Hsu and Su, 1991; Lee et al., 2007; Ramanathan and Vittal, 2008). Multi-pass DP was used in (Wei and Chen, 1995) while fuzzy DP was used in (Bhattacharyya and Crow, 1996; Yang and Huang, 1999) to solve the dynamic program of the DLC scheduling problem. Several other methods have also been proposed, such as an iterative deepening genetic algorithm in (Yao et al., 2005) and a scheduling algorithm based on queuing system model in (Lee et al., 2011). Aggregation and control of a large population of TCL have been investigated recently in (Kundu et al., 2011; Koch et al., 2011; Molina-Garcia et al., 2011; Mathieu and Callaway, 2012).

Many of these studies have focused on reducing cost and the peak loads. However, DLC of

thermal equipment such as air-conditioners can adversely impact customer thermal comfort and cause significant discomfort. The DLC scheduling problem with thermal comfort constraints has been investigated in recent research (Chu et al., 2005; Chu and Jong, 2008) where thermal comfort level was directly formulated in the scheduling problem, for which techniques such as DP and fuzzy control were then used to solve. Most of the aforementioned studies assumed simple load models with no interactions between the loads. Furthermore, disturbances such as ambient air temperature and heat gains were often ignored.

The Green Scheduling problem studied in this dissertation is motivated by the peak demand reduction problem as well as the scheduling problem for DLC and TCL. Unlike the above approaches, we focus on dynamical systems that are interdependent and subject to disturbances. We will develop a wide range of scheduling analysis and synthesis methods, each of which is suitable for different types of systems at different scales (from a few to hundreds of subsystems).

1.2. Running Examples

To better illustrate and define the Green Scheduling problem, we introduce two running examples that will be used throughout the dissertation.

1.2.1. Thermostatically Controlled Loads (Room-Heater)

We have mentioned in Section 1.1.1 the motivating example of thermostatically controlled loads (TCL) and direct load control (DLC). To illustrate the theory developed in this dissertation, we will use a running example of a system of TCL, namely a room-heater example. Consider $n > 1$ rooms. These rooms are not necessarily in a same building, for instance they can be rooms in residential houses in a neighborhood. Each room is heated by a heater that can be switched on, when it provides a constant heat input rate to the room, and switched off, when it consumes no energy and provides no heat input. It is assumed that these heaters can be remotely controlled by a centralized scheduler, as in DLC.

We will call the heater in room i ($i = 1, \ldots, n$) heater i. Let $T_i \in \mathbb{R}$ denote the air temperature (°C) of room i and $P_i > 0$ be the constant heat input rate (kW) of heater i when it is switched on. The control input to heater i is its on/off state, denoted by a binary control variable $u_i \in \{0, 1\}$ where $u_i = 0$ corresponds to the off state and $u_i = 1$ the on state of the heater. Thermal comfort specifications require that T_i should be between a lower temperature threshold l_i and an upper temperature threshold $h_i > l_i$, i.e., T_i should be bounded in the range $[l_i, h_i]$. For room i, let $T_{a,i}$ be its ambient air temperature (°C) and $Q_{g,i}$ its internal heat gain (kW) from its occupants and equipment such as lights and computers. It is reasonable to assume that the ambient air temperatures are the same for all rooms, i.e., $T_{a,i} = T_a \ \forall i$, when they are in the same area. Then the law of conservation of energy gives us the following heat balance differential equation for room i, for $t \geq 0$,

$$C_i \dot{T}_i(t) = K_i \left(T_a(t) - T_i(t)\right) + \sum_{j \neq i} K_{ij} \left(T_j(t) - T_i(t)\right) + Q_{g,i}(t) + P_i u_i(t), \tag{1.1}$$

in which $\dot{T}_i(t)$ denotes the time derivative of $T_i(t)$, $C_i > 0$ is the thermal capacity of the room (kJ/K), $K_i > 0$ is the thermal conductance between the ambient air and the room (kW/K), and $K_{ij} \geq 0$ is the thermal conductance between room i and room $j \neq i$ (kW/K). The parameters K_{ij} model the heat transfers between adjacent rooms due to their temperature difference.

Define the state vector $x = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$ and the binary control vector $u = [u_1, \ldots, u_n]^T \in \{0, 1\}^n$. We consider T_a and $Q_{g,i}$, $i = 1, \ldots, n$, as disturbances to the system and define the ambient disturbance variable $d_a := T_a$ and the internal heat gain disturbance vector $d_g := [Q_{g,1}, \ldots, Q_{g,n}]^T \in \mathbb{R}^n$. The differential equations (1.1) for all the rooms can be collected in a state-space dynamical model of the system:

$$\dot{x}(t) = Ax(t) + Bu(t) + W_g d_g(t) + W_a d_a(t) = Ax(t) + Bu(t) + Wd(t) \tag{1.2}$$

where

$$A_{i,i} = -\frac{K_i + \sum_{j \neq i} K_{ij}}{C_i}, \quad A_{i,j} = \frac{K_{ij}}{C_i} \qquad \text{for } i, j \in \{1, \ldots, n\},\ j \neq i \tag{1.3}$$

$$B = \operatorname{diag}\left(\frac{P_1}{C_1}, \ldots, \frac{P_n}{C_n}\right), \quad W_g = \operatorname{diag}\left(\frac{1}{C_1}, \ldots, \frac{1}{C_n}\right) \tag{1.4}$$

$$W_a = \left[\frac{K_1}{C_1}, \ldots, \frac{K_n}{C_n}\right]^T, \quad d = \begin{bmatrix} d_g \\ d_a \end{bmatrix}, \quad W = [W_g, W_a]\,. \tag{1.5}$$

The notation $A_{i,j}$ denotes the i–j element of matrix A and $\operatorname{diag}(a_1, \ldots a_n)$ denotes the diagonal matrix with diagonal entries $a_1, \ldots, a_n$. Note that the state matrix A is always Hurwitz, i.e., all eigenvalues of A have strictly negative real parts, because it is a strictly diagonally dominant matrix with negative diagonal entries (Horn and Johnson, 1990). Therefore the room-heater system is always stable.

We will consider a small-scale room-heater system with 6 rooms and 6 heaters ($n = 6$) whose parameter values are summarized in Table 1.1 on the next page. For large-scale example systems, where n is up to 1000, we generated the parameter values randomly as follows. For each room i, its thermal capacity C_i is from 2000 kJ/K to 3000 kJ/K and its thermal conductance K_i is in the range $[0.2, 0.3]$ (kW/K). The thermal capacity of a room is an indicator of its size, so a greater value of C_i corresponds to a larger room. The heaters' input rates P_i (kW) were chosen based on the size of the room: $P_i = 6$ kW if $C_i \leq 2300$ kJ/K, $P_i = 8$ kW if $C_i > 2700$ kJ/K, and $P_i = 7$ kW otherwise. Since n is large, we randomly assigned rooms which can thermally interact with each other and the value of K_{ij} were chosen from 0 kW/K to 0.3 kW/K, with the value 0 implying that the rooms do not interact.

Green Scheduling Problem For thermal comfort of the occupants, the room temperatures are required to be kept between $l_i = 20\,^\circ$C and $h_i = 24\,^\circ$C, for all $i = 1, \ldots, n$. The ambient air temperature T_a and the heat gains $Q_{g,i}$ are assumed to be unknown but bounded, that is $T_a \in [\underline{T}_a, \overline{T}_a]$ and $Q_{g,i} \in [\underline{Q}_{g,i}, \overline{Q}_{g,i}]$ for $i = 1, \ldots, n$, where $\underline{T}_a$, $\overline{T}_a$, $\underline{Q}_{g,i}$, and $\overline{Q}_{g,i}$ are given. In practice, each heater is typically controlled by a thermostat with a simple

Table 1.1: Parameter values of the small-scale room-heater running example in Section 1.2.1.

(a) Thermal capacitance of rooms and power ratings of heaters.

	Room 1	Room 2	Room 3	Room 4	Room 5	Room 6
C_i (kJ/K)	2927	2679	2074	2070	2011	2227
P_i (kW)	8	7	6	6	6	6

(b) Thermal conductance between rooms and ambient air ($K_i = K_{ii}$) and between adjacent rooms (K_{ij}).

$$K = \begin{array}{c} i/j \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0.2634 & 0.1590 & 0 & 0 & 0 & 0.2970 \\ 0.1590 & 0.2384 & 0.1380 & 0 & 0 & 0 \\ 0 & 0.1380 & 0.1950 & 0.2490 & 0 & 0 \\ 0 & 0 & 0.2490 & 0.1904 & 0.1440 & 0 \\ 0 & 0 & 0 & 0.1440 & 0.1810 & 0.2190 \\ 0.2970 & 0 & 0 & 0 & 0.2190 & 0.1804 \end{pmatrix} \end{array}$$

two-position control rule: the room temperature T_i is monitored, the heater is switched on whenever T_i is below l_i and switched off whenever T_i is above h_i. This uncoordinated operation of the heaters usually results in temporally correlated spikes in their aggregated electricity demand, as we discussed in Section 1.1. The goal of green scheduling for the room-heater system is to design a scheduling strategy $u(\cdot)$ so that the room temperatures T_i are driven to and maintained between l_i and h_i, for all $i = 1, \ldots, n$ and for any admissible disturbance signal $d(\cdot)$. In addition, to reduce the peak electricity demand of the heaters, it is required that $\sum_{i=1}^{n} u_i(t) \leq k$ at all time $t \geq 0$, where $k \in \{0, \ldots, n\}$ is a given peak constraint.

1.2.2. Two-Body Mass-Spring-Damper System

The simple mass-spring-damper system can be found in virtually every introductory course and textbook on dynamical systems and controls (Franklin et al., 1998; Dorf and Bishop, 2008). In this dissertation, a slightly more complex version of this system will be used as a running example to illustrate the various theoretical results. Consider a two-body mass-spring-damper system depicted in Figure 1.2 on the following page. It consists of two

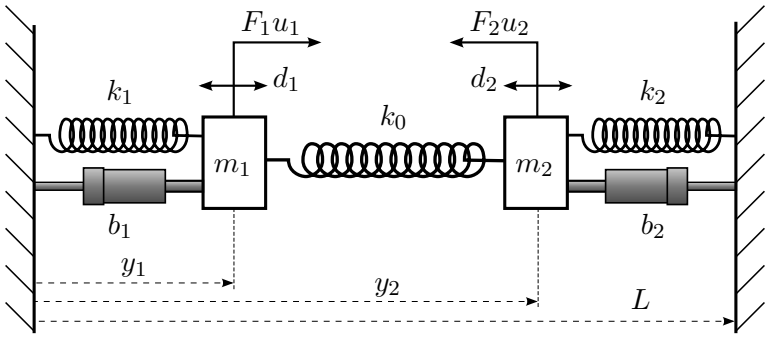


Figure 1.2: The two-body mass-spring-damper running example: two bodies of mass are connected to two walls and to each other by three linear springs and two dampers as illustrated in the figure; a force F_i can be applied to each mass m_i, depending on the binary control input u_i; an unknown but constrained disturbance force d_i is also applied to each mass m_i.

Table 1.2: Parameters of the two-body mass-spring-damper running example.

Parameter	Symbol	Dimension
Mass	m_i	kg
Spring constant	k_i	$\mathrm{kg/s^2}$
Damping coefficient	b_i	kg/s
Control force	F_i	$\mathrm{kg\,m/s^2}$
Maximal disturbance force	$d_{\max}$	$\mathrm{kg\,m/s^2}$
Distance between walls	L	m

bodies of mass m_1 and m_2 which are connected to two walls and to each other by three linear springs and two dampers. A constant force $F_i > 0$ can be applied to each mass m_i, $i = 1, 2$, depending on the binary control input $u_i \in \{0, 1\}$: when $u_i = 0$ there is no force, when $u_i = 1$ the force F_i is applied. There is also an unknown but constrained disturbance force d_i to be applied to each mass m_i. It is assumed that d_i is bounded by $|d_i| \leq d_{\max}$ where $d_{\max} > 0$ is a given maximal disturbance force. The distance between the walls is L and the positions y_1 and y_2 of the masses are measured from the left wall as illustrated in the figure. The parameters of the system and their dimensions are summarized in Table 1.2.

From the theory of classical mechanics, specifically the behaviors of linear springs and

dampers and the Newton's second law, the dynamics of each mass m_i can be derived as

$$m_1\ddot{y}_1 = -k_1 y_1 + k_0(y_2 - y_1) - b_1\dot{y}_1 + F_1 u_1 + d_1$$

and

$$m_2\ddot{y}_2 = k_2(L - y_2) - k_0(y_2 - y_1) - b_2\dot{y}_2 - F_2 u_2 + d_2.$$

Let v_1 and v_2 denote the velocities of the two masses, that is $v_1 := \dot{y}_1$ and $v_2 := \dot{y}_2$. The dynamics of the masses can be rewritten as

$$\dot{v}_1 = -\frac{k_0 + k_1}{m_1} y_1 - \frac{b_1}{m_1} v_1 \frac{k_0}{m_1} y_2 + \frac{F_1}{m_1} u_1 + \frac{1}{m_1} d_1$$

and

$$\dot{v}_2 = -\frac{k_0 + k_2}{m_2} y_2 + \frac{k_0}{m_2} y_1 - \frac{b_2}{m_2} v_2 + \frac{k_2}{m_2} L - \frac{F_2}{m_2} u_2 + \frac{1}{m_2} d_2.$$

By defining the state vector $x = [y_1, v_1, y_2, v_2]^T$, the control input vector $u = [u_1, u_2]^T$ and the disturbance vector $d = [d_1, d_2]^T$, we can arrange the differential equations of the system in a state-space form as $\dot{x}(t) = Ax(t) + B_0 + Bu(t) + Wd(t)$ in which

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_0+k_1}{m_1} & -\frac{b_1}{m_1} & \frac{k_0}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_0}{m_2} & 0 & -\frac{k_0+k_2}{m_2} & -\frac{b_2}{m_2} \end{bmatrix}, B_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_2 L}{m_2} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ \frac{F_1}{m_1} & 0 \\ 0 & 0 \\ 0 & -\frac{F_2}{m_2} \end{bmatrix}, W = \begin{bmatrix} 0 & 0 \\ \frac{1}{m_1} & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_2} \end{bmatrix}. \quad (1.6)$$

Note that the system is stable, hence matrix A is Hurwitz. With no control ($u = 0$) and no disturbances ($d = 0$), the system rests at the equilibrium $x^\star = -A^{-1}B_0 = [y_1^\star, 0, y_2^\star, 0]^T$ where

$$y_1^\star = \frac{k_0 k_2}{k_0 k_2 + (k_0 + k_2)k_1} L, \quad y_2^\star = \frac{(k_0 + k_1)k_2}{k_0 k_2 + (k_0 + k_2)k_1} L. \quad (1.7)$$

Table 1.3 on the next page summarizes the variables of the system. Note that, as in most academic mass-spring-damper examples, for simplicity we do not consider the width of the masses nor collision between them. The masses are allowed to penetrate each other or the

Table 1.3: Variables of the two-body mass-spring-damper running example.

Variable	Symbol	Dimension
Position of mass	y_i	m
Binary control input	u_i	–
Disturbance force	d_i	$\mathrm{kg\,m/s^2}$
Velocity of mass	v_i	m/s

walls. However this did not happen in all instances of the example in this dissertation, thus the simplifying assumption can be justified.

The parameter values that will be used are

$$m_1 = 1, m_2 = 1.2, k_0 = k_1 = k_2 = 0.6, b_1 = b_2 = 0.8, F_1 = F_2 = 1, L = 2, d_{\max} = 0.1. \quad (1.8)$$

Green Scheduling Problem Let $\texttt{Safe} \subset [0, L] \times [0, L]$ be a set of desired positions of the two masses. The goal of the green scheduling problem for this system is to design a control signal $u(\cdot)$ or a control strategy so that the positions $y = [y_1, y_2]^T$ are driven to and maintained indefinitely inside `Safe`, from any initial state $x(0)$ and with any admissible disturbance signal $d(\cdot)$. In addition, the control input is subject to the constraint $\|u(t)\|_1 = u_1(t) + u_2(t) \leq 1$ for all $t \geq 0$, i.e., at any time, at most one of the control forces (F_1 or F_2) can be applied to the masses.

Compared to the room-heater system, the mass-spring-damper system has different characteristics for illustration of the theoretical results developed in the dissertation:

- It has second order dynamics, which include damping (oscillation). For example, under no control ($u = [0, 0]^T$) and no disturbances, the position trajectories of the masses starting from $x(0) = [0, 0, L, 0]^T$ have oscillations as depicted in Figure 1.3 on the following page. This feature of the system is interesting because second-order dynamics are representative of higher-order dynamics that exhibit oscillations and overshoot.

- The safe set `Safe` is defined for the positions y_1 and y_2, not for the state x. If we

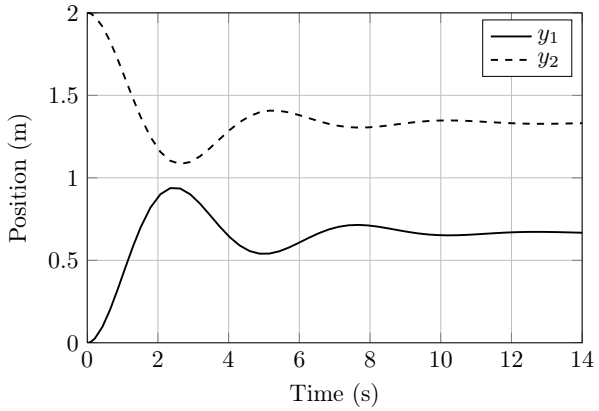


Figure 1.3: Illustration of damping (oscillations) in the mass-spring-damper system: starting from the initial state $x(0) = [0, 0, L, 0]^T$, with no control and no disturbances, the trajectories $y_1(\cdot)$ and $y_2(\cdot)$ oscillate while converging to the equilibria $y_1^\star$ and $y_2^\star$.

define the system's output as $y = Cx$ where $y = [y_1, y_2]^T$ and $C = \left[\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right]$ then `Safe` specifies the desired values of the output.

1.3. The Green Scheduling Problem

We now generalize and formally define the Green Scheduling problem as discussed in the motivation (Section 1.1) and in the running examples (Section 1.2).

1.3.1. System's Model and Assumptions

In this dissertation, we consider affine dynamical systems with constrained disturbances described by the differential equation

$$\begin{aligned} &\dot{x}(t) = Ax(t) + (B_0 + Bu(t)) + Wd(t), \quad \forall t \geq 0 \\ &x(0) \in \mathcal{X}_0. \end{aligned} \tag{1.9}$$

Here

- $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state vector where $\mathcal{X}$ is the state space, i.e., the set of all valid values of state x. In most cases $\mathcal{X}$ is the entire space $\mathbb{R}^n$. However, in practice, the state variables (especially those having physical meaning) are always bounded, hence

$\mathcal{X}$ is in fact bounded. The boundedness of $\mathcal{X}$ will be used in several results later.

- $\mathcal{X}_0 \subseteq \mathcal{X}$ is the set of initial states.
- $d \in \mathcal{D} \subseteq \mathbb{R}^q$ is the disturbance input vector (i.e., unknown inputs) and is constrained in a convex and compact (i.e., closed and bounded) subset $\mathcal{D}$: $d(t) \in \mathcal{D}\ \forall t \geq 0$.
- We suppose that the control inputs are binary, that is each control input u_i can only receive either value 0 or value 1. Therefore the control input vector $u \in \{0,1\}^m$ is a binary vector. Furthermore, u is constrained in a non-empty subset $\mathcal{U} \subseteq \{0,1\}^m$ of valid control inputs, meaning that $u(t) \in \mathcal{U}$ for all $t \geq 0$. Note that since m is finite, both $\{0,1\}^m$ and $\mathcal{U}$ are finite sets.
- Matrix $A \in \mathbb{R}^{n\times n}$ is called the state matrix and is assumed to be Hurwitz, i.e., all its eigenvalues have strictly negative real parts.
- Matrices $B \in \mathbb{R}^{n\times m}$ and $W \in \mathbb{R}^{n\times q}$ are respectively the control input matrix and the disturbance input matrix. Vector $B_0 \in \mathbb{R}^n$ is called the affine vector.

Let $\mathfrak{F}(X,Y)$ denote the set of all measurable functions from set X to set Y. A signal corresponding to variable y in a set Y, denoted $y(\cdot)$, is a member of the set $\mathfrak{F}(\mathbb{R}^+,Y)$ of all measurable functions from the time set $\mathbb{R}^+ := \{t \in \mathbb{R} : t \geq 0\}$ to Y. A disturbance signal $d(\cdot)$ for the system is *admissible* if it satisfies $d(t) \in \mathcal{D}$ for all $t \geq 0$, that is $d(\cdot) \in \mathfrak{F}(\mathbb{R}^+,\mathcal{D})$. Similarly, an admissible control signal $u(\cdot)$ for the system is a member of $\mathfrak{F}(\mathbb{R}^+,\mathcal{U})$. Given an admissible disturbance signal $d(\cdot)$ and an admissible control signal $u(\cdot)$, a state trajectory $x(\cdot)$ of the system is a continuous signal that satisfies the differential equation (1.9) at all time $t \geq 0$. For any initial state $x(0)$, the state trajectory $x(\cdot)$ exists and is unique (Rugh, 1996).

The model in Equation (1.9) can be extended to a system with switching state matrix

$$\dot{x}(t) = \left(A_0 + \textstyle\sum_{i=1}^{m} A_i u_i(t)\right) x(t) + (B_0 + Bu(t)) + Wd(t), \quad \forall t \geq 0 \tag{1.10}$$

$$x(0) \in \mathcal{X}_0$$

where $A_i \in \mathbb{R}^{n \times n}$ for $i = 0, 1, \ldots, m$. Compared to Equation (1.9), the state matrix is no longer constant but depends affinely on the control input u. Obviously the system (1.9) is a special case of the system (1.10) with $A_i = 0$ for all i. The system may also have output $y \in \mathbb{R}^p$ defined as $y(t) = Cx(t)$ where $C \in \mathbb{R}^{p \times n}$ is called the output matrix. For clarity of presentation, in this dissertation we will mainly consider the system (1.9) with constant state matrix and no outputs, and briefly discuss extensions of the results to the system (1.10) with switching state matrix and systems with outputs y.

1.3.2. Green Scheduling Problem

A control signal $u(\cdot)$ can be thought of as a schedule that turns on-off and coordinates the individual systems. Hence, in this dissertation the terms *control signal* and *schedule* will be used interchangeably for $u(\cdot)$. Let $\texttt{Safe} \subset \mathcal{X}$ be a compact set of desired states of the system. For systems with outputs y, $\texttt{Safe} \subset \mathbb{R}^p$ is defined for the outputs and represents desired system's outputs. The goal of the Green Scheduling problem is to devise a scheduling strategy for the system so that from any initial state $x(0)$, the state trajectory $x(\cdot)$ is always driven to the set `Safe` in finite time and is maintained inside `Safe` indefinitely, regardless of the disturbances (constrained in $\mathcal{D}$). Such a state trajectory is said to be *safe* and is formally defined in Definition 1.1.

Definition 1.1 *A state trajectory $x(\cdot)$ is safe if there exists a finite time $0 \leq \tau < +\infty$ such that $x(t) \in$ `Safe` for all $t \geq \tau$.* □

Because the scheduler/controller does not know the disturbances d except their constraint set $\mathcal{D}$, it must be *robust* to the disturbances, meaning that it must be able to render the system's trajectory safe regardless of the disturbances. Schedules and scheduling strategies for the system can generally be classified into two types:

- A *feedforward schedule* $u(\cdot)$ is predetermined at the initial time (possibly based on

the initial state) and is applied unalterably. Thus, no information of the current system's states will be utilized to correct the system's behavior for the influence of the disturbances. Representative of feedforward schedules are periodic schedules, for which the control inputs are always repeated after exactly a certain amount of time, i.e., $u(t) = u(t + \delta)$ for all $t \geq 0$ where δ is the time period.

- Contrarily, a *state feedback scheduling strategy* uses the knowledge of the current (and possibly past) system's states, which are fed back to the scheduler, to adjust the behavior of the system for the disturbances. A feedback scheduling strategy is typically represented as a feedback law, which is a function $\kappa : \mathcal{X} \to \mathcal{U}$ that maps state $x \in \mathcal{X}$ to an admissible control input $\kappa(x) \in \mathcal{U}$. The resulted state trajectory $x(\cdot)$ satisfies the *closed-loop* differential equation $\dot{x}(t) = Ax(t) + B_0 + B\kappa(x(t)) + Wd(t)$, $\forall t \geq 0$, with initial state $x(0) \in \mathcal{X}_0$.

Obviously, by definition, feedback scheduling strategies are more robust to disturbances than feedforward schedules are. However, when disturbances are absent or very small, feedforward schedules can still achieve the goal of Green Scheduling while being simpler to synthesize and implement. Hereafter, the term *schedule* (or *control signal*) will mean a feedforward schedule and the term *scheduling strategy* (or *control strategy*, or *control law*) will mean a feedback scheduling strategy. We now define the notions of safe schedules and safe scheduling strategies.

Definition 1.2 (Safe Schedules) *Given an initial state $x(0) \in \mathcal{X}_0$, a schedule $u(\cdot)$ is safe if for any admissible disturbance signal $d(\cdot)$, the resulted state trajectory $x(\cdot)$ is safe.* □

Definition 1.3 (Safe Scheduling Strategies) *A scheduling strategy $\kappa : \mathcal{X} \to \mathcal{U}$ is safe if for any initial state $x(0) \in \mathcal{X}_0$ and any admissible disturbance signal $d(\cdot)$, the resulted state trajectory $x(\cdot)$ is safe.* □

In general, the actuation of the control inputs u is restricted to a known finite set $\mathcal{U}$ of binary vectors. In many cases, being inspired by the peak demand reduction problem in Section 1.1, we describe $\mathcal{U}$ by a constraint on the number of inputs u_i, $i = 1, \ldots, m$, that can be activated

simultaneously. Let k be a given integer such that $0 \leq k \leq m$. Then at any time $t \geq 0$, $\|u(t)\|_1 = \sum_{i=1}^{m} u_i(t) \leq k$. In other words, $\mathcal{U}$ is defined as $\mathcal{U} := \{u \in \{0,1\}^m : \|u\|_1 \leq k\}$. Henceforth, this type of constraint on u will be called the *n-choose-k* case.

1.3.3. Two Important Questions of the Green Scheduling Problem

Given a system's model as in Section 1.3.1 and a safe set `Safe` of desired states, there are two important questions regarding the Green Scheduling problem:

1. **Schedulability analysis:** Does there exist a safe schedule or a safe scheduling strategy for the system?

2. **Scheduling synthesis:** If there does, then how to synthesize a safe schedule or a safe scheduling strategy for the system?

This dissertation will answer these questions.

1.4. Approaches to Green Scheduling from Different Disciplines

The Green Scheduling problem can be viewed as a scheduling problem or a control problem. Thus, different disciplines will provide different approaches to the analysis and synthesis questions. In this section, we briefly discuss the Green Scheduling problem from the view of various disciplines and how we will use their approaches in the dissertation.

1.4.1. Real-time Scheduling

The Green Scheduling problem can be viewed as a resource allocation problem, in which multiple systems share a limited resource (e.g., electricity energy) while each of them needs to satisfy certain local safety conditions. From this aspect, the Green Scheduling problem is similar to multiprocessor real-time scheduling with full migration (Davis and Burn, 2009). Scheduling of real-time computing tasks under resource constraints is a well-developed research area with a wide range of well-studied scheduling algorithms (Liu, 2000; Buttazzo, 2011). Although these real-time scheduling algorithms, e.g., the rate monotonic and the

earliest deadline first (EDF) algorithms, may be applied to such resource sharing problems as the Green Scheduling problem, they impose stringent constraints on the task model. Generally, conventional real-time scheduling is restricted to tasks whose worst case execution times are fixed and known in advance. However, for control systems, this assumption does not effectively capture the system's behavior whose evolution is dependent on the plant dynamics, the safety specifications, and the environmental conditions. Therefore, conventional real-time scheduling algorithms are not directly applicable to the Green Scheduling problem.

Nevertheless, there have been several recent attempts to apply real-time scheduling algorithms, for example the EDF algorithm, to scheduling of electric loads for peak demand reduction (Vedova et al., 2010; Facchinetti et al., 2010; Facchinetti and Vedova, 2011; Subramanian et al., 2012). However, they were limited to simple dynamics of decoupled systems with no interactions, and did not directly handle disturbances. We will show in Chapters 2 and 3 that, under certain restrictive assumptions, the periodic scheduling approach of real-time scheduling is applicable to the Green Scheduling problem and results in simple and scalable schedulability analysis and scheduling synthesis methods.

1.4.2. Control Theory

Obviously, the Green Scheduling problem as it is formulated in Section 1.3 is a control problem. Nonetheless, it is distinct from conventional control problems in two aspects:

- In conventional control problems, the control inputs are continuous (u can take any value in $\mathbb{R}^m$); while in Green Scheduling, the control inputs are binary and are constrained in a finite set $\mathcal{U}$.
- Conventional control theory is usually interested in the stability of the controlled system, while Green Scheduling focuses on safety conditions of the controlled systems.

Since there are only a finite number of control input vectors in $\mathcal{U}$, the Green Scheduling problem can be formulated as a switched or hybrid system (Alur et al., 1995; Henzinger, 1996;

Liberzon, 2003; Sun and Ge, 2005; Lin and Antsaklis, 2009). Then analysis and synthesis methods for switched systems can be directly applied. There has been a vast literature on safety verification and safe switching controller synthesis for switched and hybrid systems (see e.g., Alur et al., 1995, 1997; Lygeros et al., 1999; Alur et al., 2000; Asarin et al., 2000; De Santis et al., 2004; Jha et al., 2011; Alur, 2011). However, the number of discrete modes in the Green Scheduling problem is often very large. For example, the small-scale room-heater system in Section 1.2.1 has 6 control inputs, thus a peak constraint $k = 4$ will result in 57 distinct modes. A medium-scale system with 20 control inputs and peak constraint $k = 14$ already has more than one million discrete modes, let alone a large-scale system with 100 or even 1000 control inputs. Most of the analysis and synthesis methods developed for switched systems are not scalable in terms of the number of discrete modes and the dimension of the state space. Therefore their applicability to the Green Scheduling problem is limited to only small-scale systems.

Despite the aforementioned differences, techniques from control theory and switched system theory will be used extensively in the later development of schedulability analysis and scheduling synthesis for the Green Scheduling problem. In particular, the averaging technique (Tokarzewski, 1987; Sun and Ge, 2005) will be used to derive schedulability conditions and to construct periodic schedules. Playing a very important role in the analysis and synthesis methods for systems subject to disturbances are the theory of Lyapunov functions (Artstein, 1983; Freeman and Kokotovic, 1996; Liberzon et al., 2002) and attracting sets (Khalil, 1992; Grüne, 2002). Finally, reachability analysis and computation (Mitchell, 2007) will be utilized to develop safe scheduling policies for discrete-time systems.

1.5. Contributions

The main contributions of the dissertation, also an overview of what to come in the later chapters, are summarized below:

- We establish a spectrum of schedulability analysis methods for various types of systems.

Depending on whether the systems are decoupled or coupled and whether there are disturbances, an appropriate method can be selected. These methods are established and presented in Chapter 2, in the increasing order of complexity of the Green Scheduling problem. They also differ in their scalability, with simple methods being applicable to large-scale systems with 1000 control inputs or even more, and more complex methods being applicable to systems with a few hundreds control inputs at best.

- We develop simple and scalable periodic scheduling synthesis methods for systems with simple dynamics and no disturbances in Chapter 3.

- When disturbances are available, state feedback is necessary to make scheduling robust. We develop various feedback scheduling strategies for the Green Scheduling problem in Chapter 4. In Section 4.2, basic periodic scheduling strategies are extended to be robust to small disturbances. For more complex Green Scheduling problems with larger disturbances, event-triggered and self-triggered feedback scheduling strategies based on attracting sets are developed in Section 4.3. Finally, backward reachability analysis is used to synthesize safe scheduling strategies for discrete-time Green Scheduling problems in Section 4.4.

- We apply the results developed in Chapters 2 to 4 to scheduling of radiant heating and cooling systems for peak electricity demand reduction in Chapter 5. The case studies presented in this chapter demonstrate the effectiveness as well as the limitations of several Green Scheduling strategies.

Chapter 2

Green Schedulability

This chapter develops necessary and sufficient conditions for a Green Scheduling system to be schedulable safely under constraints on the control inputs. These conditions not only enable verification of schedulability and calculation of the peak constraint but also establish the theoretical foundations for the scheduling synthesis methods presented in later chapters. Thus, this chapter provides the key theoretical results for the entire dissertation. Most of the content of this chapter is expanded from our previous work (Nghiem et al., 2011b,a; Li et al., 2011; Nghiem et al., 2012a,b).

2.1. Introduction

We consider the Green Scheduling problem for the general affine dynamical system with constant state matrix

$$\dot{x}(t) = Ax(t) + (B_0 + Bu(t)) + Wd(t) \tag{2.1}$$

or with switching state matrix

$$\dot{x}(t) = \left(A_0 + \sum_{i=1}^{m} A_i u_i(t) \right) x(t) + (B_0 + Bu(t)) + Wd(t). \tag{2.2}$$

As stated in Section 1.3.1 on page 12, we assume that the state matrix A in Equation (2.1) is Hurwitz. For the sake of clarity, the results in this chapter are developed for systems without outputs, where the safe set $\texttt{Safe} \subseteq \mathcal{X}$ is defined for the state variables x. For systems with outputs $y(t) = Cx(t)$ and with $\texttt{Safe} \subseteq \mathbb{R}^p$ being defined for output variable y, similar results can be derived straightforwardly and will be discussed in separate subsections.

2.1.1. Green Schedulability

Recall that a state trajectory $x(\cdot)$ is *safe* if there exists a finite time $\tau \geq 0$ such that $x(t) \in \texttt{Safe}$ for all $t \geq \tau$. A schedule or a scheduling strategy is safe if it always results in safe state trajectories regardless of the disturbances (cf. Definitions 1.2 and 1.3 on page 15). The system is said to be *schedulable* if for any initial state and any admissible disturbance signal, there exists a schedule or a scheduling strategy that drives the system's state to the safe set.

Definition 2.1 (Schedulability) *System* (2.1) *(or* (2.2)*) is* schedulable *if there exists a safe scheduling strategy* $\kappa : \mathcal{X} \to \mathcal{U}$ *for it, or if there exists a safe schedule* $u(\cdot) \in \mathfrak{F}(\mathbb{R}^+, \mathcal{U})$ *for each initial state* $x(0) \in \mathcal{X}_0$*. If the system is not schedulable, it is* non-schedulable. □

In the specific case when the valid control input set $\mathcal{U}$ is n-choose-k, we use the term *k-schedulable* to emphasize the peak constraint k on the control inputs.

In this chapter, necessary and sufficient conditions for which the system is (k-)schedulable are derived.

Necessary Schedulability Conditions A necessary schedulability condition for the system is a condition that must be satisfied if the system is schedulable. In other words, if this condition is not satisfied, the system is non-schedulable. A necessary schedulability condition often allows us to derive a lower bound for the peak constraint k.

Sufficient Schedulability Conditions If a sufficient schedulability condition for the system holds, the system is schedulable.

In this dissertation, we focus more on sufficient conditions than on necessary conditions for three reasons:

1. We are more interested in schedulable systems than non-schedulable ones.

2. Sufficient conditions not only allow determining whether a system is schedulable but

also help us derive scheduling policies for it. Indeed, most of the scheduling algorithms presented in the following chapters are based on the sufficient schedulability conditions established in this chapter.

3. Different sufficient schedulability conditions result in different approaches for scheduling, each of which is suitable for a particular type of systems or applications.

Organization of Chapter 2

This chapter is organized so that the schedulability results are developed in the increasing order of complexity of the system's dynamics. The next section discusses two general approaches used in this chapter for deriving necessary and sufficient schedulability conditions. In Sections 2.3 and 2.4, schedulability conditions based on the periodic scheduling approach are developed for systems with decoupled affine dynamics and for systems without disturbances. Using another approach, schedulability analysis results for general affine systems with disturbances are presented in Section 2.5. In this section, we also discuss how the previous conditions for simpler dynamics are special cases of these general conditions. Finally, we summarize the results and conclude the chapter in Section 2.6.

2.2. General Approaches

2.2.1. Sufficient Schedulability Conditions

Generally, to show that a system is schedulable requires constructing a schedule or a scheduling algorithm that can be proved to drive the system's state to the safe set, regardless of the initial state and the disturbances. In Section 1.3.2 on page 14 we classified scheduling policies as (a) feedforward schedules, for which periodic schedules are representative, and (b) feedback scheduling. These two types of scheduling strategies correspond to two approaches to Green Scheduling synthesis, hence to derivation of sufficient schedulability conditions.

In the first approach, namely *periodic scheduling*, we construct a periodic schedule with a

positive time period and show that under this schedule, the state trajectory will enter the safe set after some finite time and will remain safe indefinitely. Because a periodic schedule does not take into account the influence of the disturbances, this approach is only applicable to systems without or with small disturbances. For systems where the disturbances can greatly impact the systems' behavior, it is essential to use feedback scheduling. In the second approach, a scheduling algorithm will determine the feedback control input $u(t)$ based on the current state $x(t)$ so that the trajectory $x(\cdot)$ is always driven to an *attracting* subset of the safe set, regardless of the initial state and the disturbances. This suffices to show the schedulability of the system.

While the feedback scheduling approach is more general and is also applicable to systems without or with small disturbances, it is less scalable computationally than the periodic-scheduling approach. Moreover, implementation of periodic scheduling is simpler than that of feedback scheduling because it usually does not require online computation of the control input nor real-time monitoring of the system's state. Therefore, in this chapter, both approaches will be used to develop sufficient schedulability conditions for the Green Scheduling problem. Specifically, in Sections 2.3 and 2.4 for systems without disturbances, we use the periodic-scheduling approach; while in Section 2.5 for general affine systems with disturbances, we use the feedback-scheduling approach.

2.2.2. Necessary Schedulability Conditions

All the necessary schedulability conditions established in this chapter rely on the notion of *non-schedulability certificates*, or *barrier certificates* (see Prajna, 2005; Prajna and Jadbabaie, 2004; Prajna et al., 2004; Prajna, 2006; Prajna et al., 2007). Let $g : \mathcal{X} \to \mathbb{R}$ be a function of state whose zero superlevel set contains the entire safe set `Safe` (see Figure 2.1 on the following page). If, in addition, its time derivative along the flow of the dynamical system is always negative whenever the system's state is in the safe set, regardless of the control input, then we can conclude that the system is non-schedulable. This is because for any $u(\cdot)$, the value of g evaluated along the state trajectory, i.e., $g(x(t))$, always decreases as long as

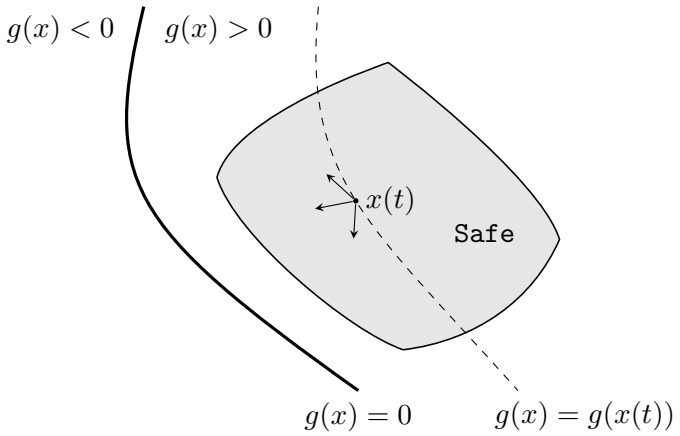


Figure 2.1: Intuition of the non-schedulability certificate g to prove the system is non-schedulable: the solid curve is the zero level set of g; to its right is the zero superlevel set which contains the entire safe set `Safe`. For any $x(t) \in$ `Safe`, the level set of g at $x(t)$ (dashed curve) always moves towards the zero level set, i.e., the value of g always decreases, along any flow of the system (arrows). Hence, no schedule $u(\cdot)$ can keep the state safe indefinitely.

$x(t) \in$ `Safe`. Eventually $x(t)$ must become unsafe. Pictorially, in Figure 2.1, when the state $x(t)$ belongs to `Safe`, the level set of g at $x(t)$ (dashed curve) always moves towards its zero level set (thick solid curve) along any flow of the system (drawn as arrows from $x(t)$). The function g provides a certificate to prove that the system's state will always move away from the safe set, and therefore will never be safe indefinitely.

The following Theorem states the conditions that are satisfied by a non-schedulability certificate g.

Theorem 2.1 *Suppose there exist a differentiable function $g : \mathcal{X} \to \mathbb{R}$ and a positive number $\epsilon > 0$ that satisfy*

$$\forall x \in \texttt{Safe},\ g(x) \geq 0 \tag{2.3a}$$

$$\forall (x, u) \in \texttt{Safe} \times \mathcal{U},\ \exists d \in \mathcal{D} \,:\, \nabla g(x) \cdot f(x, u, d) \leq -\epsilon \tag{2.3b}$$

in which $f(x, u, d)$ is the right-hand side of the dynamics' differential equation (2.1) *or* (2.2)*. Then the system is non-schedulable.* □

The proof of this Theorem can be found in Appendix A.1.1 on page 167.

2.3. Decoupled Affine Monotone Dynamics

We start with a special case of Equation (2.2) where the dynamics of the state variables x_i, $i = 1, \ldots, n$, are decoupled from each other and are monotone. Furthermore, there are no disturbances ($d \equiv 0$). The safe set is a hyper-rectangle in $\mathbb{R}^n$: $\texttt{Safe} = [l_1, h_1] \times \cdots \times [l_n, h_n]$, where $l_i < h_i$ define the desired lower and upper bounds for each state variable x_i. Precisely, the following assumptions are made about the dynamics of the system.

Assumption 2.1 (Decoupled affine monotone dynamics) *The dynamics of each state variable x_i, $i = 1, \ldots, n$, is given by the affine differential equation*

$$\dot{x_i}(t) = \begin{cases} -a_{\text{on},i} x_i(t) + b_{\text{on},i} & \text{if } u_i(t) = 1 \\ -a_{\text{off},i} x_i(t) + b_{\text{off},i} & \text{if } u_i(t) = 0 \end{cases} \tag{2.4}$$

where $a_{\text{on},i} > 0$, $a_{\text{off},i} > 0$, $b_{\text{on},i}$ and $b_{\text{off},i}$ are parameters. Note that each x_i is controlled by an individual input u_i, thus $m = n$. Furthermore, within the bound $[l_i, h_i]$, x_i always grows when $u_i = 1$ and decays when $u_i = 0$, that is

$$-a_{\text{off},i} x_i + b_{\text{off},i} < 0 < -a_{\text{on},i} x_i + b_{\text{on},i} \quad \forall x_i \in [l_i, h_i] \tag{2.5}$$

From inequalities (2.5) and that $a_{\text{on},i} > 0$ and $a_{\text{off},i} > 0$, it is straightforward to verify that

$$\frac{b_{\text{off},i}}{a_{\text{off},i}} < l_i < h_i < \frac{b_{\text{on},i}}{a_{\text{on},i}} \quad \forall i = 1, \ldots, n. \tag{2.6}$$

The monotone dynamics of x_i is illustrated in Figure 2.2 on the following page.

The decoupled monotone affine dynamics is obviously a special case of the dynamics in Equation (2.2) where

$$A_0 = -\operatorname{diag}(a_{\text{off},1}, \ldots, a_{\text{off},n}); \qquad B_0 = [b_{\text{off},1}, \ldots, b_{\text{off},n}]^T$$

$$A_i = (a_{\text{off},i} - a_{\text{on},i})\, E_{i,i} \quad \forall i = 1, \ldots, n; \qquad B = \operatorname{diag}(b_{\text{on},1} - b_{\text{off},1}, \ldots, b_{\text{on},n} - b_{\text{off},n})$$

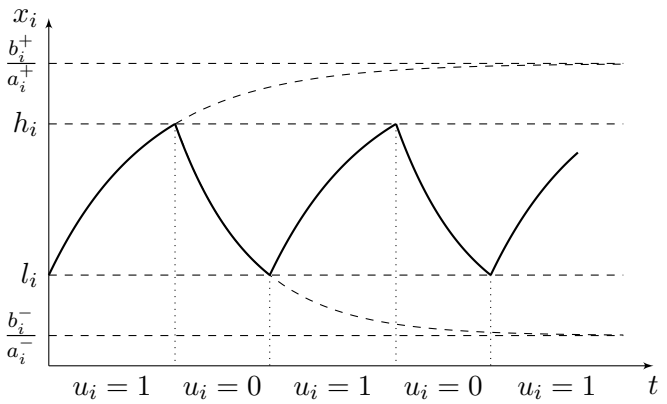


Figure 2.2: Illustration of the decoupled affine monotone dynamics in Assumption 2.1: when $u_i = 1$, x_i increases monotonically; when $u_i = 0$, x_i decreases monotonically. In this figure, x_i is regulated by u_i so that it is always kept within the desired bounds $[l_i, h_i]$.

in which $\text{diag}(a_1, a_2, \ldots)$ denotes the diagonal matrix with diagonal entries $a_1, a_2, \ldots$; and $E_{i,j}$ is the n-by-n matrix with 1 in the $(i,j)^{\text{th}}$ position and 0 everywhere else. An example of this type of systems is the room-heater system in Section 1.2.1 when there are no thermal interactions between the rooms, the ambient air temperature T_a is constant, and there are no internal heat gains.

We consider the n-choose-k case where $k \in \{0, 1, \ldots, n\}$ is the peak constraint imposed on u, i.e., $\mathcal{U} = \{u \in \{0,1\}^n : \|u\|_1 \leq k\}$ or $\|u(t)\|_1 \leq k \; \forall t \geq 0$.

Remark 2.1 The results in this section are not limited to the assumption in Equation (2.5) (i.e., x_i always grows when $u_i = 1$ and decays when $u_i = 0$). Indeed, if some state variables have dynamics satisfying Equation (2.5) while the others have dynamics satisfying the opposite (i.e., x_i always decays when $u_i = 1$ and grows when $u_i = 0$), the results can still be applied by a simple change of variables. □

2.3.1. Necessary and Sufficient Schedulability Condition

As mentioned in Section 2.2.1, an approach to show the schedulability of a system is to construct a safe periodic schedule for it. The sufficient schedulability condition developed in this section is based on periodic scheduling.

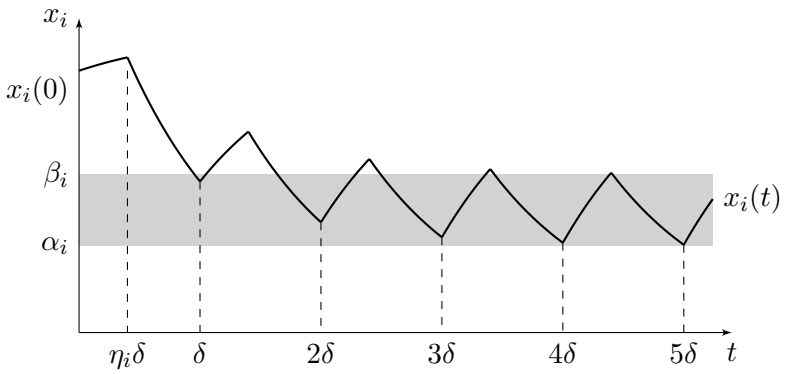


Figure 2.3: Illustration of trajectory $x_i(\cdot)$ under δ-periodic schedule $u_i(\cdot)$ in Equation (2.7): $u_i = 1$ for a fraction η_i of the period then $u_i = 0$ till the end. The trajectory converges to an interval $[\alpha_i, \beta_i]$, painted gray in the figure, which is determined by both η_i and δ.

A δ-periodic control signal $u(\cdot)$, where $\delta > 0$, satisfies $u(t+\delta) = u(t)$ for all $t \geq 0$. The positive number δ is the *time period* of the periodic control signal. Define $\eta_i := \frac{1}{\delta}\int_0^\delta u_i(s)\,ds \in [0,1]$ for each i, which is essentially the fraction of time in a period when the control input u_i is 1. Because $u(\cdot)$ is δ-periodic, $\frac{1}{\delta}\int_t^{t+\delta} u_i(s)\,ds = \eta_i$ for all $t \geq 0$. The value η_i is called the *utilization* of periodic control input i, following the convention in real-time scheduling (Liu, 2000). The utilization vector $\eta \in [0,1]^m$ of all control inputs is defined by stacking the individual utilizations, i.e., $\eta := [\eta_1, \ldots, \eta_m]^T$.

Consider the periodic control input $u_i(\cdot)$. Since u_i can only be either 0 or 1, a basic periodic form of $u_i(\cdot)$ is to alternate between 0 and 1 exactly once in each period δ. Specifically, $u_i(\cdot)$ is given by the following equation

$$u_i(t) = \begin{cases} 1 & \text{if } j\delta \leq t < (j+\eta_i)\,\delta,\ j \in \mathbb{N} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

The trajectory $x_i(\cdot)$ under periodic schedule (2.7) is illustrated in Figure 2.3 on the current page. Intuitively, as $t \to \infty$, $x_i(t)$ will converge to an interval $[\alpha_i, \beta_i]$ determined by both η_i and δ. In Figure 2.3, this interval is visualized as the gray-filled region. By selecting appropriate values for η_i and δ, the interval can be placed inside the desired range $[l_i, h_i]$, thus realizing the state variable x_i safe. Indeed, this intuition is validated by the following

Lemma.

Lemma 2.1 *Given any η_i such that $\underline{\eta}_i < \eta_i < \bar{\eta}_i$ where*

$$\underline{\eta}_i = \frac{a_{\text{off},i} l_i - b_{\text{off},i}}{(a_{\text{off},i} l_i - b_{\text{off},i}) - (a_{\text{on},i} l_i - b_{\text{on},i})} \text{ and } \bar{\eta}_i = \frac{a_{\text{off},i} h_i - b_{\text{off},i}}{(a_{\text{off},i} h_i - b_{\text{off},i}) - (a_{\text{on},i} h_i - b_{\text{on},i})} \quad (2.8)$$

There exists $\delta_i^\star > 0$ such that under the periodic schedule (2.7) with any $0 < \delta < \delta_i^\star$, $x_i(\cdot)$ is safe, i.e., $x_i(t) \in [l_i, h_i]$ $\forall t \geq \tau_i$ for some finite $\tau_i \geq 0$. □

The proof of this Lemma can be found in Appendix A.1.2 on page 167.

According to Lemma 2.1, for each periodic control input $u_i(\cdot)$, there is a range $(\underline{\eta}_i, \bar{\eta}_i)$ for the utilization η_i so that x_i will be safe. The total utilization for all i therefore must also be bounded:

$$\sum_{i=1}^{n} \underline{\eta}_i < \sum_{i=1}^{n} \eta_i < \sum_{i=1}^{n} \bar{\eta}_i$$

It is well-known in real-time scheduling (Liu, 2000; Buttazzo, 2011) that if the total utilization does not exceed k, i.e., $\sum_{i=1}^{n} \eta_i \leq k$, then there exist periodic schedules $u(\cdot)$ satisfying the peak constraint $\sum_{i=1}^{n} u_i(t) \leq k$ for all $t \geq 0$. Thus, we must have $\sum_{i=1}^{n} \underline{\eta}_i < k$. A sufficient schedulability condition can now be stated in the following Theorem, whose proof can be found in Appendix A.1.3 on page 169.

Theorem 2.2 *Given a peak constraint k on the control signal $u(\cdot)$, $k \in \{0, 1, \ldots, n\}$. If $\sum_{i=1}^{n} \underline{\eta}_i < k$ then the system (2.4) is k-schedulable.* □

By Theorem 2.2, $\sum_{i=1}^{n} \underline{\eta}_i < k$ is sufficient for the k-schedulability of system (2.4). It turns out that this condition is also necessary, as justified in the following Theorem.

Theorem 2.3 *Given a peak constraint k on the control signal $u(\cdot)$, $k \in \{0, 1, \ldots, n\}$. If $\sum_{i=1}^{n} \underline{\eta}_i \geq k$ then the system (2.4) is not k-schedulable.* □

Its proof is given in Appendix A.1.4 on page 171.

The following straightforward Corollary restates Theorems 2.2 and 2.3 as a necessary and

sufficient k-schedulability condition.

Corollary 2.1 *The system* (2.4) *is k-schedulable if and only if* $\sum_{i=1}^{n} \underline{\eta}_i < k$. □

Example 2.1 Consider the room-heater example in Section 1.2.1. However, we assume that there is no thermal interaction between the rooms, i.e., the thermal conductance K_{ij}, $i \neq j$, between adjacent rooms are all zeros. Furthermore, there are no heat gains in the rooms ($Q_{g,i} = 0,\ \forall i$) and the ambient air temperature is constant and known. In other words, the system does not have any disturbance. With these assumptions, the thermal dynamics of the rooms are decoupled and are given by $C_i\dot{T}_i(t) = -K_iT_i(t) + K_iT_a + P_iu_i(t)$ for each $i = 1, \ldots, n$ (cf. Equation (1.1) on page 6). Here, T_a is the fixed ambient air temperature: $T_a(t) = T_a$ for all $t \geq 0$. These differential equations can be written in the form of Equation (2.4) as

$$\dot{x_i}(t) = \begin{cases} -a_{\text{on},i}x_i(t) + b_{\text{on},i} & \text{if } u_i(t) = 1 \\ -a_{\text{off},i}x_i(t) + b_{\text{off},i} & \text{if } u_i(t) = 0 \end{cases}$$

where $x_i := T_i$, $a_{\text{on},i} = a_{\text{off},i} = \frac{K_i}{C_i} > 0$, $b_{\text{on},i} = \frac{K_iT_a+P_i}{C_i}$, and $b_{\text{on},i} = \frac{K_iT_a}{C_i}$.

We now apply the schedulability condition in Theorem 2.2 to the small-scale case with parameter values given in Table 1.1 on page 8. The desired temperature thresholds are the same for all rooms and are $l_i = 20\,^{\circ}\text{C}$, $h_i = 24\,^{\circ}\text{C}$. The ambient air temperature is fixed at $T_a = 5\,^{\circ}\text{C}$. The valid range $(\underline{\eta}_i, \bar{\eta}_i)$ for the utilization η_i of each heater i are calculated according to Equation (2.8) and are given in the following table

	Heater 1	Heater 2	Heater 3	Heater 4	Heater 5	Heater 6
$\underline{\eta}_i$	0.4939	0.5109	0.4875	0.4760	0.4525	0.4510
$\bar{\eta}_i$	0.6256	0.6471	0.6175	0.6029	0.5732	0.5713

Because $\sum_{i=1}^{n} \underline{\eta}_i = 2.8717$, the system is k-schedulable with any peak constraint $k \geq 3$. By Theorem 2.3, the condition $k \geq 3$ is also necessary.

Since the calculation of $\underline{\eta}_i$ and $\bar{\eta}_i$ involves only simple algebraic operations, it can be performed

efficiently. To demonstrate how scalable checking the schedulability condition is, we scale up the room-heater system by randomly generating the parameters for 1000 rooms and 1000 heaters (cf. Section 1.2.1). The computation of $\underline{\eta}_i$ and $\bar{\eta}_i$ for all heaters took an average of 0.1 ms on MATLAB™ to complete, and the minimal feasible peak constraint is 485. □

2.3.2. Intuition of the Necessary and Sufficient Schedulability Condition

Consider the value $\underline{\eta}_i$ as defined in Equation (2.8). It is straightforward to show that $0 < \underline{\eta}_i < 1$ and $\underline{\eta}_i$ is strictly increasing as l_i increases. We have that

$$\underline{\eta}_i \left|-a_{\text{on},i} l_i + b_{\text{on},i}\right| = (1 - \underline{\eta}_i) \left|-a_{\text{off},i} l_i + b_{\text{off},i}\right|$$

in which $|-a_{\text{on},i} l_i + b_{\text{on},i}|$ and $|-a_{\text{off},i} l_i + b_{\text{off},i}|$ are respectively the growing and decaying rates of x_i at l_i. As x_i increases, the growing rate decreases while the decaying rate increases. It follows that, intuitively, $\underline{\eta}_i$ is the minimum fraction of time that u_i must be 1 in order to keep x_i above l_i. For example, if $\underline{\eta}_i = 0.6$ then on average u_i must be 1 for at least 60% of the time for x_i to stay safe. In the language of real-time scheduling (Liu, 2000), $\underline{\eta}_i$ is the *minimum utilization* of control input u_i.

With this utilization-based interpretation of $\underline{\eta}_i$, Theorem 2.2 and Theorem 2.3 become more intuitive. If $\sum_{i=1}^{n} \underline{\eta}_i > k$, the total minimum utilization of the control inputs exceeds the resource capacity (or resource constraint), thus they are non-schedulable. On the other hand, if $\sum_{i=1}^{n} \underline{\eta}_i < k$ then they are schedulable.

2.3.3. Dynamics Bounded by Affine Monotone Dynamics

The decoupled affine monotone dynamics in Equation (2.4) may not accurately characterize the actual systems because in practice, systems are usually subject to disturbances and interactions between their sub-systems. In this section, the schedulability condition for decoupled affine monotone dynamics is extended to a larger class of dynamics which can take into account small disturbances and small inter-system interactions. Specifically, we will

consider system dynamics that are bounded between affine monotone dynamics (hereafter called *affinely bounded monotone dynamics*), as defined below.

The system's dynamics is described by

$$\dot{x_i}(t) = f_i(x, u_i, d), \qquad i = 1, \ldots, n$$

in which x_i, x, u and d have the usual interpretations. For each i, the function $f_i : \mathcal{X} \times \{0,1\} \times \mathcal{D} \to \mathbb{R}$ specifies the dynamics of x_i subject to the influence of other state variables (vector x) and of the disturbances (vector d). We assume that for each i, the functions $f_i(x, u_i, d)|_{u_i=0}$ and $f_i(x, u_i, d)|_{u_i=1}$ are Lipschitz-continuous, so that the trajectories $x_i(\cdot)$ exist and are unique. Furthermore, we assume that for all $x \in \mathcal{X}$ and all $d \in \mathcal{D}$, the dynamics of each x_i is bounded between decoupled affine monotone dynamics.

Assumption 2.2 (Affinely bounded monotone dynamics) *For each $i = 1, \ldots, n$, there exist $\underline{a}_{\text{off},i} > 0$, $\bar{a}_{\text{off},i} > 0$, $\underline{a}_{\text{on},i} > 0$, $\bar{a}_{\text{on},i} > 0$, $\underline{b}_{\text{off},i}$, $\bar{b}_{\text{off},i}$, $\underline{b}_{\text{on},i}$, $\bar{b}_{\text{on},i}$ such that for all $(x, d) \in \mathcal{X} \times \mathcal{D}$*

$$-\underline{a}_{\text{off},i} x_i + \underline{b}_{\text{off},i} \leq f_i(x, u_i, d)|_{u_i=0} \leq -\bar{a}_{\text{off},i} x_i + \bar{b}_{\text{off},i}$$

and

$$-\underline{a}_{\text{on},i} x_i + \underline{b}_{\text{on},i} \leq f_i(x, u_i, d)|_{u_i=1} \leq -\bar{a}_{\text{on},i} x_i + \bar{b}_{\text{on},i}.$$

In addition, the quadruples $(\underline{a}_{\text{off},i}, \underline{b}_{\text{off},i}, \underline{a}_{\text{on},i}, \underline{b}_{\text{on},i})$ and $(\bar{a}_{\text{off},i}, \bar{b}_{\text{off},i}, \bar{a}_{\text{on},i}, \bar{b}_{\text{on},i})$ satisfy the monotonicity assumption in Equation (2.5). □

Essentially, Assumption 2.2 means that each trajectory $x_i(\cdot)$ is always bounded between two affine monotone trajectories under the same control signal $u_i(\cdot)$ and starting from the same initial state $x_i(0)$, as illustrated in Figure 2.4 on the next page. Furthermore, these bound trajectories are decoupled from the disturbances as well as from the other state variables.

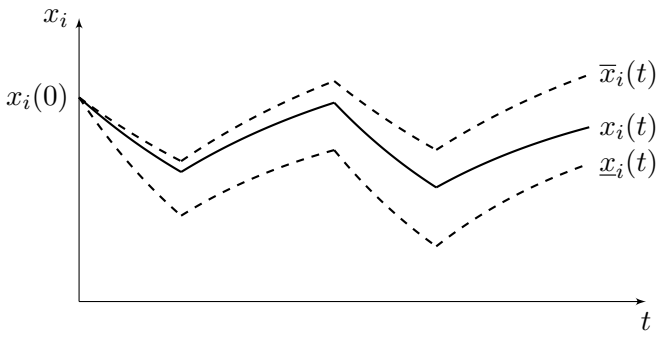


Figure 2.4: Illustration of an affinely bounded monotone trajectory: the actual trajectory $x_i(\cdot)$ (solid line) is bounded between two affine monotone trajectories (dashed lines): the lower-bound trajectory $\underline{x}_i(\cdot)$ and the upper-bound trajectory $\bar{x}_i(\cdot)$. The bound trajectories start from the same initial state $x_i(0)$ and are under the same control signal $u_i(\cdot)$.

We first define the dynamical system which generates a trajectory $\underline{x}_i(\cdot)$ that bounds $x_i(\cdot)$ from below (Figure 2.4).

Definition 2.2 (Lower-bound system) *For each i, define the lower-bound system $\underline{\Sigma}_i$ with state variable $\underline{x}_i \in \mathbb{R}$, control input $\underline{u}_i \in \{0,1\}$, and dynamics given by the differential equation*

$$\dot{\underline{x}}_i(t) = \begin{cases} -\underline{a}_{\text{off},i}\underline{x}_i(t) + \underline{b}_{\text{off},i} & \text{if } \underline{u}_i(t) = 0, \\ -\underline{a}_{\text{on},i}\underline{x}_i(t) + \underline{b}_{\text{on},i} & \text{if } \underline{u}_i(t) = 1. \end{cases}$$

□

The dynamical system whose trajectory $\bar{x}_i(\cdot)$ bounds $x_i(\cdot)$ from above can also be defined similarly.

Definition 2.3 (Upper-bound system) *For each i, define the upper-bound system $\overline{\Sigma}_i$ with state variable $\bar{x}_i \in \mathbb{R}$, control input $\bar{u}_i \in \{0,1\}$, and dynamics given by the differential equation*

$$\dot{\bar{x}}_i(t) = \begin{cases} -\bar{a}_{\text{off},i}\bar{x}_i(t) + \bar{b}_{\text{off},i} & \text{if } \bar{u}_i(t) = 0, \\ -\bar{a}_{\text{on},i}\bar{x}_i(t) + \bar{b}_{\text{on},i} & \text{if } \bar{u}_i(t) = 1. \end{cases}$$

□

Intuitively, these bound systems represent the extreme cases of the dynamics of x_i and therefore $x_i(\cdot)$ must be bounded by their trajectories, as confirmed by the following Lemma.

Lemma 2.2 *Given any control input $u(\cdot) \in \mathfrak{F}(\mathbb{R}^+, \mathcal{U})$ and disturbance signals $d(\cdot) \in$*

$\mathfrak{F}(\mathbb{R}^+, \mathcal{D})$. *For each* $i = 1, \ldots, n$, *the trajectory* $x_i(\cdot)$ *is bounded by* $\underline{x}_i(t) \leq x_i(t) \leq \bar{x}_i(t)$ *for all* $t \geq 0$. *Here* $\underline{x}_i(\cdot)$ *and* $\bar{x}_i(\cdot)$ *are respectively the trajectories of the lower-bound system* $\underline{\Sigma}_i$ *(Definition 2.2) and the upper-bound system* $\overline{\Sigma}_i$ *(Definition 2.3) with the same initial condition* $\underline{x}_i(0) = \bar{x}_i(0) = x_i(0)$ *and the same control input* $\underline{u}_i(\cdot) \equiv \bar{u}_i(\cdot) \equiv u_i(\cdot)$. □

PROOF This result directly follows from the classical Müller's existence theorem, i.e., the comparison theorem for differential inequalities (Müller, 1926; Walter, 1997; Kieffer et al., 2006). ■

Evidently, if these bound systems are k-schedulable then the original system is *robustly* k-schedulable, meaning that it is k-schedulable despite the disturbances and the interactions between the state variables. The following sufficient schedulability condition is analogous to the condition in Theorem 2.2 on page 28, hence its proof is omitted.

Theorem 2.4 *Suppose that for each* i, $\underline{\eta}_i < \bar{\eta}_i$ *where*

$$\underline{\eta}_i = \frac{\underline{a}_{\text{off},i} l_i - \underline{b}_{\text{off},i}}{\left(\underline{a}_{\text{off},i} l_i - \underline{b}_{\text{off},i}\right) - \left(\underline{a}_{\text{on},i} l_i - \underline{b}_{\text{on},i}\right)} \quad \textit{and} \quad \bar{\eta}_i = \frac{\bar{a}_{\text{off},i} h_i - \bar{b}_{\text{off},i}}{\left(\bar{a}_{\text{off},i} h_i - \bar{b}_{\text{off},i}\right) - \left(\bar{a}_{\text{on},i} h_i - \bar{b}_{\text{on},i}\right)}.$$

Given a peak constraint k *on the control signal* $u(\cdot)$, $k \in \{0, 1, \ldots, n\}$. *If* $\sum_{i=1}^{n} \underline{\eta}_i < k$ *then the affinely bounded monotone system is* k*-schedulable.* □

Similar to Theorem 2.3 on page 28, a necessary schedulability condition can be derived in Theorem 2.5. However, note that the necessary condition does not complement the sufficient condition in Theorem 2.4 due to their robustness to disturbances and inter-system interactions.

Theorem 2.5 *Given a peak constraint* k *on the control signal* $u(\cdot)$, $k \in \{0, 1, \ldots, n\}$. *If*

$$\sum_{i=1}^{n} \frac{\bar{a}_{\text{off},i} l_i - \bar{b}_{\text{off},i}}{\left(\bar{a}_{\text{off},i} l_i - \bar{b}_{\text{off},i}\right) - \left(\bar{a}_{\text{on},i} l_i - \bar{b}_{\text{on},i}\right)} \geq k$$

then the affinely bounded monotone system is not k*-schedulable.* □

We remark that the class of affinely bounded monotone dynamics is larger and more practical than the class of decoupled affine monotone dynamics in Equation (2.4) because it can take

account of small inter-dependencies between the state variables as well as small disturbances. This capability is illustrated in the following example.

Example 2.2 We continue the room-heater example in Example 2.1 and make it more realistic by restoring the thermal interactions between adjacent rooms and allowing small variations in the ambient air temperature. Specifically, the thermal conductance K_{ij}, $i \neq j$, can be non-zero and are given in Table 1.1 on page 8; T_a can vary between $T_{a,\min} = 5\,^\circ\text{C}$ and $T_{a,\max} = 8\,^\circ\text{C}$ while the heat gains are still not present ($Q_{g,i} = 0\ \forall i = 1, \ldots, n$). Furthermore, because the room temperatures are going to be kept between $20\,^\circ\text{C}$ and $24\,^\circ\text{C}$, we can assume that $20 \leq T_i \leq 24$ for all $i = 1, \ldots, n$ and obtain the following bounds on the differential equation for each T_i (cf. Equation (1.1) on page 6).

$$
\begin{aligned}
-\frac{K_i + \sum_{j\neq i} K_{ij}}{C_i} T_i(t) + \frac{K_i}{C_i} T_{a,\min} + \frac{\sum_{j\neq i} K_{ij}}{C_i} l_j + \frac{P_i}{C_i} u_i(t) \leq \\
\dot{T}_i(t) = -\frac{K_i + \sum_{j\neq i} K_{ij}}{C_i} T_i(t) + \frac{K_i}{C_i} T_a(t) + \frac{\sum_{j\neq i} K_{ij}}{C_i} T_j(t) + \frac{P_i}{C_i} u_i(t) \\
\leq -\frac{K_i + \sum_{j\neq i} K_{ij}}{C_i} T_i(t) + \frac{K_i}{C_i} T_{a,\max} + \frac{\sum_{j\neq i} K_{ij}}{C_i} h_j + \frac{P_i}{C_i} u_i(t).
\end{aligned}
$$

Therefore, the system satisfies the affinely bounded monotone dynamics assumptions in Assumption 2.2 with

$$
\begin{aligned}
\underline{a}_{\text{off},i} = \underline{a}_{\text{on},i} = \bar{a}_{\text{off},i} = \bar{a}_{\text{on},i} = \frac{1}{C_i}\left(K_i + \sum_{j\neq i} K_{ij}\right) \\
\underline{b}_{\text{off},i} = \frac{1}{C_i}\left(K_i T_{a,\min} + \sum_{j\neq i} K_{ij} l_j\right), \quad \underline{b}_{\text{on},i} = \frac{1}{C_i}\left(K_i T_{a,\min} + \sum_{j\neq i} K_{ij} l_j P_i\right) \\
\bar{b}_{\text{off},i} = \frac{1}{C_i}\left(K_i T_{a,\max} + \sum_{j\neq i} K_{ij} h_j\right), \quad \bar{b}_{\text{on},i} = \frac{1}{C_i}\left(K_i T_{a,\max} + \sum_{j\neq i} K_{ij} h_j P_i\right).
\end{aligned}
$$

Using the parameters for the small-scale room-heater example Table 1.1 on page 8, we can compute the utilization bounds $\underline{\eta}_i$ and $\bar{\eta}_i$ for each heater i according to Theorem 2.4, as reported below:

	Heater 1	Heater 2	Heater 3	Heater 4	Heater 5	Heater 6
$\underline{\eta}_i$	0.4939	0.5109	0.4875	0.4760	0.4525	0.4510
$\bar{\eta}_i$	0.5268	0.5449	0.5200	0.5077	0.4827	0.4811

Since $\sum_{i=1}^{n} \underline{\eta}_i = 2.8718$, from Theorem 2.4, we can conclude that the system is k-schedulable for all $k \geq 3$. □

2.4. Affine Dynamics Without Disturbances

Section 2.3 considered systems in which the dynamics of the state variables are decoupled, either by nature or by bound dynamics. However, in practical systems, the interactions between their sub-systems might be significant enough so that the analysis method by bound dynamics in Section 2.3.3 cannot be applied. In this section, we extend the previous results to these systems.

2.4.1. System's Dynamics

Consider the general affine dynamical system in Equation (2.1). However, we assume that there are no disturbances ($d \equiv 0$). Consequently, the dynamics is described by the differential equation

$$\dot{x}(t) = Ax(t) + (B_0 + Bu(t)) \tag{2.9}$$

in which the state matrix A is Hurwitz. Unlike the decoupled dynamics in Assumption 2.1, the number of control inputs is m, i.e., $u(t) \in \{0,1\}^m$, which can be different from the number of state variables n. The safe set `Safe` is no longer restricted to a hyper-rectangle but can be any compact (i.e., closed and bounded) convex set. It is reasonable to assume `Safe` to be compact and convex since it is usually the case in practice. Again, we consider the n-choose-k case where k is the peak constraint imposed on the control inputs u.

In the next subsection, a necessary and sufficient schedulability condition is developed for system (2.9). A similar condition for systems with switching state matrix (cf. Equation (2.2))

is derived and discussed in Section 2.4.6.

2.4.2. Necessary and Sufficient Schedulability Conditions

Using the same approach as in Section 2.3.1, we investigate periodic control signals for system (2.9) to derive a sufficient schedulability condition for it.

Recall that a δ-periodic control signal $u(\cdot)$, where $\delta > 0$, satisfies $u(t+\delta) = u(t)\ \forall t \geq 0$. The utilization η_i of periodic control input i is the fraction of time in a period where $u_i = 1$, and is defined as $\eta_i := \frac{1}{\delta}\int_0^\delta u_i(t)\,dt \in [0,1]$. The utilization vector $\eta \in [0,1]^m$ of all control inputs is $\eta := [\eta_1, \ldots, \eta_m]^T$.

A standard tool to analyze a linear system under periodic control signals is the *averaging technique* (see Tokarzewski, 1987; Sun and Ge, 2005). Given a δ-periodic control signal $u(\cdot)$ with utilization vector η, the *average system* of (2.9) with respect to η is defined as the time-invariant affine dynamical system

$$\dot{\bar{x}}(t) = A\bar{x}(t) + (B_0 + B\eta) \tag{2.10}$$

with state variable $\bar{x}$ and starting from the same initial state: $\bar{x}(0) = x(0)$. Note that the average system is autonomous, meaning that it does not have any control input. For each initial state $\bar{x}(0)$ and a fixed utilization vector, the trajectory $\bar{x}(\cdot)$ of the average system is unique. Since A is Hurwitz, the average system is uniformly exponentially stable and its trajectories always converge to the unique equilibrium $\bar{x}^\star = -A^{-1}(B_0 + B\eta)$ (Rugh, 1996). Note that A is invertible because it is Hurwitz. If $\bar{x}^\star$ is in the interior of `Safe`, denoted int(`Safe`), then obviously $\bar{x}(\cdot)$ will be safe. If, in addition, the trajectory $x(\cdot)$ of system (2.9) always stays close *enough* to $\bar{x}(\cdot)$ regardless of the initial state then $x(\cdot)$ is also safe and therefore $u(\cdot)$ is a safe schedule. More precisely, if there exists $\epsilon > 0$ such that for any initial state $x(0) = \bar{x}(0) \in \mathcal{X}_0$,

1. $\|x(t) - \bar{x}(t)\| < \epsilon$ for all $t \geq 0$; and

2. $\mathfrak{B}(\bar{x}^\star, \epsilon) \subseteq \texttt{Safe}$

then $x(\cdot)$ is safe. Here, the symbol $\|\cdot\|$ denotes both the Euclidean vector norm and the corresponding induced matrix norm. The notation $\mathfrak{B}(c, r)$ denotes the ball with center c and radius r, i.e., $\mathfrak{B}(c, r) := \{x : \|x - c\| \leq r\}$. Indeed, we will study the state error $(x(t) - \bar{x}(t))$ and will show that for any $\epsilon > 0$, there exists $\delta > 0$ and a δ-periodic control signal $u(\cdot)$ such that $x(t)$ is always ϵ-close to $\bar{x}(t)$ for all $t \geq 0$.

State Error

Let $\xi(t) = x(t) - \bar{x}(t)\ \forall t \geq 0$ be the state error between system (2.9) and its average system (2.10). Intuitively, as the period δ gets smaller, the trajectory $x(\cdot)$ gets closer to the trajectory $\bar{x}(\cdot)$, that is $\lim_{\delta \to 0} \|x(t) - \bar{x}(t)\| = \lim_{\delta \to 0} \|\xi(t)\| = 0$ for all $t \geq 0$. In what follows, we justify this intuition by directly calculating the state error $\xi(t)$ and deriving an upper-bound on its norm.

From the differential equations of the dynamical systems we obtain the dynamics of the error:

$$\dot{\xi}(t) = A\xi(t) + B\left(u(t) - \eta\right), \qquad \xi(0) = 0. \tag{2.11}$$

Note that the error $\xi(t)$ does not depend on the initial state $x(0)$. Its solution can be written explicitly as (Rugh, 1996)

$$\xi(t) = \int_0^t \mathrm{e}^{A(t-s)} B\left(u(s) - \eta\right)\, ds, \qquad t \geq 0. \tag{2.12}$$

Let $\sigma = \lfloor \frac{t}{\delta} \rfloor$, where $\lfloor c \rfloor$ denotes the largest integer not exceeding c. Then we have

$$\xi(t) = \sum_{i=0}^{\sigma-1} \int_{i\delta}^{(i+1)\delta} \mathrm{e}^{A(t-s)} B\left(u(s) - \eta\right)\, ds + \int_{\sigma\delta}^{t} \mathrm{e}^{A(t-s)} B\left(u(s) - \eta\right)\, ds \tag{2.13}$$

in which the sum disappears if $\sigma = 0$ (i.e., $t < \delta$). We transform the integral inside the sum

as

$$\int_{i\delta}^{(i+1)\delta} \mathrm{e}^{A(t-s)} B\left(u(s)-\eta\right)\, ds = \mathrm{e}^{A(t-(i+1)\delta} \int_{i\delta}^{(i+1)\delta} \mathrm{e}^{A((i+1)\delta-s)} B\left(u(s)-\eta\right)\, ds$$

and use the fact that $u(\cdot)$ is δ-periodic to obtain

$$\begin{aligned} &= \mathrm{e}^{A(t-(i+1)\delta)} \int_0^{\delta} \mathrm{e}^{A(\delta-s)} B\left(u(s)-\eta\right)\, ds \\ &= \mathrm{e}^{A(t-(i+1)\delta)} \xi_\delta \end{aligned}$$

in which $\xi_\delta = \xi(\delta) = \int_0^\delta \mathrm{e}^{A(\delta-s)} B\left(u(s)-\eta\right)\, ds$ is the state error after one period δ. Similarly, the tail integral in Equation (2.13) can be written as

$$\int_{\sigma\delta}^{t} \mathrm{e}^{A(t-s)} B\left(u(s)-\eta\right)\, ds = \int_0^{t-\sigma\delta} \mathrm{e}^{A((t-\sigma\delta)-s)} B\left(u(s)-\eta\right)\, ds = \xi(t-\sigma\delta)$$

where $\xi(t-\sigma\delta)$ is the state error at time instant $(t-\sigma\delta)$. Therefore,

$$\xi(t) = \left(\sum_{i=0}^{\sigma-1} \mathrm{e}^{A(t-(i+1)\delta)}\right) \xi_\delta + \xi(t-\sigma\delta). \tag{2.14}$$

The following Lemma gives us a *uniform* upper-bound on $\|\xi(t)\|$, *independent of time* t. Its proof can be found in Appendix A.1.5 on page 172.

Lemma 2.3 *There exist finite and positive constants α, β and γ, which are independent of the time period δ and the control signal $u(\cdot)$, such that*

$$\|\xi(t)\| \leq \frac{1}{2}\|A\|\gamma\beta^2 \frac{\delta^2}{1-\mathrm{e}^{-\alpha\delta}} + \gamma\beta\delta, \qquad \forall t \geq 0. \qquad \square$$

It is worth noting that in Lemma 2.3 the constants α and β depend only on the dynamics (2.9), in particular the matrices A and B, and the constant γ depends only on η. In addition, the obtained upper-bound on $\|\xi(t)\|$ holds for all time t.

Let $\|\xi(\cdot)\|_\infty = \sup_{t\geq 0}\|\xi(t)\|$ be the $\mathcal{L}_\infty$-norm of the signal $\xi(\cdot)$, that is the supremum of the

point-wise distance between $x(\cdot)$ and $\bar{x}(\cdot)$. Obviously $\|\xi(\cdot)\|_\infty$ is bounded above by the same upper-bound in Lemma 2.3. By simple calculations and the l'Hôpital's rule, we can show that this upper-bound goes to 0 as $\delta \to 0$. The following Lemma confirms the intuition that $x(\cdot)$ gets closer to $\bar{x}(\cdot)$ as δ gets smaller.

Lemma 2.4 *The state error between system* (2.9) *under δ-periodic control signals and its average system* (2.10) *vanishes as δ goes to* 0. *That is* $\lim_{\delta \to 0} \|\xi\|_\infty = 0$ *for all δ-periodic control signals $u(\cdot)$.* □

Lemma 2.4 implies that for any $\epsilon > 0$, there exists $\delta_\epsilon > 0$ such that for all $0 < \delta \leq \delta_\epsilon$, any δ-periodic control signal $u(\cdot)$ with utilization η will drive the trajectory $x(\cdot)$ to be ϵ-close to the trajectory $\bar{x}(\cdot)$ of the average system, regardless of the initial state $x(0)$.

Remark 2.2 It can be proved that Lemmas 2.3 and 2.4 still hold for control signals $u(\cdot)$ that are not δ-periodic but only satisfy the utilization constraint $\frac{1}{\delta}\int_{i\delta}^{(i+1)\delta} u(s)\,ds = \eta$, $\forall i \in \mathbb{N}$. This enables conventional real-time scheduling algorithms (Liu, 2000) to be used to schedule the system. However, to derive schedulability conditions, we only need to consider periodic control signals. □

Sufficient Schedulability Condition

Now that we have shown the state trajectory of the original system and its average system can be made arbitrarily close, we only need to construct a δ-periodic control signal $u(\cdot)$ such that $\|u(t)\|_1 \leq k\ \forall t \geq 0$ to complete the schedulability condition. Similarly to Theorem 2.2 on page 28, this construction is always possible if $\sum_{i=1}^m \eta_i \leq k$. Therefore, we arrive at a sufficient schedulability condition stated in the following Theorem, whose proof is given in Appendix A.1.6 on page 174.

Theorem 2.6 *Given a peak constraint k on the control signal $u(\cdot)$, $k \in \{0, 1, \ldots, n\}$. If there exists $\eta \in [0,1]^m$ such that*

1. $\|\eta\|_1 = \sum_{i=1}^m \eta_i \leq k$, *and*
2. $-A^{-1}(B_0 + B\eta) \in \text{int}(\texttt{Safe})$,

then the system (2.9) *is k-schedulable.* □

Necessary Schedulability Condition

The next Theorem provides a necessary condition for the system to be k-schedulable. Its proof can be found in Appendix A.1.7 on page 174.

Theorem 2.7 *Given a peak constraint k on the control signal $u(\cdot)$, $k \in \{0, 1, \ldots, n\}$. If the system* (2.9) *is k-schedulable then there exists $\eta \in [0,1]^m$ such that*

1. $\|\eta\|_1 = \sum_{i=1}^m \eta_i \leq k$, *and*
2. $-A^{-1}(B_0 + B\eta) \in \texttt{Safe}$. □

Observe that although the necessary schedulability condition in Theorem 2.7 seems to complement the sufficient schedulability condition in Theorem 2.6, there is a small gap between them: one uses the interior of `Safe` while the other uses the entire `Safe`. However, the difference ($\texttt{Safe} - \text{int}(\texttt{Safe})$) is often negligible in practice.

2.4.3. Feasible Peak Constraint

Given system (2.9), it is usually of interest to find a feasible peak constraint k because k is unknown at the beginning. From Theorems 2.6 and 2.7, the smallest feasible peak constraint $k_{\min}$ can be computed by solving the optimization

$$\begin{aligned} \underset{\eta}{\text{minimize}} \quad & \sum_{i=1}^m \eta_i = \mathbf{1}^T \eta && (2.15) \\ \text{subject to} \quad & \eta \in [0,1]^m \\ & -A^{-1}(B_0 + B\eta) \in \text{int}(\texttt{Safe}) && (2.15a) \end{aligned}$$

and letting $k_{\min} = \lceil \mathbf{1}^T \eta^\star \rceil$, in which $\eta^\star$ is an optimal solution of (2.15) and $\lceil c \rceil$ denotes the smallest integer not less than c. In practice, `Safe` is usually a hyper-rectangle or a polytope, for which the constraint (2.15a) becomes linear and the linear programming problem (2.15) can be solved efficiently (Boyd and Vandenberghe, 2006):

- If `Safe` is a hyper-rectangle $\texttt{Safe} := \{x : l \leq x \leq h\}$ where $l, h \in \mathbb{R}^n$ and $l < h$ then

$$(2.15a) \Leftrightarrow l + A^{-1}B_0 < -A^{-1}B\eta < h + A^{-1}B_0;$$

- If `Safe` is a polytope $\texttt{Safe} := \{x : Hx \leq K\}$ where $H \in \mathbb{R}^{q \times n}$, $K \in \mathbb{R}^q$ then

$$(2.15a) \Leftrightarrow -HA^{-1}B\eta < K + HA^{-1}B_0.$$

Any peak constraint k satisfying $k \geq k_{\min}$ will be feasible.

2.4.4. Systems with Outputs

Suppose that system (2.9) now has output $y = Cx \in \mathbb{R}^p$ where $C \in \mathbb{R}^{p \times n}$ and the safe set $\texttt{Safe} \subset \mathbb{R}^p$ is defined for the output instead of the state. It is straightforward to verify that the results in Theorems 2.6 and 2.7 still hold if $-A^{-1}(B_0 + B\eta)$ is replaced with $-CA^{-1}(B_0 + B\eta)$.

2.4.5. Illustrative Examples

We illustrate the schedulability conditions obtained above through the room-heater and the mass-spring-damper running examples.

Example 2.3 Consider the full room-heater model as in Equation (1.2) on page 6, with thermal interactions between rooms. However, we assume that there are no disturbances, that is the ambient air temperature is fixed at a known constant T_a and the heat gains are not present. This assumption allows us to write the dynamics of the system in the form in Equation (2.9) with $B_0 = W_a T_a$. In this example, we use the constant ambient air temperature $T_a = 5\,^{\circ}\text{C}$.

For the small-scale case with parameter values given in Table 1.1 on page 8, the linear program in Equation (2.15) gives an optimal solution $\eta^\star = [0.4939, 0.5109, 0.4875, 0.4760, 0.4525, 0.4510]^T$. Therefore the minimal feasible peak

constraint is $k_{\min} = 3$. The computation took 6.7 ms on average. For a large-scale system which has 1000 rooms and 1000 heaters with randomly generated parameters (cf. Section 1.2.1), MATLAB™ took an average of 6.4 s to calculate $k_{\min}$. Hence the computation is scalable. □

Example 2.4 In this example, we will verify the schedulability of the mass-spring-damper system without disturbances (see Section 1.2.2 on page 8 for the descriptions of this running example). We will also demonstrate how the safe set `Safe` affects the schedulability. Because there are no disturbances ($d(t) = 0$ for all $t \geq 0$), the state-space model of the system is

$$\dot{x}(t) = Ax(t) + B_0 + Bu(t), \qquad y(t) = Cx(t)$$

where A, B_0 and B are given in Equation (1.6), $y = [y_1, y_2]^T$ is the vector of positions, and $C = \left[\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right]$. Using the parameter values in Equation (1.8) on page 11 and the safe set $\texttt{Safe} = [0.75, 0.85] \times [1.15, 1.25]$, we solve the linear program (2.15) and obtain an optimal solution $\eta^\star = [0.15, 0.15]^T$. Therefore the system is schedulable with the specified safe set and the peak constraint $k = 1$.

Let us define the set $\mathcal{H} := \{y \in \mathbb{R}^2 : y = -CA^{-1}(B_0 + B\eta),\ \eta \in [0,1]^m,\ \|\eta\|_1 \leq 1\}$. It is not difficult to verify that $\mathcal{H}$ is a polytope whose vertices are

$$\left\{-CA^{-1}(B_0 + Bu) : u \in \left\{\left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right], \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right]\right\}\right\} = \left\{\begin{bmatrix} 1/9 \\ 2/9 \end{bmatrix}, \begin{bmatrix} 16/9 \\ 17/9 \end{bmatrix}, \begin{bmatrix} 2/3 \\ 4/3 \end{bmatrix}\right\}.$$

From Theorem 2.6, if $\mathcal{H}$ and $\text{int}(\texttt{Safe})$ intersect then the system is schedulable. On the other hand, from Theorem 2.7, if $\mathcal{H}$ and `Safe` are disjoint then the system is not schedulable. In Figure 2.5 on the next page, the polytope $\mathcal{H}$ is the gray-filled triangle while `Safe` is the square inside it. Obviously the system is schedulable. However, if `Safe` were the dashed square outside of $\mathcal{H}$, the system would be non-schedulable. □

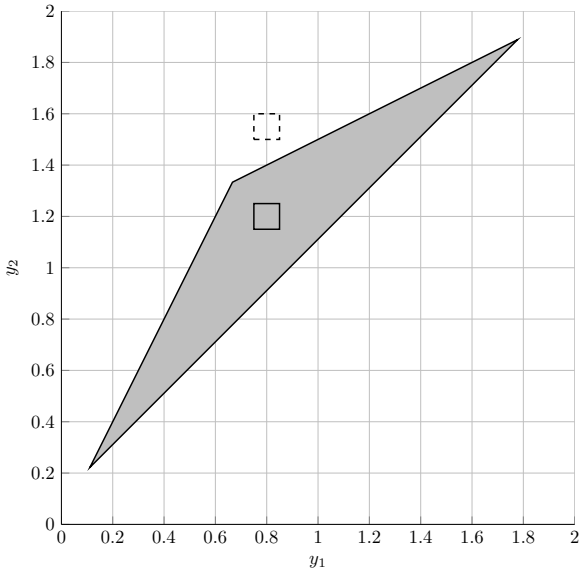


Figure 2.5: The set $\mathcal{H}$ and the safe set `Safe` in Example 2.4: $\mathcal{H}$ is the gray-filled triangle and the original `Safe` is the square inside it. When $\mathcal{H}$ and `Safe` are disjoint (e.g., if `Safe` is the dashed square) then the system is not schedulable.

2.4.6. Systems with Switching State Matrix

We have considered so far the affine dynamics (2.9) with constant state matrix A. For affine dynamics with switching state matrix

$$\dot{x}(t) = \left(A_0 + \sum_{i=1}^{m} A_i u_i(t) \right) x(t) + (B_0 + Bu(t)), \tag{2.16}$$

similar results can be obtained, as will be presented in this section.

Sufficient Schedulability Condition

To derive a sufficient schedulability condition, we also investigate periodic control signals for the system using its average system with respect to utilization vector η, which is defined by the differential equation

$$\dot{\bar{x}}(t) = \overline{A}_\eta \bar{x}(t) + \overline{B}_\eta \tag{2.17}$$

in which

$$\overline{A}_\eta = A_0 + \sum_{i=1}^{m} A_i \eta_i \quad \text{and} \quad \overline{B}_\eta = B_0 + B\eta.$$

Again, the average system is autonomous and starts from the same initial state: $\bar{x}(0) = x(0)$.

Suppose that $\overline{A}_\eta$ is Hurwitz. Similarly to Lemma 2.4, the trajectory $x(\cdot)$ of system (2.16) can be made arbitrarily close to the trajectory $\bar{x}(\cdot)$ of the average system by switching sufficiently fast. This result is stated in Lemma 2.5 and its proof is provided in Appendix A.1.8 on page 175. A similar result was proved in (Roberson and Stilwell, 2009).

Lemma 2.5 *Suppose that $\overline{A}_\eta$ is Hurwitz. For any $\epsilon > 0$, there exists $\delta_\epsilon > 0$ such that for all δ-periodic control signals $u(\cdot)$ with $0 < \delta \leq \delta_\epsilon$,*

$$\|x(t) - \bar{x}(t)\|_\infty \leq \epsilon \qquad \forall t \geq 0. \qquad \square$$

A sufficient schedulability condition, analogous to Theorem 2.6, can now be stated in Theorem 2.8. Its proof is omitted due to its similarity to the proof of Theorem 2.6.

Theorem 2.8 *Given a peak constraint k on the control signal $u(\cdot)$, $k \in \{0, 1, \ldots, n\}$. If there exists $\eta \in [0,1]^m$ such that*

1. $\|\eta\|_1 = \sum_{i=1}^{m} \eta_i \leq k$, *and*
2. $\overline{A}_\eta$ *is Hurwitz, and*
3. $-\overline{A}_\eta^{-1}(B_0 + B\eta) \in \text{int}(\texttt{Safe})$,

then the system (2.16) *is k-schedulable.* $\square$

Necessary Schedulability Condition

A necessary schedulability condition similar to Theorem 2.7, which complements the sufficient condition stated above, has not been verified and is unlikely. Indeed, Theorem 2.7 relies on the fact that the set $\{-A^{-1}(B_0 + B\eta) : \eta \in [0,1]^m, \|\eta\|_1 \leq k\}$ is convex and compact, and

if it is disjoint from `Safe` then there exists a strictly separating hyperplane between them (cf. Appendix A.1.7 on page 174). However, for system (2.16), the analogous set

$$\mathcal{P} := \{-\overline{A}_\eta^{-1}\left(B_0 + B\eta\right) \,:\, \eta \in [0,1]^m, \|\eta\|_1 \leq k\}$$

is not well-defined because $\overline{A}_\eta$ is not always invertible for all η. Even if $\mathcal{P}$ is well-defined, it is generally non-convex.

Nevertheless, to prove non-schedulability of system (2.16), we can seek for a certificate function $g : \mathcal{X} \to \mathbb{R}$ (cf. Section 2.2.2 and Theorem 2.1) using optimization techniques. For example, a linear certificate function of the form $g(x) = a^T x + b$ can be searched for by solving

$$\underset{a,b}{\text{minimize}} \quad \|a\|_1$$

subject to the constraints (ϵ is any small positive number)

$$\begin{aligned} & a^T x + b \geq 0 \quad \forall x \in \texttt{Safe} \\ & a^T \left(\left(A_0 + \sum_{i=1}^{m} A_i u_i \right) x + B_0 + Bu \right) \leq -\epsilon \quad \forall (x,u) \in \mathcal{X} \times \mathcal{U} \end{aligned}$$

which essentially formulate the conditions in Theorem 2.1 for g. Robust linear optimization problems of this form can be solved numerically by available software tools such as YALMIP (Löfberg, 2012). If an optimal solution exists then the system is non-schedulable. Another form of the certificate function are multivariate polynomials, for which sum-of-square (SOS) decomposition techniques (Parrilo, 2000) can be used to formulate the conditions of g as an SOS optimization problem. SOS programs are solvable by software tools such as SOSTOOLS (Prajna et al., 2005) and YALMIP (Löfberg, 2009).

Feasible Peak Constraint

Using Theorem 2.8, a feasible peak constraint k can be computed by solving the optimization

$$\begin{aligned} \underset{\eta}{\text{minimize}} \quad & \sum_{i=1}^{m} \eta_i = \mathbf{1}^T \eta \\ \text{subject to} \quad & \eta \in [0,1]^m && \text{(2.18a)} \\ & \overline{A}_\eta = A_0 + \sum_{i=1}^{m} A_i \eta_i \text{ is Hurwitz} && \text{(2.18b)} \\ & -\overline{A}_\eta^{-1} (B_0 + B\eta) \in \text{int}(\texttt{Safe}) && \text{(2.18c)} \end{aligned}$$

Unlike the similar optimization (2.15) on page 40 for the constant state matrix case, this optimization poses a significant challenge due to the stability constraint (2.18b) and the matrix inverse in the constraint (2.18c). These constraints make the problem nonlinear and non-convex and therefore difficult to solve. The stability constraint (2.18b) can be relaxed using either a linear approximation technique or a semidefinite approximation technique proposed by Zavlanos et al. (2011). Since systems with switching state matrix are not the focus of this work, we will skip the details and refer the reader to (Zavlanos et al., 2011) and the references therein.

2.5. General Affine Dynamics

In the previous sections, with the periodic scheduling approach, we have developed necessary and sufficient schedulability conditions for decoupled affine monotone dynamics (Section 2.3) and (coupled) affine dynamics without disturbances (Section 2.4). As mentioned in Section 2.2.1, being a feedforward scheduling strategy, the periodic scheduling approach neglects the influence of the disturbances and therefore is only applicable when the disturbances are absent or are small. To overcome this limitation, a feedback scheduling approach is employed in this section to derive more general schedulability results for affine systems in Equations (2.1) and (2.2) with disturbances. We will also show that the previous results are

special cases of these general results.

2.5.1. System's Dynamics

We consider first the general affine dynamical system in Equation (2.1) with constant state matrix (repeated here for convenience of reference):

$$\dot{x}(t) = Ax(t) + (B_0 + Bu(t)) + Wd(t) \tag{2.19}$$

As usual, $x \in \mathcal{X} \subseteq \mathbb{R}^n$ denotes the state vector, $u \in \mathcal{U} \subseteq \{0,1\}^m$ are the binary control inputs and $d \in \mathcal{D} \subset \mathbb{R}^q$ are the disturbances. Extended results for systems with switching state matrix and systems with outputs will be discussed in Sections 2.5.5 and 2.5.6 respectively.

The state matrix A is assumed to be Hurwitz. Both the disturbance set $\mathcal{D}$ and the safe set $\texttt{Safe} \subset \mathcal{X}$ are convex and compact. We will not restrict the control inputs u to the n-choose-k case, so the control input set $\mathcal{U}$ can be any subset of the finite set $\{0,1\}^m$.

2.5.2. Attracting Sets of Control Systems

The main analysis tool used for deriving schedulability conditions as well as scheduling algorithms for the general affine system is the concept of *attracting sets* of control systems. Consider a general nonlinear control system expressed as

$$\dot{x}(t) = f(t, x(t), u(t), d(t)), \qquad t \geq 0 \tag{2.20}$$

in which $x \in \mathbb{R}^n$, $u \in \mathcal{U}$ and $d \in \mathcal{D}$ have the usual interpretations. The sets $\mathcal{U}$ and $\mathcal{D}$ are compact. The function $f(\cdot,\cdot,\cdot,\cdot)$ satisfies the Lipschitz continuous assumption so that for an initial state $x(0)$, a control signal $u(\cdot)$ and a disturbance signal $d(\cdot)$, a trajectory $x(\cdot)$ of system (2.20) exists and is unique.

If regardless of the initial state and the disturbances, the state can always be controlled to converge to a subset $\mathcal{A}$ of the state space then $\mathcal{A}$ is called an *attracting set* (also called an

attractive set). In the context of this section, we only consider feedback control of the form $u(t) = \kappa(x(t))\ \forall t \geq 0$ that is robust to the disturbances. The concept of attracting sets of control systems is formally defined below.

Definition 2.4 *A set $\mathcal{A} \subset \mathbb{R}^n$ is an attracting set of control system* (2.20) *and a set $\mathcal{B} \subseteq \mathbb{R}^n$ is a basin (or domain) of attraction of $\mathcal{A}$ if there exists an admissible feedback control law $\kappa : \mathbb{R}^n \to \mathcal{U}$ such that for any initial state $x(0) \in \mathcal{B}$ and any admissible disturbance signal $d(\cdot)$, the resulted state trajectory $x(t)$ converges to $\mathcal{A}$ as $t \to +\infty$, that is*

$$\lim_{t \to +\infty} \text{dist}(x(t), \mathcal{A}) = 0$$

where $\text{dist}(x, \mathcal{A}) := \inf_{y \in \mathcal{A}} \|x - y\|$ *is the distance between point x and set $\mathcal{A}$.* □

Since in Green Scheduling we are interested in trajectories that enter the safe set in finite time, the definition of attracting sets is modified to take into account this requirement.

Definition 2.5 (Finite-time Attracting Sets) *A set $\mathcal{A} \subset \mathbb{R}^n$ is a finite-time attracting set of control system* (2.20) *and a set $\mathcal{B} \subseteq \mathbb{R}^n$ is a basin (or domain) of attraction of $\mathcal{A}$ if there exists an admissible feedback control law $\kappa : \mathbb{R}^n \to \mathcal{U}$ such that for any initial state $x(0) \in \mathcal{B}$ and any admissible disturbance signal $d(\cdot)$, the resulted state trajectory $x(\cdot)$ enters $\mathcal{A}$ in finite time and remains in $\mathcal{A}$ indefinitely, that is $x(t) \in \mathcal{A}$, $\forall t \geq \tau$ for some finite $\tau \geq 0$.* □

It is readily seen that Definition 2.5 is related to the notion of safety for the Green Scheduling problem (Definition 1.1) and the notion of schedulability (Definition 2.1). For brevity, we will use the term "attracting set" to mean "finite-time attracting set."

Attracting sets and their basins of attraction can be determined by means of the classical Lyapunov functions (see, e.g., Khalil, 1992; Grüne, 2002). The following Theorem provides a way to compute an attracting set, with respect to Definition 2.5, of control system (2.20).

Theorem 2.9 *Consider control system* (2.20). *Let $V : \mathbb{R}^n \to \mathbb{R}^+$ be a continuously differentiable function of state x, and let $\alpha > 0$ and $\gamma > 0$. Suppose that for all $x \in \mathbb{R}^n$ such that*

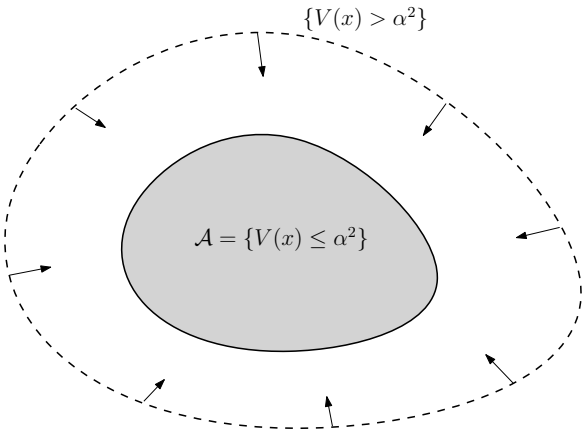


Figure 2.6: Intuition of Theorem 2.9 to determine an attracting set by a function V of state: on the phase space of the control system, the gray-filled set is $\mathcal{A}$, the α^2 sublevel set of V. Outside $\mathcal{A}$, there exists a control $u \in \mathcal{U}$ such that along the flow of the dynamical system, for all disturbances $d \in \mathcal{D}$, $V(x(t))$ is strictly decreasing (indicated by the arrows), hence $x(t)$ moves towards $\mathcal{A}$. After some finite time, $V(x(t))$ is guaranteed to reach α^2, thus $x(t)$ is inside $\mathcal{A}$.

$V(x) \geq \alpha^2$,

$$\inf_{u \in \mathcal{U}} \sup_{d \in \mathcal{D}} \nabla V(x) \cdot f(x, u, d) \leq -\gamma. \tag{2.21}$$

Then $\mathcal{A} := \{x \in \mathbb{R}^n \,:\, V(x) \leq \alpha^2\}$ *is an attracting set of the control system with basin of attraction* $\mathcal{B} := \mathbb{R}^n$. □

The proof of this result is given in Appendix A.1.9 on page 182.

The intuition of Theorem 2.9 is illustrated in the phase space of the control system in Figure 2.6. The set $\mathcal{A}$, i.e., the α^2 sublevel set of V, is filled in gray. The Lyapunov-like differential inequality (2.21) implies that for all $x \notin \mathcal{A}$, there exists a control $u \in \mathcal{U}$ that causes $V(x(t))$ to *strictly decrease* at a rate not less than γ along the flow of the dynamics, regardless of the disturbances $d \in \mathcal{D}$. Thus $x(t)$ can always be moved towards the level set $\{x \in \mathbb{R}^n \,:\, V(x) = \alpha^2\}$. After some finite time, $V(x(t))$ is guaranteed to reach α^2, hence $x(t)$ is in $\mathcal{A}$.

In the control theory literature, the function V in Theorem 2.9 is commonly known as a robust control Lyapunov functions (see Artstein, 1983; Freeman and Kokotovic, 1996; Liberzon et al., 2002).

Remark 2.3 The sublevel set $\mathcal{A}$ also has another property, which follows directly from Lemma A.5 on page 183: for any $x(0) \in \mathcal{A}$, there exists a feedback control law that keeps $x(t)$ inside $\mathcal{A}$, that is $x(t) \in \mathcal{A}\ \forall t \geq 0$, for all admissible disturbance signals $d(\cdot)$. This property is commonly known as *forward invariance*. □

2.5.3. Sufficient Schedulability Condition

Having discussed attracting sets of control systems in the previous section, we now return to the Green Scheduling problem of system (2.19) and apply this analysis tool to derive sufficient schedulability conditions. Essentially, if a subset of `Safe` is attracting then the system is schedulable because, according to Definition 2.5, there exists an admissible feedback control law that drives the system's state to this set, hence to `Safe`, in finite time. From Theorem 2.9, to characterize this attracting subset, we need to find a function $V : \mathbb{R}^n \to \mathbb{R}^+$ of state and a number α that satisfy the conditions of the Theorem.

It is well-known in linear system theory that a Lyapunov function for a linear time-invariant system is quadratic of the form $V(x) = x^T M x$, where $M \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix (Rugh, 1996). Because function V in Theorem 2.9 must satisfy the Lyapunov-like differential inequality (2.21), it is natural to seek a quadratic function V for system (2.19). A sublevel set of a quadratic function is an ellipsoid in the $\mathbb{R}^n$ vector space, thus we are searching for an attracting ellipsoidal subset of `Safe` (if one exists). Because `Safe` does not necessarily contain the origin, this ellipsoidal subset might not center at $\mathbf{0}$. Let $x_c \in \text{int}(\texttt{Safe})$ be the center of this ellipsoid. Then the quadratic function V is of the form

$$V(x) := (x - x_c)^T M (x - x_c)$$

in which $M \in \mathbb{R}^{n \times n}$ is positive semidefinite, denoted $M \succeq 0$. The attracting ellipsoid we are looking for is thus given by $\mathcal{A} := \left\{ x \in \mathbb{R}^n \,:\, V(x) = (x - x_c)^T M (x - x_c) \leq \alpha^2 \right\}$. Differential

inequality (2.21) reads, for all $x \in \mathbb{R}^n$ such that $V(x) \geq \alpha^2$,

$$\begin{aligned}
&\inf_{u\in\mathcal{U}} \sup_{d\in\mathcal{D}} 2\left(x - x_c\right)^T M \left(Ax + B_0 + Bu + Wd\right) \\
=&2\left(x - x_c\right)^T MA\left(x - x_c\right) + 2 \inf_{u\in\mathcal{U}} \sup_{d\in\mathcal{D}} \left(x - x_c\right)^T M \left(Ax_c + B_0 + Bu + Wd\right) \\
\leq& -\gamma. \qquad (2.22)
\end{aligned}$$

As with Lyapunov functions, M is not any positive semidefinite matrix but must also satisfy the Lyapunov inequality with respect to the dynamics (2.19) (Rugh, 1996): $A^T M + MA \preceq 0$. Moreover, it is desirable to obtain the maximum decay rate of V so that we can bound $V(x(t))$ as in Equation (2.21). To this end, we consider the generalized eigenvalue problem (GEVP)

$$\begin{aligned}
\underset{\lambda, M}{\text{maximize}} \quad & \lambda \\
\text{subject to} \quad & M \succeq 0 \\
& A^T M + MA \preceq -2\lambda M
\end{aligned}$$

in which λ, being the decay rate of $V(x(t))$, is maximized. Suppose that the above GEVP has an optimal solution with $\lambda > 0$. We then have

$$2\left(x - x_c\right)^T MA\left(x - x_c\right) \leq -2\lambda \left(x - x_c\right)^T M \left(x - x_c\right) \qquad \forall x \in \mathbb{R}^n.$$

It follows that if for all $x \in \mathbb{R}^n$ such that $V(x) \geq \alpha^2$,

$$-2\lambda \left(x - x_c\right)^T M \left(x - x_c\right) + 2 \inf_{u\in\mathcal{U}} \sup_{d\in\mathcal{D}} \left(x - x_c\right)^T M \left(Ax_c + B_0 + Bu + Wd\right) \leq -\gamma \quad (2.23)$$

then the inequality (2.22) is verified. In particular, if the static game in Equation (2.23) (i.e., the inf-sup term on the left-hand side) is non-positive for all x then the inequality is satisfied. We observe that in Equation (2.23), the optimization variables u and d are

decoupled, therefore

$$\begin{aligned}
&\inf_{u\in\mathcal{U}}\sup_{d\in\mathcal{D}}(x-x_c)^T M\left(Ax_c+B_0+Bu+Wd\right)\\
=&(x-x_c)^T M\left(Ax_c+B_0\right)+\inf_{u\in\mathcal{U}}(x-x_c)^T MBu+\sup_{d\in\mathcal{D}}(x-x_c)^T MWd\\
=&\sup_{d\in\mathcal{D}}\inf_{u\in\mathcal{U}}(x-x_c)^T M\left(Ax_c+B_0+Bu+Wd\right).
\end{aligned}$$

Because $\mathcal{U}$ is finite, the inner infimum is equivalent to the minimization

$$\min_{u\in\mathcal{U}}(x-x_c)^T M\left(Ax_c+B_0+Bu+Wd\right).$$

Let $\mathrm{co}(\mathcal{U})$ denote the convex hull of $\mathcal{U}$ in the continuous space $\mathbb{R}^m$. Then this minimization has the same optimal value as $\min_{u\in\mathrm{co}(\mathcal{U})}(x-x_c)^T M\left(Ax_c+B_0+Bu+Wd\right)$. Indeed, assume otherwise then there exists $u^\star\in\mathrm{co}(\mathcal{U})$ such that

$$(x-x_c)^T M\left(Ax_c+B_0+Bu^\star+Wd\right)<\min_{u\in\mathcal{U}}(x-x_c)^T M\left(Ax_c+B_0+Bu+Wd\right).$$

From the definition of $\mathrm{co}(\mathcal{U})$ (Boyd and Vandenberghe, 2006, sec. 2.1.4), $u^\star=\theta_1u^{(1)}+\cdots+\theta_ku^{(k)}$ where $u^{(i)}\in\mathcal{U}$ and $\theta_i\geq 0$ for $i=1,\ldots,k$, and $\theta_1+\cdots+\theta_k=1$. It follows that

$$\begin{aligned}
(x-x_c)^T M\left(Ax_c+B_0+Bu^\star+Wd\right)&=\sum_{i=1}^k\theta_i(x-x_c)^T M\left(Ax_c+B_0+Bu^{(i)}+Wd\right)\\
&\geq\sum_{i=1}^k\theta_i\min_{u\in\mathcal{U}}(x-x_c)^T M\left(Ax_c+B_0+Bu+Wd\right)\\
&=\min_{u\in\mathcal{U}}(x-x_c)^T M\left(Ax_c+B_0+Bu+Wd\right),
\end{aligned}$$

a contradiction. Therefore Equation (2.23) is equivalent to

$$-2\lambda(x-x_c)^T M(x-x_c)+2\sup_{d\in\mathcal{D}}\min_{u\in\mathrm{co}(\mathcal{U})}(x-x_c)^T M\left(Ax_c+B_0+Bu+Wd\right)\leq-\gamma. \tag{2.24}$$

We observe that $\sup_{d\in\mathcal{D}}\min_{u\in\mathrm{co}(\mathcal{U})}(x-x_c)^T M\left(Ax_c+B_0+Bu+Wd\right)\leq 0\ \forall x\in\mathbb{R}^n$ im-

plies the inequality (2.22) being satisfied. This leads to the following result on attracting sets of control system (2.19). Its proof is given in Appendix A.1.10 on page 186.

Lemma 2.6 *Suppose there exist* $M \in \mathbb{R}^{n\times n}$ *and* $\lambda > 0$ *such that*

$$M \succeq 0, \tag{2.25a}$$

$$A^T M + MA \preceq -2\lambda M. \tag{2.25b}$$

Furthermore, there exists $x_c \in \mathbb{R}^n$ *satisfying for all* $d \in \mathcal{D}$*, there exists* $u \in \text{co}(\mathcal{U})$ *such that* $Ax_c + B_0 + Bu + Wd = 0$*. Then for any* $\alpha > 0$*,* $\mathcal{A} := \{x \in \mathbb{R}^n \,:\, (x - x_c)^T M (x - x_c) \leq \alpha^2\}$ *is an attracting set of control system* (2.19) *with basin of attraction* $\mathcal{B} := \mathbb{R}^n$*.* □

The validity of Lemma 2.6 depends on the existence of M and λ. Recall that the state matrix A is assumed to be Hurwitz. Similar to the existence of Lyapunov functions for asymptotically stable linear systems (Rugh, 1996), there always exist M and λ satisfying the hypothesis of Lemma 2.6. This result is confirmed in Proposition 2.1, whose proof can be found in Appendix A.1.11 on page 186.

Proposition 2.1 *If the state matrix* A *is Hurwitz then there exist* $M \in \mathbb{R}^{n\times n}$ *and* $\lambda > 0$ *that satisfy the conditions in Lemma 2.6.* □

As mentioned at the beginning of this section, if `Safe` contains an attracting subset then the system is schedulable. This is illustrated on the phase space of control system (2.19) in Figure 2.7 on the next page. The gray-filled ellipsoid in this figure is the attracting subset $\mathcal{A} := \{x \in \mathbb{R}^n \,:\, (x - x_c)^T M (x - x_c) \leq \alpha^2\}$ of `Safe`. Outside this subset, there exists a control input $u \in \mathcal{U}$ such that $x(t)$ is always attracted towards $\mathcal{A}$ (indicated by the arrows) for all disturbances $d \in \mathcal{D}$, thus it is safe. Therefore the system is schedulable. Using Lemma 2.6, the following sufficient schedulability condition is straightforward, hence its proof is omitted.

Proposition 2.2 *If there exists* $x_c \in \text{int}(\texttt{Safe})$ *such that*

$$\forall d \in \mathcal{D},\ \exists u \in \text{co}(\mathcal{U}) : Ax_c + B_0 + Bu + Wd = 0$$

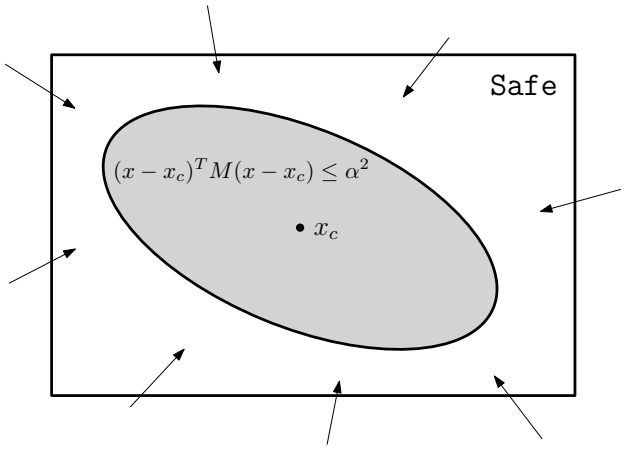


Figure 2.7: Illustration of an attracting subset of `Safe` for sufficient schedulability condition: the gray-filled subset $\{x \in \mathbb{R}^n : (x - x_c)^T M (x - x_c) \leq \alpha^2\}$ of `Safe` is attracting. Outside this subset, there exists a control $u \in \mathcal{U}$ that drives $x(t)$ towards the subset (indicated by the arrows) for all disturbances $d \in \mathcal{D}$, hence making it safe.

then the system (2.19) *is schedulable.* □

Proposition 2.2 requires us to explicitly find a point $x_c \in \text{int}(\texttt{Safe})$ that satisfies its condition. However, for determining whether a system is schedulable, it is sufficient to verify the existence of such a point. The sufficient schedulability condition stated in Theorem 2.10 below can be checked by performing standard geometric operations on the sets `Safe`, $\mathcal{D}$ and $\mathcal{U}$. First, let us introduce several geometric operations on sets.

- The negation[1] of a set $X \subseteq \mathbb{R}^n$

$$-X := \{-x \in \mathbb{R}^n : x \in X\}.$$

- The sum of a set $X \subseteq \mathbb{R}^n$ and a vector $v \in \mathbb{R}^n$

$$X + v := \{x + v \in \mathbb{R}^n : x \in X\}.$$

[1]Do not confuse this with set difference $X \setminus Y := \{x \in X : x \notin Y\}$ and set complement $X^C := \mathbb{R}^n \setminus X$.

- The product of a matrix $A \in \mathbb{R}^{m\times n}$ and a set $X \subseteq \mathbb{R}^n$ is a set in $\mathbb{R}^m$

$$AX := \{Ax \in \mathbb{R}^m \,:\, x \in X\}.$$

- The Pontryagin difference of two sets $X, Y \subseteq \mathbb{R}^n$

$$X \ominus Y := \{z \in \mathbb{R}^n \,:\, z + y \in X \;\forall y \in Y\} = \{z \in \mathbb{R}^n \,:\, Y + z \subseteq X\}.$$

- The Minkowski sum of two sets $X, Y \subset \mathbb{R}^n$

$$X \oplus Y := \{x + y \in \mathbb{R}^n \,:\, x \in X, y \in Y\}.$$

Using these set operations, the condition in Proposition 2.2 is equivalent to

$$\exists x_c \in \operatorname{int}(\texttt{Safe}) \,:\, Ax_c + B_0 + W\mathcal{D} \subseteq -B\operatorname{co}(\mathcal{U}). \tag{2.26}$$

We can now state the main sufficient schedulability condition for system (2.19).

Theorem 2.10 *If* $\operatorname{int}(\texttt{Safe}) \cap Q \neq \emptyset$*, where* $Q := -A^{-1}B\operatorname{co}(\mathcal{U}) \ominus A^{-1}(B_0 + W\mathcal{D})$*, then the system* (2.19) *is schedulable.* □

Proof See Appendix A.1.12 on page 186. ■

Checking the schedulability condition in Theorem 2.10 does not require solving the GEVP (Equations (2.25a) and (2.25b)) to compute the matrix M and the value λ. In practice, the sets $\operatorname{co}(\mathcal{U})$, `Safe` and $\mathcal{D}$ are usually hyper-rectangles or polytopes. Therefore, checking the schedulability condition in Theorem 2.10 involves only standard operations on polytopic sets, which can be computed numerically with readily available scientific software, for instance the Multi-Parametric Toolbox for MATLAB™ (Kvasnica et al., 2004) and the Parma Polyhedra Library (BUGSENG, 2012). For the n-choose-k case, i.e., $\mathcal{U} = \{u \in \{0,1\}^m \,:\, \|u\|_1 \leq k\}$ where k is the peak constraint imposed on u, it is straightforward to verify that $\operatorname{co}(\mathcal{U}) =$

$\{u \in [0,1]^m : \|u\|_1 \leq k\}$ – a polytope.

Remark 2.4 A minor catch in Theorem 2.10 is that the interior operator ($\text{int}(\texttt{Safe})$) might not be supported in some scientific computation software. In that case, we can replace ($\text{int}(\texttt{Safe}) \cap Q$) with ($\texttt{Safe} \cap Q$). It is possible to show that $\text{int}(\texttt{Safe} \cap Q) \subseteq \text{int}(\texttt{Safe}) \cap Q$.□

2.5.4. Necessary Schedulability Condition

The sufficient schedulability conditions developed in the previous section essentially means that at any state x outside the subset $\mathcal{A}$ of $\texttt{Safe}$, there exists a control u that drives the state to $\mathcal{A}$ regardless of the disturbance d. For necessary schedulability conditions, we switch the roles of u and d and do the opposite (cf. Theorem 2.1 on page 24): at any state x in the safe set and for any control $u \in \mathcal{U}$, there exists a disturbance d that drives the state outside the safe set. Based on this idea, a necessary schedulability condition can be stated in the following Theorem. Its proof can be found in Appendix A.1.13 on page 187.

Theorem 2.11 *Consider the control system* (2.19). *If there exists* $d^\star \in \mathcal{D}$ *such that* $(B_0 + B\,\text{co}(\mathcal{U}) + Wd^\star) \cap (-A\texttt{Safe}) = \emptyset$ *then the system is non-schedulable. In particular, if* $-W\mathcal{D} \not\subseteq (A\texttt{Safe} \oplus B\,\text{co}(\mathcal{U})) + B_0$ *then the system is non-schedulable.* □

Example 2.5 We continue the room-heater running example. The new schedulability results allow us to consider the full system's model as described in Section 1.2.1 on page 5, with disturbances. Suppose that the ambient air temperature T_a can vary between $T_{a,\min} = 2\,^\circ\text{C}$ and $T_{a,\max} = 12\,^\circ\text{C}$ while the heat gain $Q_{g,i}$ in each room $i = 1, \ldots, n$ is bounded in $[0, 0.5]$kW. Therefore $\mathcal{D} = [0, 0.5] \times \cdots \times [0, 0.5] \times [2, 12]$. The desired room air temperatures are from $20\,^\circ\text{C}$ to $24\,^\circ\text{C}$ for each room.

Let us consider the small-scale system with 6 rooms and 6 heaters whose parameters are given in Section 1.2.1. The constraint on u is the n-choose-k case with peak constraint $k = 4$, i.e., at any time, at most 4 heaters can be activated simultaneously. Using Theorem 2.10, we can compute the set $Q = -A^{-1}B\,\text{co}(\mathcal{U}) \ominus A^{-1}(B_0 + W\mathcal{D})$ as a polytope of the form

$Q = \{x \in \mathbb{R}^n : Hx \leq K\}$ where

$$H = \begin{bmatrix} 0.9056 & -0.2002 & 0.0000 & 0.0000 & 0.0000 & -0.3739 \\ -0.2764 & 0.9306 & -0.2399 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.2130 & 0.8983 & -0.3843 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.3828 & 0.8969 & -0.2214 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -0.2385 & 0.9009 & -0.3627 \\ -0.3768 & 0.0000 & 0.0000 & 0.0000 & -0.2779 & 0.8836 \\ -0.9056 & 0.2002 & 0.0000 & 0.0000 & 0.0000 & 0.3739 \\ 0.2764 & -0.9306 & 0.2399 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.2130 & -0.8983 & 0.3843 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.3828 & -0.8969 & 0.2214 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.2385 & -0.9009 & 0.3627 \\ 0.3768 & 0.0000 & 0.0000 & 0.0000 & 0.2779 & -0.8836 \\ 0.2207 & 0.4188 & 0.4459 & 0.3954 & 0.3758 & 0.5288 \end{bmatrix}, \quad K = \begin{bmatrix} 10.7340 \\ 12.9962 \\ 9.8627 \\ 9.8097 \\ 10.5358 \\ 8.0708 \\ -4.6084 \\ -5.8418 \\ -4.3834 \\ -4.2813 \\ -4.4250 \\ -3.3812 \\ 54.6061 \end{bmatrix}.$$

We can verify that $\text{int}(\texttt{Safe}) \cap Q \neq \emptyset$, hence the system is k-schedulable with $k = 4$. However, if we set $k = 3$ then we can verify that the condition of Theorem 2.11 is satisfied; therefore the system is non-schedulable. Each computation took about 21 ms to complete on MATLAB™.

We then scale the system to 100 rooms and 100 heaters, with randomly generated parameters (see Section 1.2.1). The peak constraint is set to $k = 60$. Again, we can verify that the system is k-schedulable by Theorem 2.10. The computation took 29 s on MATLAB™. □

2.5.5. Systems with Outputs

In this subsection, we extend the schedulability results to the same affine system (2.19) but with output $y \in \mathbb{R}^p$ defined as $y = Cx$ where $C \in \mathbb{R}^{p \times n}$ is the output matrix. The safe set is defined for the output instead of the state, i.e., $\texttt{Safe} \subset \mathbb{R}^p$. The definitions of safety and schedulability are also modified accordingly.

For sufficient schedulability conditions, the same approach using attracting sets can be applied: we are looking for an attracting set $\mathcal{A} := \{x \in \mathbb{R}^n : V(x) \leq \alpha^2\}$ such that $C\mathcal{A} \subseteq \texttt{Safe}$. The following sufficient schedulability result is analogous to Theorem 2.10 and its proof is thus omitted for brevity.

Theorem 2.12 *If* $C\left(-A^{-1}B\,\text{co}(\mathcal{U}) \ominus A^{-1}(B_0 + W\mathcal{D})\right) \cap \text{int}(\texttt{Safe}) \neq \emptyset$ *then the system is schedulable.* □

Similarly, a necessary schedulability condition analogous to Theorem 2.11 can be stated.

Theorem 2.13 *If there exists* $d^\star \in \mathcal{D}$ *such that the set* $-(B_0 + B\,\text{co}(\mathcal{U}) + Wd^\star)$ *and the*

set $\mathcal{Q} := \{Ax \in \mathbb{R}^p : x \in \mathbb{R}^n, Cx \in \texttt{Safe}\}$ *are disjoint then the system is non-schedulable. In particular, if* $-W\mathcal{D} \not\subseteq (\mathcal{Q} \oplus B\,\mathrm{co}(\mathcal{U})) + B_0$ *then the system is non-schedulable.* □

PROOF See Appendix A.1.14 on page 188. ■

2.5.6. Systems with Switching State Matrix

This subsection extends the schedulability results to affine systems with switching state matrix

$$\dot{x}(t) = \left(A_0 + \sum_{i=1}^{m} A_i u_i(t)\right) x(t) + B_0 + Bu(t) + Wd(t). \tag{2.27}$$

Define a map $A : [0,1]^m \to \mathbb{R}^{n\times n}$ as[2] $A(u) := A_0 + \sum_{i=1}^m A_i u_i(t)$. We note that: (a) the domain of $A(\cdot)$ is the continuous set $[0,1]^m$, not the finite set $\{0,1\}^m$ of binary vectors; (b) $A(u)$ is affine in u. The following Theorem, similar to Proposition 2.2 and Theorem 2.10, provides a sufficient schedulability condition for this system.

Theorem 2.14 *Consider system* (2.27). *Suppose there exist* $M \in \mathbb{R}^{n\times n}$ *and* $\lambda > 0$ *such that*

$$M \succeq 0, \tag{2.28a}$$

$$A(u)^T M + MA(u) \preceq -2\lambda M, \qquad \forall u \in \mathcal{U}. \tag{2.28b}$$

The following statements are equivalent:

1. *There exists* $x_c \in \mathrm{int}(\texttt{Safe})$ *such that for all* $d \in \mathcal{D}$, *there exists* $u \in \mathrm{co}(\mathcal{U})$ *such that* $A(u)x_c + B_0 + Bu + Wd = 0$
2. *There exists* $x_c \in \mathrm{int}(\texttt{Safe})$ *such that* $-W\mathcal{D} \subseteq \mathrm{co}(\{A(u)x_c + B_0 + Bu : u \in \mathcal{U}\})$.

If any of the above statements holds then the system is schedulable. □

The proof of Theorem 2.14 is omitted due to its similarity with those of Proposition 2.2 and Theorem 2.10. Note that in this case, we do not have a compact geometric schedulability condition as in Theorem 2.10 because x_c is multiplied by the switching state matrix $A(u)$.

[2] Although using the same letter A, the map $A(\cdot)$ is not to be confused with the state matrix A because there is no such matrix for this system.

Furthermore, even though the values of M and λ are not used in the conditions 1 and 2, it is important to verify that they exist because, unlike system (2.19) with Hurwitz state matrix A, their existence is not guaranteed. M and λ can be computed by solving the GEVP consisting of Equations (2.28a) and (2.28b).

Necessary schedulability conditions can be attained by optimization techniques as presented in Section 2.4.6. However, Theorem 2.15 provides a simple necessary condition similar to Theorem 2.11, without having to solve any optimization problem.

Theorem 2.15 *Consider system* (2.27). *If* $-W\mathcal{D} \not\subseteq \text{co}(\{A(u)\texttt{Safe} + Bu + B_0 : u \in \mathcal{U})$ *then the system is non-schedulable.* □

For a proof, see Appendix A.1.15 on page 188.

2.5.7. Comparison with Previous Results

It is possible to show that the schedulability results in the previous Sections 2.3 and 2.4, obtained for simpler system's models and without disturbances, are special cases of the schedulability results obtained in this section for the general system's model with disturbances. Take Theorem 2.6 on page 39 for example and compare it with Theorem 2.10 on page 55. When the disturbances are absent, the set Q in Theorem 2.10 is reduced to $Q = -A^{-1}B\,\text{co}(\mathcal{U}) \ominus A^{-1}B_0 = -A^{-1}(B_0 + B\,\text{co}(\mathcal{U}))$. Also note that in the n-choose-k case, $\text{co}(\mathcal{U})$ is the set $\text{co}(\mathcal{U}) = \{\eta \in [0,1]^m : \|\eta\|_1 \leq k\}$. Therefore, the condition $\text{int}(\texttt{Safe}) \cap Q \neq \emptyset$ is equivalent to $\exists \eta \in \text{co}(\mathcal{U})$: $-A^{-1}(B_0 + B\eta) \in \text{int}(\texttt{Safe})$, which is exactly the condition in Theorem 2.6.

As we mentioned in Section 2.2, the schedulability results obtained in this section are general and can be applied to the systems in the previous sections. However, we still developed and presented these special-case results because they are based on the periodic scheduling approach and they establish the theoretical foundations for the periodic scheduling synthesis to be presented in Chapter 3.

2.6. Conclusions

In this chapter, we have studied the schedulability analysis problem for Green Scheduling. In particular, we have developed sufficient and necessary schedulability conditions for a wide range of types of systems, from decoupled affine monotone dynamics in Section 2.3 to general affine systems in Section 2.5. Two different approaches were used for schedulability analysis: the periodic scheduling approach for systems without disturbances and the feedback scheduling approach for those with constrained disturbances.

Several schedulability results obtained in this chapter share similarities with the classical schedulability analysis in real-time scheduling (Liu, 2000; Buttazzo, 2011). In particular, we also used the notion of *utilization* to capture the timing constraint on the actuation of each control input so that the system is safe. Schedulability analysis can then be performed using these values as in conventional real-time scheduling. Thus, our contribution in periodic scheduling is the development of mapping from system's dynamics and safety specifications to utilization values. Recently, there have been several attempts to apply real-time scheduling algorithms to scheduling of electric loads for peak demand reduction (Facchinetti et al., 2010; Facchinetti and Vedova, 2011; Subramanian et al., 2012). However, they focused on the scheduling synthesis aspect and did not derive comprehensive schedulability analysis methods as we did in this chapter. In addition, they were mostly limited to simple dynamics of decoupled systems with no interactions and no disturbances. This limitation of these results highlights another major contribution of this chapter: the schedulability conditions for general affine systems with constrained disturbances. We employed the feedback scheduling approach to derive compact geometric conditions for verifying schedulability of such systems.

The results developed in this chapter not only allow deciding whether a system is schedulable but also establish the theoretical framework for the scheduling synthesis in the chapters that follow. In particular, periodic scheduling strategies will be studied in Chapter 3 and feedback scheduling strategies will be presented in Chapter 4.

Chapter 3

Periodic Green Scheduling

Periodic scheduling (or time-division scheduling) refers to a simple scheduling strategy where the schedule is repeated every $\delta > 0$ time units. The duration δ is called the time period of the periodic schedule. As discussed in Section 1.3.2, periodic scheduling belongs to the class of feedforward scheduling strategies in which the schedules are pre-determined and are executed on the system as time progresses without taking into account the state of the system nor the influence of the disturbances. Because of the lack of feedback, these scheduling strategies are not suitable for systems with significant exogenous disturbances. However, for systems where disturbances are not present or are insignificant, feedforward scheduling might attain acceptable performance while being simple in implementation. For example, in the room-heater example (Section 1.2.1 on page 5), if the building envelope[1] is tight so that the ambient air temperature only has very small effect on the inside air temperature and does not vary significantly, then periodic scheduling can be used.

This chapter studies periodic scheduling policies for the Green Scheduling problem, based on the mathematical analysis in Sections 2.3 and 2.4. We will only consider systems with no disturbances. If disturbances are present but insignificant, they can usually be either ignored or replaced by nominal constant disturbance values. For each type of system's dynamics, we will first investigate its trajectories under periodic scheduling, then we will present a method to synthesize safe periodic schedules for the system given that it is schedulable. Section 3.1 mathematically formulates the periodic schedules that will be considered in this chapter. Then periodic scheduling synthesis for decoupled affine monotone dynamics and for general affine dynamics will be presented in Sections 3.2 and 3.3 respectively. The sequencing problem for periodic scheduling synthesis will be discussed in Section 3.4. Section 3.5 concludes the

[1] The building envelope is the physical separator between the interior and the exterior environments of a building.

chapter with a discussion on related work.

The results presented in this chapter are expanded from our published work (Nghiem et al., 2011a, 2012a,b).

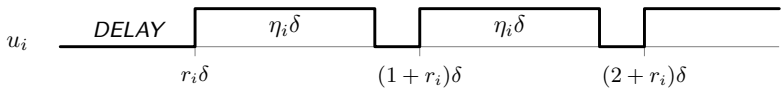
3.1. Periodic Scheduling Formulation

To study the behavior of a dynamical system under periodic scheduling, we first formulate a mathematical representation of the periodic schedules to be considered in this chapter. Let $u_i : \mathbb{R}^+ \to \{0, 1\}$ be a periodic scalar schedule with time period $\delta > 0$. In practical systems, the actuators usually have physical constraints that prohibit them from switching too fast. Furthermore, high-frequency switching often causes performance degradation of, or even damages to, the actuators. Therefore it is usually desirable, sometimes imperative, to minimize or at least reduce the switching frequency of the signals. For that reason, we restrict $u_i(\cdot)$ to switching only once during each time period. We can also allow for a finite delay (or *offset*), which is shorter than a time period, at the beginning without changing the overall periodic behavior of the system. Hence we consider periodic schedules of the form

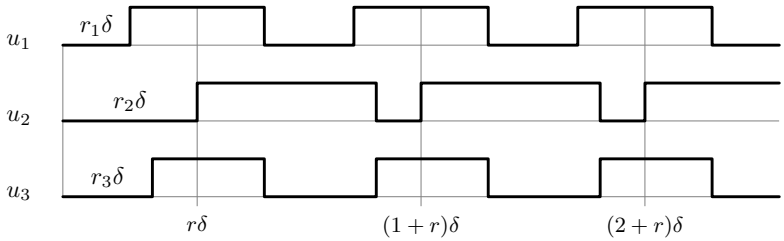
$$u_i(t) = \begin{cases} 1 & \text{if } (r_i + j)\,\delta \leq t < (r_i + j + \eta_i)\,\delta,\ j \in \mathbb{N} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

in which $t \geq 0$, $0 \leq \eta_i \leq 1$ and $0 \leq r_i < 1$. In this formulation, r_i specifies the optional initial delay of the signal. It is evident that $u_i(\cdot)$ is periodic after this delay, i.e., $u_i(t) = u_i(t + \delta)$ for all $t \geq r_i\delta$. As introduced in Section 2.3.1, the value $\eta_i \in [0, 1]$ is the fraction of time in a period when u_i is 1. It is termed the *utilization* of periodic schedule u_i, following the convention in real-time scheduling (Liu, 2000). We can verify that $\frac{1}{\delta}\int_t^{t+\delta} u_i(s)\,ds = \eta_i$ for all $t \geq r_i\delta$. Therefore, each periodic schedule $u_i(\cdot)$ is parameterized by three parameters: (δ, η_i, r_i). The timing diagram in Figure 3.1a on the next page illustrates such a periodic schedule where the notions of time period δ, utilization η_i and delay r_i are clearly indicated.

The aggregated schedule $u(\cdot)$ consists of multiple scalar schedules $u_1(\cdot), \ldots, u_m(\cdot)$, each



(a) A periodic scalar schedule $u_i(\cdot)$ with time period δ and delay $r_i\delta$. During each time period, u_i switches only once.



(b) Multiple periodic scalar schedules with a common time period δ and different delays. The aggregated schedule $u(\cdot)$ is periodic after the maximum delay $r\delta$, where $r = \max(r_1, r_2, r_3)$.

Figure 3.1: Illustration of the periodic scheduling formulation.

having the form in Equation (3.1). We assume that they all have the same time period $\delta > 0$. This assumption will simplify the system analysis and scheduling synthesis to be presented later in this chapter, yet being reasonable because this is usually the case in practice. It follows that the schedule $u(\cdot) = [u_1(\cdot), \ldots, u_m(\cdot)]^T$ is also periodic with the same time period δ and a delay equal to the maximum delay of the component signals, that is $r\delta$ where $r = \max(r_1, \ldots, r_m)$. In Figure 3.1b on this page, an example of the aggregation of three periodic scalar schedules is depicted. The aggregated signal is therefore specified by the set of parameters

$$\delta, (\eta_1, r_1), \ldots, (\eta_m, r_m)$$

where δ is the common time period and (η_i, r_i) denote the utilization and delay of the individual component signals. In the periodic scheduling synthesis in the subsequent subsections, we will present methods to determine these parameters.

3.2. Periodic Scheduling for Decoupled Affine Monotone Dynamics

In this section, we consider the decoupled affine monotone dynamics as defined in Section 2.3. Recall that the dynamics of the state variables x_i, $i = 1, \ldots, n$, are decoupled from each other and are monotone of the form in Equation (2.4), reproduced in the following equation:

$$\dot{x_i}(t) = \begin{cases} -a_{\text{on},i} x_i(t) + b_{\text{on},i} & \text{if } u_i(t) = 1 \\ -a_{\text{off},i} x_i(t) + b_{\text{off},i} & \text{if } u_i(t) = 0 \end{cases} \tag{3.2}$$

where $a_{\text{on},i} > 0$, $a_{\text{off},i} > 0$, $b_{\text{on},i}$ and $b_{\text{off},i}$ are parameters. Note that each x_i is controlled only by input u_i. We also assumed in Assumption 2.1 that:

- There are no disturbances ($d \equiv 0$);
- The safe set is a hyper-rectangle in $\mathbb{R}^n$: $\texttt{Safe} = [l_1, h_1] \times \cdots \times [l_n, h_n]$, where $l_i < h_i$ are the desired lower and upper bounds for each state variable x_i;
- The dynamics of x_i is monotone within its bound $[l_i, h_i]$, i.e., x_i always grows when $u_i = 1$ and decays when $u_i = 0$, or mathematically

$$-a_{\text{off},i} x_i + b_{\text{off},i} < 0 < -a_{\text{on},i} x_i + b_{\text{on},i} \quad \forall x_i \in [l_i, h_i]. \tag{3.3}$$

Because of the decoupling of the state variables in terms of dynamics and controls, we can analyze their individual behavior under periodic scheduling independently of each other, and only integrate them in the scheduling synthesis.

3.2.1. Trajectories under Periodic Scheduling

In this subsection, we investigate the behavior of a state variable x_i, for $i = 1, \ldots, n$, under a periodic schedule $u_i(\cdot)$. In particular, we will study its limit behavior as $t \to \infty$. As

illustrated in Figure 2.3 on page 27 and formally proved in Appendix A.1.2 on page 167, as time t progresses, $x_i(t)$ converges to a bounded interval $[\alpha_i, \beta_i]$. Using calculus and linear system theory, the limits α_i and β_i can be calculated explicitly by[2]

$$\alpha_i = \frac{N_i}{1 - M_i} \quad \text{and} \quad \beta_i = \frac{P_i}{1 - M_i} \tag{3.4}$$

where

$$M_i = \mathrm{e}^{-\delta\left(a_{\mathrm{off},i}(1-\eta_i)+a_{\mathrm{on},i}\eta_i\right)} \tag{3.5a}$$

$$N_i = \mathrm{e}^{-a_{\mathrm{off},i}(1-\eta_i)\delta}\left(\frac{b_{\mathrm{on},i}}{a_{\mathrm{on},i}} - \frac{b_{\mathrm{off},i}}{a_{\mathrm{off},i}}\right) - \frac{b_{\mathrm{on},i}}{a_{\mathrm{on},i}}M_i + \frac{b_{\mathrm{off},i}}{a_{\mathrm{off},i}} \tag{3.5b}$$

$$P_i = \mathrm{e}^{-a_{\mathrm{on},i}\eta_i\delta}\left(\frac{b_{\mathrm{off},i}}{a_{\mathrm{off},i}} - \frac{b_{\mathrm{on},i}}{a_{\mathrm{on},i}}\right) - \frac{b_{\mathrm{off},i}}{a_{\mathrm{off},i}}M_i + \frac{b_{\mathrm{on},i}}{a_{\mathrm{on},i}}. \tag{3.5c}$$

Observe that the limits α_i and β_i depend on the values of δ and η_i, but not r_i. This is because the initial delay $r_i\delta$ is shorter than a time period and does not affect the overall behavior of x_i.

3.2.2. Periodic Green Scheduling Synthesis

Having described the limit behavior of each individual state variable x_i under periodic schedule $u_i(\cdot)$, we can now derive a method for synthesizing a safe periodic schedule $u(\cdot)$ for the entire system. We mentioned in Section 3.1 that the periodic schedule $u(\cdot)$ is determined by the set of parameters $(\delta, (\eta_1, r_1), \ldots, (\eta_m, r_m))$. Therefore, we only need to find the values of these parameters so that for each $i = 1, \ldots, n$, $x_i(t)$ enters and stays indefinitely in the desired range $[l_i, h_i]$ after some finite time $\tau_i \geq 0$. As presented in the previous subsection, under periodic schedule $u_i(\cdot)$, the state variable x_i converges to the bounded interval $[\alpha_i, \beta_i]$ given by Equations (3.4) to (3.5c). It follows that if for all $i = 1, \ldots, n$, the interval $[\alpha_i, \beta_i]$ is inside the desired range $[l_i, h_i]$, i.e., $[\alpha_i, \beta_i] \subseteq [l_i, h_i]$, then the system will be safe. Thus the next Corollary is straightforward.

Corollary 3.1 *If the parameters of the periodic schedule $u(\cdot)$ are such that for all $i = 1, \ldots, n$,*

[2]For details, see Appendix A.1.2 on page 167.

$\alpha_i \geq l_i$ and $\beta_i \leq h_i$, then the system is safe under this schedule. □

We recall that α_i and β_i do not depend on r_i. Therefore, we can construct periodic schedules $u_i(\cdot)$ in two steps: (1) compute δ and η_i for each i so as to make the limits $[\alpha_i, \beta_i]$ bounded in $[l_i, h_i]$; (2) given parameters δ and η_i for all i, find r_i so that at any time $t \geq 0$, $\sum_{i=1}^{n} u_i(t) \leq k$.

Step 1: Compute δ and η_i for each i

In this step we compute the common time period δ and the utilization η_i for each i so that $\alpha_i \geq l_i$ and $\beta_i \leq h_i$. In practice, the time period δ is determined by the characteristics of the physical equipment and the hardware platform. Thus, we assume that $\delta > 0$ is provided and we need to compute η_i for $i = 1, \ldots, n$. Because α_i and β_i now depend on η_i, we will write $\alpha_i(\eta_i)$ and $\beta_i(\eta_i)$ to emphasize that they are functions of η_i.

By taking the derivative of α_i with respect to η_i, it is straightforward to verify that

$$\frac{\mathrm{d}\alpha_i}{\mathrm{d}\eta_i} = \frac{\delta M_i}{(1-M_i)^2}\left(\frac{\underline{b}_{\mathrm{on},i}}{\underline{a}_{\mathrm{on},i}} - \frac{\underline{b}_{\mathrm{off},i}}{\underline{a}_{\mathrm{off},i}}\right)\left[\underline{a}_{\mathrm{on},i}\left(1 - \mathrm{e}^{-\underline{a}_{\mathrm{off},i}(1-\eta_i)\delta}\right) + \underline{a}_{\mathrm{off},i}\left(\mathrm{e}^{\underline{a}_{\mathrm{on},i}\eta_i\delta} - 1\right)\right].$$

Therefore $\frac{\mathrm{d}\alpha_i}{\mathrm{d}\eta_i} > 0$ for all $0 \leq \eta_i \leq 1$ and $\delta > 0$, i.e., the function $\alpha_i(\eta_i)$ is strictly increasing with respect to η_i. It follows that $\alpha_i(\eta_i) \geq l_i$ is equivalent to $\eta_i \geq \underline{\eta}_{\delta,i}$ where $\underline{\eta}_{\delta,i}$ is the root of the equation $\alpha_i(\eta_i) = l_i$. Although we do not have a closed-form expression for $\underline{\eta}_{\delta,i}$, the equation can be numerically solved efficiently using Newton's method since $\alpha_i(\eta_i)$ is strictly monotonic and its derivative can be calculated exactly. Similarly, the constraint $\beta_i(\eta_i) \leq h_i$ is equivalent to $\eta_i \leq \overline{\eta}_{\delta,i}$ where $\beta_i(\overline{\eta}_{\delta,i}) = h_i$, which can also be solved numerically.

It is worth noting the relation between the bounds $\underline{\eta}_{\delta,i}$, $\overline{\eta}_{\delta,i}$ and the bounds $\underline{\eta}_i$, $\overline{\eta}_i$ obtained in Lemma 2.1 on page 28 for schedulability conditions. It is shown in Appendix A.1.2 on page 167 that

$$\lim_{\delta \to 0^+} \alpha_i(\eta_i) = \frac{b_{\mathrm{on},i}\eta_i + b_{\mathrm{off},i}(1-\eta_i)}{a_{\mathrm{on},i}\eta_i + a_{\mathrm{off},i}(1-\eta_i)}.$$

By simple calculations, we can verify that $\underline{\eta}_i$ is the root of the equation $\lim_{\delta \to 0^+} \alpha_i(\eta_i) = l_i$.

Thus $\underline{\eta}_i$ is the value of $\underline{\eta}_{\delta,i}$ as δ goes to 0. Similarly $\overline{\eta}_i$ is the value of $\overline{\eta}_{\delta,i}$ as $\delta \to 0$. Therefore, the sufficient schedulability condition in Theorem 2.2 on page 28 corresponds to the extreme case when δ is infinitesimal, that is when u_i switches infinitely fast.

If $\underline{\eta}_{\delta,i} \leq \overline{\eta}_{\delta,i}$ then we can choose any value η_i in the range $[\underline{\eta}_{\delta,i}, \overline{\eta}_{\delta,i}]$. Otherwise, if $\underline{\eta}_{\delta,i} > \overline{\eta}_{\delta,i}$, the chosen time period δ is infeasible and δ needs to be reduced. Indeed, there exists a maximal feasible time period $\delta_i^\star$, for which $\underline{\eta}_{\delta,i} = \overline{\eta}_{\delta,i}$, that can be computed numerically. Figure 3.2 on page 70 in Example 3.1 illustrates such a feasible region $(\delta, [\underline{\eta}_{\delta,i}, \overline{\eta}_{\delta,i}])$ for the room-heater example. Obviously, the chosen common time period δ must satisfy $\delta \leq \delta_i^\star$ for all $i = 1, \ldots, n$.

Because the dynamics of the state variables are decoupled, the simple calculation of $\underline{\eta}_{\delta,i}$, $\overline{\eta}_{\delta,i}$ and η_i for each x_i can be carried out independently of the other state variables. Hence, the computation of this step is efficient even for large-scale systems, as will be demonstrated later in Example 3.1.

Step 2: Compute initial delay r_i

In step 1, we have chosen a feasible common period δ and the utilizations η_i for all component schedules $u_i(\cdot)$. In this step, given a feasible peak constraint k (cf. schedulability conditions and feasible peak constraints in Section 2.3.1 on page 26), we compute the initial delays r_i, $i = 1, \ldots, n$, so that the aggregated periodic schedule $u(\cdot)$ satisfies $\sum_{i=1}^n u_i(t) \leq k$, $\forall t \geq 0$.

In general, this problem is similar to multiprocessor real-time scheduling of periodic tasks with full migration (Davis and Burn, 2009), in which δ is the tasks' period, η_i are the tasks' utilizations, and k is the number of identical processors. Conventional multiprocessor scheduling algorithms can then be used to derive a schedule for the system. We present instead in Appendix A.1.3 on page 169 a simple algorithm for obtaining the values r_i so that the peak constraint is always respected. The algorithm is reproduced below for the readers' reference. It is assumed that $\sum_{i=1}^n \eta_i \leq k$.

1. Distribute n non-overlapping right-open subintervals, each of length η_i respectively, into the interval $[0, k]$ on the real line (see Figure A.1a on page 170). Because $\sum_{i=1}^{n} \eta_i \leq k$, such a distribution is always possible.

2. For each i, let subinterval i be $[s_i, s_i + \eta_i) \subseteq [0, k]$. Then choose $r_i = s_i - \lfloor s_i \rfloor$.

Therein, we also prove that with these initial delays, the aggregated periodic schedule $u(\cdot)$ satisfies the peak constraint (see Figure A.1b for an illustration with $n = 3$ and $k = 2$). This algorithm is simple and hence scalable for a large n. However, we have not discussed in the first step how the subintervals $[s_i, s_i + \eta_i)$ are sequenced in the interval $[0, k]$, i.e., the order of actuation of u_i. This ordering problem will be discussed in Section 3.4.

Putting everything together, the periodic green scheduling synthesis for decoupled affine monotone dynamics is summarized in Algorithm 3.1 on the next page. On Line 12, the function SequenceTasks[3] performs the ordering of the non-overlapping subintervals $[s_i, s_i + \eta_i)$ into the interval $[0, k]$, and returns the offsets $r_1, \ldots, r_n$. The simplest ordering algorithm is given in Algorithm 3.2 on the following page, which simply sequences the subintervals in the given order from 1 to n.

Example 3.1 In Example 2.1 on page 29 we have verified that the small-scale room-heater system with 6 rooms and 6 heaters, when there are no thermal interactions nor disturbances, is schedulable with any peak constraint $k \geq 3$. We now apply the periodic scheduling synthesis in Algorithm 3.1 to this system. We chose the peak constraint $k = 4$ and the time period $\delta = 1800\,\mathrm{s} = 30\,\mathrm{min}$, which resulted in an utilization vector $\eta = [0.61, 0.63, 0.60, 0.58, 0.55, 0.55]^T$ and initial delays $(r_1 = 0, r_2 = 0.61, r_3 = 0.23, r_4 = 0.83, r_5 = 0.42, r_6 = 0.97)$. Since $\sum_{i=1}^{6} \eta_i = 3.52 < 4$, the peak constraint $k = 4$ is feasible. For this system, the feasible region $(\delta, [\underline{\eta}_{\delta,1}, \overline{\eta}_{\delta,1}])$ is plotted in Figure 3.2, which shows that the maximal feasible period $\delta^\star$ for heater 1 is about $6000\,\mathrm{s} = 100\,\mathrm{min}$. Also, we observe that as $\delta \to 0$, the range of utilization $[\underline{\eta}_{\delta,1}, \overline{\eta}_{\delta,1}]$ converge to the values $\underline{\eta}_1 = 0.4939$ and $\overline{\eta}_1 = 0.6256$ obtained in the schedulability

[3] Each subinterval corresponds to a *task* in the terminology of real-time scheduling, hence the name of this function.

Algorithm 3.1 Periodic Scheduling Synthesis for Decoupled Affine Monotone Dynamics

Input: parameters of decoupled affine monotone dynamics; bounds l_i, h_i for each $i = 1, \ldots, n$; peak constraint k; δ.

Output: parameters (η_i, r_i) for each $i = 1, \ldots, n$.

for $i \leftarrow 1, \ldots, n$ **do**
  $\underline{\eta}_{\delta,i} \leftarrow$ root of $\alpha_i(\eta_i) = l_i$ ▷ cf. Equation (3.4)
  $\overline{\eta}_{\delta,i} \leftarrow$ root of $\beta_i(\eta_i) = h_i$ ▷ cf. Equation (3.4)
  if $\underline{\eta}_{\delta,i} > \overline{\eta}_{\delta,i}$ **then**
    return "δ is infeasible"
  end if
end for
if $\sum_{i=1}^{n} \underline{\eta}_{\delta,i} > k$ **then**
  return "Not schedulable"
end if
Select $\eta_i \in [\underline{\eta}_{\delta,i}, \overline{\eta}_{\delta,i}]$ for each $i = 1, \ldots, n$ so that $\sum_{i=1}^{n} \eta_i \leq k$
$(r_1, \ldots, r_n) \leftarrow$ SequenceTasks$(k, \eta_1, \ldots, \eta_n)$

Algorithm 3.2 Simple Subinterval Sequencing for Periodic Scheduling Synthesis

function SimpleSequenceTasks$(k, \eta_1, \ldots, \eta_n)$
  Require: $\sum_{i=1}^{n} \eta_i \leq k$
  $s \leftarrow 0$
  for $i \leftarrow 1, \ldots, n$ **do**
    $r_i \leftarrow s - \lfloor s \rfloor$
    $s \leftarrow s + \eta_i$
  end for
  return $(r_1, \ldots, r_n)$
end function

analysis in Example 2.1. The scheduling synthesis algorithm took only 15 ms to complete.

The synthesized periodic schedule was simulated for 10 hours, starting from the initial temperature 18 °C for all rooms. For comparison, we also simulated the uncoordinated two-position control rule (see Section 1.2.1 on page 5). Their results are reported in Figure 3.3 on the next page. For periodic green scheduling, the room temperatures converged more slowly to the desired temperature range than for uncoordinated scheduling. However, it reduced the peak demand from 39 kW to 27 kW – a 30.77% saving. Both scheduling strategies resulted in similar total energy consumption: 224.41 kW h for uncoordinated scheduling and 226.24 kW h for periodic scheduling. It is also evident from Figure 3.3a that the switching frequency

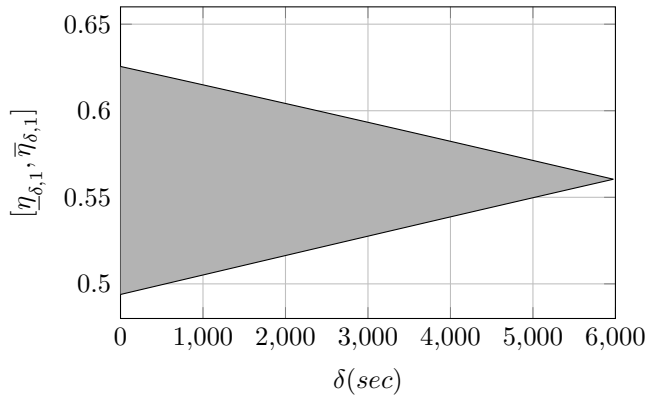
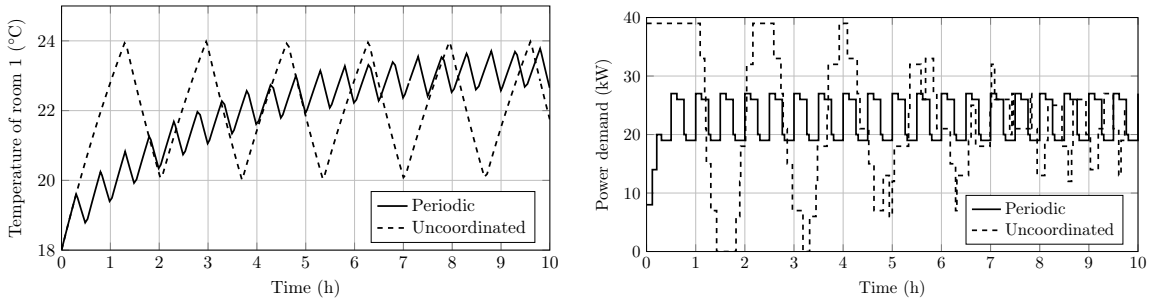


Figure 3.2: The feasible region $(\delta, [\underline{\eta}_{\delta,1}, \overline{\eta}_{\delta,1}])$ for heater 1 in Example 3.1. The maximal feasible period $\delta^\star$ for heater 1 is about $6000\,\mathrm{s} = 100\,\mathrm{min}$.



(a) Temperature of room 1

(b) Aggregated power demand of 6 heaters

Figure 3.3: Simulation results of periodic scheduling for decoupled affine systems in Example 3.1. Compared to uncoordinated scheduling, periodic green scheduling reduced the peak demand from 39 kW to 27 kW. However, the room temperatures converged more slowly to the desired temperature range and the switching frequency was higher.

induced by periodic scheduling is higher than that induced by uncoordinated scheduling.

To demonstrate the scalability of the synthesis method, we scaled the room-heater system to 1000 rooms and 1000 heaters. Algorithm 3.1 took 12.45 s to synthesize periodic schedules for the heaters, and the simulation results were in line with those for the small-scale system. □

3.2.3. Safety Guarantee

Let $x_i(0)$ be any initial value of the state variable x_i, for $i = 1, \ldots, n$. It is possible to determine a finite time $\tau_i \geq 0$ such that, under the periodic schedule $u_i(\cdot)$ with parameters (δ, η_i, r_i), $x_i(t)$ remains inside the interval $[l_i, h_i]$ for all $t \geq \tau_i$. Without loss of generality, we

assume that $r_i = 0$. Indeed, if $r_i > 0$, we simply adjust the initial state to the state after the initial delay, which is $\mathrm{e}^{-a_{\mathrm{off},i} r_i \delta} \left(x_i(0) - \frac{b_{\mathrm{off},i}}{a_{\mathrm{off},i}} \right) + \frac{b_{\mathrm{off},i}}{a_{\mathrm{off},i}}$, then add $r_i \delta$ to the calculated τ_i.

It follows from Equation (A.2) on page 168 and straightforward calculations that the worst-case time instant $\underline{\tau}_i$ after which $x_i(t)$ is guaranteed to be above l_i is:

$$\underline{\tau}_i = \begin{cases} 0 & \text{if } x_i(0) \geq l_i \\ \delta \left\lceil \frac{\log \frac{x_i(0) - \alpha_i(\eta_i)}{l_i - \alpha_i(\eta_i)}}{\delta \left(a_{\mathrm{off},i}(1-\eta_i) + a_{\mathrm{on},i} \eta_i \right)} \right\rceil & \text{otherwise.} \end{cases}$$

Similarly, $x_i(t)$ is guaranteed to be below h_i after the time instant $\overline{\tau}_i$ computed as:

$$\overline{\tau}_i = \begin{cases} \eta_i \delta & \text{if } x_i(\eta_i \delta) \leq h_i \\ \delta \left(\eta_i + \left\lceil \frac{\log \frac{x_i(\eta_i \delta) - \beta_i(\eta_i)}{h_i - \beta_i(\eta_i)}}{\delta \left(a_{\mathrm{off},i}(1-\eta_i) + a_{\mathrm{on},i} \eta_i \right)} \right\rceil \right) & \text{otherwise} \end{cases}$$

where

$$x_i(\eta_i \delta) = \mathrm{e}^{-a_{\mathrm{on},i} \eta_i \delta} \left(x_i(0) - \frac{b_{\mathrm{on},i}}{a_{\mathrm{on},i}} \right) + \frac{b_{\mathrm{on},i}}{a_{\mathrm{on},i}}.$$

Therefore $\tau_i = \max\{\underline{\tau}_i, \overline{\tau}_i\}$, which is finite.

Let $\tau = \max_{i=1,\ldots,n} \tau_i$. Then the system (i.e., all $x_i(t)$) is guaranteed to be safe after this finite time instant τ.

3.2.4. Periodic Scheduling for Affinely Bounded Monotone Dynamics

In Section 2.3.3 on page 30, we introduced the class of *affinely bounded monotone dynamics*, which is more general than the class of decoupled affine monotone dynamics and which can handle small disturbances and small interactions between the state variables. The most important characteristic of the dynamics in this class is summarized in Assumption 2.2. Essentially, for each i, the dynamics of x_i is always bounded between two decoupled affine monotone dynamics, hence its trajectory is always bounded between the trajectories of the lower-bound system $\underline{\Sigma}_i$ (Definition 2.2) and the upper-bound system $\overline{\Sigma}_i$ (Definition 2.3).

Due to this property, the analysis and periodic scheduling synthesis presented above for decoupled affine monotone dynamics can be applied immediately to this class of dynamics by replacing the parameters $a_{\text{on},i}, b_{\text{on},i}, a_{\text{off},i}, b_{\text{off},i}$ with appropriate parameters of the lower- and upper-bound systems.

3.3. Periodic Scheduling for General Affine Dynamics

As discussed in Section 2.4 on page 35, the class of decoupled dynamics, either by nature or by bound dynamics, is limited in practice because practical systems are often subject to significant interactions between their sub-systems. Building upon the analysis in Section 2.4, this section extends the periodic green scheduling synthesis method to more general affine systems of the form, for all $t \geq 0$,

$$\dot{x}(t) = Ax(t) + (B_0 + Bu(t)), \qquad y(t) = Cx(t) \tag{3.6}$$

where $y \in \mathbb{R}^p$ is the output. The assumptions on the system are summarized in the following (for details, see Section 2.4.1 on page 35):

- The state matrix A is Hurwitz;
- The safe set `Safe` is defined for the output y and can be any compact convex set in $\mathbb{R}^p$;
- We consider the n-choose-k case where $k \in \{0, 1, \ldots, m\}$ is the peak constraint imposed on the binary controls u.

3.3.1. Limit Behavior under Periodic Scheduling

We first study the limit behavior of system (3.6) under periodic scheduling and show that it converges to a limit cycle as $t \to \infty$. Consider some periodic schedule $u(\cdot)$ with time period $\delta > 0$ (i.e., $u(\cdot)$ satisfies $u(t) = u(t + \delta)$ for all $t \geq 0$). Let $\eta \in [0, 1]^m$ be the utilization vector corresponding to $u(\cdot)$, that is $\eta_i := \frac{1}{\delta} \int_0^\delta u_i(t)\, dt \in [0, 1]$. Recall that the *average system* of

(3.6) with respect to η is defined as the autonomous, time-invariant affine dynamical system

$$\dot{\bar{x}}(t) = A\bar{x}(t) + (B_0 + B\eta) \tag{3.7}$$

with state variable $\bar{x}$ and starting from the same initial state: $\bar{x}(0) = x(0)$.

The state error $\xi(t) := x(t) - \bar{x}(t)$ between system (3.6) and its average system (3.7) is given in Equation (2.14) on page 38 and can be rewritten as

$$\xi(t) = \mathrm{e}^{A(t-\sigma\delta)} \left(\sum_{i=0}^{\sigma-1} \mathrm{e}^{iA\delta} \right) \xi_\delta + \xi(t - \sigma\delta). \tag{3.8}$$

Since A is Hurwitz, $\mathrm{e}^{A\delta}$ is Schur and therefore $\lim_{\sigma\to\infty} \sum_{i=0}^{\sigma-1} \mathrm{e}^{iA\delta} = \left(\mathrm{I} - \mathrm{e}^{A\delta}\right)^{-1} = P$. The matrix inverse exists because the eigenvalues of $\left(\mathrm{I} - \mathrm{e}^{A\delta}\right)$ are $(1 - \lambda_j)$, where λ_j are the eigenvalues of $\mathrm{e}^{A\delta}$, and hence are non-zero. It follows that as $t \to \infty$, thus $\sigma = \lfloor t/\delta \rfloor \to \infty$, $\xi(t)$ converges to the trajectory $\hat{\xi}(t)$ given by

$$\hat{\xi}(t) = \mathrm{e}^{A(t-\sigma\delta)} P\xi_\delta + \xi(t - \sigma\delta). \tag{3.9}$$

This trajectory is a cycle with period δ because for any $t \geq 0$,

$$\hat{\xi}(t + \delta) = \mathrm{e}^{A(t+\delta-\sigma\delta)} P\xi_\delta + \xi(t + \delta - \sigma\delta)$$

and expanding $\xi(t + \delta - \sigma\delta)$ by Equation (3.8) yields

$$\begin{aligned} &= \mathrm{e}^{A(t-\sigma\delta)} \mathrm{e}^{A\delta} P\xi_\delta + \mathrm{e}^{A(t-\sigma\delta)} \xi_\delta + \xi(t - \sigma\delta) \\ &= \mathrm{e}^{A(t-\sigma\delta)} \left(\mathrm{e}^{A\delta} P + \mathrm{I} \right) \xi_\delta + \xi(t - \sigma\delta) \end{aligned}$$

then using the equality $\mathrm{e}^{A\delta} P + \mathrm{I} = P$ gives

$$= \mathrm{e}^{A(t-\sigma\delta)} P\xi_\delta + \xi(t - \sigma\delta) = \hat{\xi}(t).$$

Furthermore, because the state $\bar{x}(t)$ of the average system converges exponentially to the equilibrium $\bar{x}^\star = -A^{-1}(B_0 + B\eta)$, $x(t) = \bar{x}(t) + \xi(t)$ converges to the limit cycle defined by $\hat{x}(t) = \bar{x}^\star + \hat{\xi}(t)$ as $t \to \infty$. Similarly, the output $y(t)$ converges to the limit cycle given by $\hat{y}(t) = \bar{y}^\star + C\hat{\xi}(t)$ where $\bar{y}^\star = C\bar{x}^\star$.

To compute the limit cycles $\hat{x}(t)$ and $\hat{y}(t)$ for a given δ-periodic schedule $u(\cdot)$, we observe that $\hat{\xi}(t)$ in Equation (3.9) is the solution of the ordinary differential equation (ODE)

$$\dot{\hat{\xi}}(t) = A\hat{\xi}(t) + B\left(u(t) - \eta\right), \qquad \hat{\xi}(0) = P\xi_\delta. \tag{3.10}$$

Therefore, it can be computed numerically using any available ODE solver, in two steps:

1. Compute $\xi_\delta = \xi(\delta)$ by solving the ODE in Equation (2.11) on page 37 for one period;
2. Initialize $\hat{\xi}(0) = P\xi_\delta$ and solve the ODE (3.10) for one period to obtain $\hat{\xi}(t)$ for $t \in [0, \delta]$.

Then the limit cycles can be computed easily.

3.3.2. Periodic Green Scheduling Synthesis

In synthesizing periodic schedules for the system, it is necessary to ensure that the entire limit cycle $\hat{y}(t)$ is inside the safe set `Safe`, so that the system's output is driven to and maintained inside `Safe`. It is usually desirable to minimize the switching frequency of the input u, thus we wish to construct a periodic schedule $u(\cdot)$ with the largest possible time period δ. Once a feasible peak constraint $k \geq k_{\min}$ has been chosen (cf. Section 2.4.3 on page 40), the periodic scheduling synthesis can be achieved by solving the optimization:

$$\begin{aligned} \underset{\eta,\delta,u(\cdot)}{\text{maximize}} \quad & \delta \\ \text{subject to} \quad & \eta \in [0,1]^m, \quad \sum_{i=1}^{m} \eta_i \leq k, \quad \delta > 0 \\ & u(\cdot) \text{ is } \delta\text{-periodic with utilization } \eta \end{aligned} \tag{3.11a}$$

$$\sum_{i=1}^{m} u_i(t) \le k, \qquad \forall 0 \le t \le \delta \tag{3.11b}$$

$$\hat{y}(t) \in \texttt{Safe}, \qquad \forall 0 \le t \le \delta \tag{3.11c}$$

However, this optimization is difficult because $u(\cdot)$ is infinite dimensional and the limit cycle $\hat{y}(t)$ (constraint (3.11c)) cannot be solved analytically but only numerically. Therefore, we restrict $u(\cdot)$ to the specific form defined in Equation (3.1) and use the same construction method as in Section 3.2.2, so that constraints (3.11a) and (3.11b) can be removed. We then use a search algorithm to maximize δ subject to the remaining constraints. Following are the steps to synthesize a periodic schedule with a given feasible peak constraint $k \ge k_{\min}$.

Step 1: Compute utilization

Recall that $\hat{y}(t) = \bar{y}^\star + C\hat{\xi}(t)$ where $\bar{y}^\star$ is the output at the equilibrium $\bar{x}^\star$ of the average system. Intuitively, it is desirable to have $\bar{y}^\star$ not only inside `Safe` but also as far as possible from the boundary of `Safe`. Let y_c be the Chebyshev center[4] of `Safe`. From the schedulability condition in Theorem 2.6, η can be computed by solving the following optimization problem

$$\begin{aligned} \underset{\eta}{\text{minimize}} \quad & \|\bar{y}^\star - y_c\| = \left\|-CA^{-1}(B_0 + B\eta) - y_c\right\| \\ \text{subject to} \quad & \eta \in [0,1]^m \\ & \sum_{i=1}^{m} \eta_i \le k \\ & -CA^{-1}(B_0 + B\eta) \in \text{int}(\texttt{Safe}) \end{aligned} \tag{3.12}$$

in which $\|\cdot\|$ is a vector norm, e.g., Euclidean norm $\|\cdot\|_2$ or ℓ_1-norm $\|\cdot\|_1$. If `Safe` is a hyper-rectangle or a polytope, the last constraint becomes linear (cf. Section 2.4.3), hence the optimization (3.12) is convex and can be solved efficiently (Boyd and Vandenberghe, 2006).

[4]The Chebyshev center of a bounded set C with nonempty interior is a point inside C that is farthest from the exterior of C and is also the center of the largest inscribed ball of C. For convex sets C, computing the Chebyshev center is a convex optimization problem (Boyd and Vandenberghe, 2006, sec. 8.5.1).

Step 2: Construct a periodic schedule

Once η has been computed, a periodic schedule $u(\cdot)$ of the form in Equation (3.1) can be constructed using the same method in Step 2 in Section 3.2.2 (e.g., Algorithm 3.2). The outcome of this step are the initial delays r_i for $i = 1, \ldots, m$. Notice that when the time period δ varies, the constructed schedule $u(\cdot)$ keeps the same pattern (i.e., r_i and η_i for all i) and is only scaled by δ in time.

Step 3: Compute time period δ

This step completes the periodic scheduling synthesis by maximizing the time period δ subject to the constraint (3.11c). Because $\hat{y}(t)$ can only be computed numerically, there is no analytical solution to this optimization. Instead, we approximate δ using a standard bisection search algorithm presented in Algorithm 3.3 on the next page. In each iteration, the function SafeLimitCycle (Line 19) is called to compute the limit cycle $\hat{y}(t)$ for one time period and check whether it is inside `Safe`. The search function MaxPeriodBisection returns an optimal time period $\delta^\star$ with specified tolerance $\epsilon > 0$.

Putting everything together, the scheduling synthesis method is summarized in Algorithm 3.4 on the following page.

Example 3.2 Consider the mass-spring-damper system in Section 1.2.2 on page 8. Suppose that there are no disturbance forces, i.e., $d_i = 0$ for $i = 1, 2$. Because the system's model is in the form of Equation (3.6), we can apply the periodic scheduling synthesis in Algorithm 3.4 to this system. The desirable positions of the masses are $0.75 \leq y_1 \leq 0.85$ and $1.15 \leq y_2 \leq 1.25$, hence the safe set is $\texttt{Safe} = [0.75, 0.85] \times [1.15, 1.25]$. Note that `Safe` is defined for the output variables y (the positions) but not for the state variables x (which include the velocities of the masses). Furthermore, `Safe` does not contain the equilibrium $y^\star = [\frac{2}{3}, \frac{4}{3}]^T$ of the system.

By solving the optimization in Equation (3.12), we obtained the utilization vector $\eta = [0.24, 0.24]^T$, which confirms that the system is schedulable because $\|\eta\|_1 = 0.48 < 1$. The initial delays were chosen to be $r_1 = 0$ and $r_2 = 0.24$. We then maximized the time period by

Algorithm 3.3 Maximize Time Period δ by Bisection Search

Input: parameters $(r_1, \eta_1), \ldots, (r_m, \eta_m)$; initial guess $\delta_0 > 0$; tolerance $\epsilon > 0$; maximal number of iterations N.
Output: maximal time period $\delta^\star$

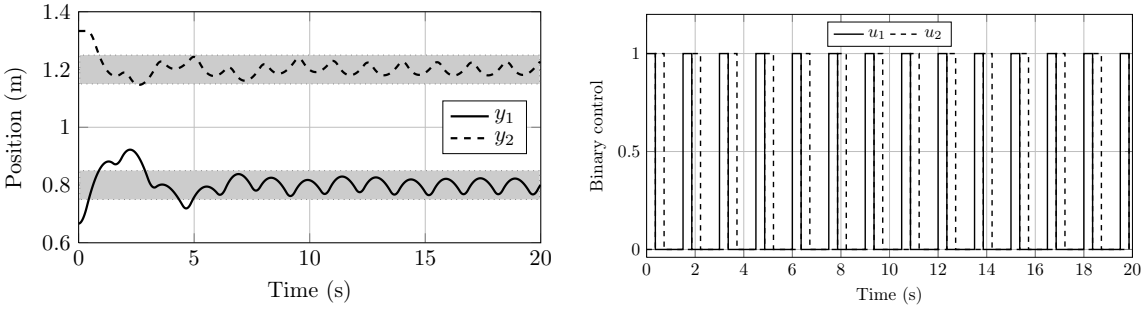
function MaxPeriodBisection
  $\underline{\delta} \leftarrow 0$ ▷ lower end point
  $\bar{\delta} \leftarrow \delta_0$ ▷ upper end point
  while SafeLimitCycle($\bar{\delta}$) **do**
    $\bar{\delta} \leftarrow 2\bar{\delta}$
  end while
  $i \leftarrow 1$ ▷ current iteration
  while $i < N$ and $\bar{\delta} - \underline{\delta} \geq \epsilon$ **do**
    $\delta \leftarrow (\bar{\delta} + \underline{\delta})/2$
    if SafeLimitCycle($\bar{\delta}$) **then**
      $\underline{\delta} \leftarrow \delta$
    else
      $\bar{\delta} \leftarrow \delta$
    end if
    $i \leftarrow i + 1$
  end while
  return $\delta^\star \leftarrow \underline{\delta}$
end function

function SafeLimitCycle(δ)
  Compute limit cycle $\hat{y}(t)$ for $t \in [0, \delta]$
  if $\hat{y}(t) \in$ `Safe` $\forall t \in [0, \delta]$ **then**
    return `true`
  else
    return `false`
  end if
end function

Algorithm 3.4 Periodic Scheduling Synthesis for General Affine Dynamics

Input: parameters (A, B_0, B, C) of the dynamics; safe set `Safe`; feasible peak constraint k.
Output: time period δ; parameters (η_i, r_i) for each $i = 1, \ldots, n$.

Compute Chebyshev center y_c of `Safe`
Solve optimization problem Equation (3.12) to compute η
$(r_1, \ldots, r_n) \leftarrow$ SequenceTasks$(k, \eta_1, \ldots, \eta_n)$
$\delta \leftarrow$ MaxPeriodBisection



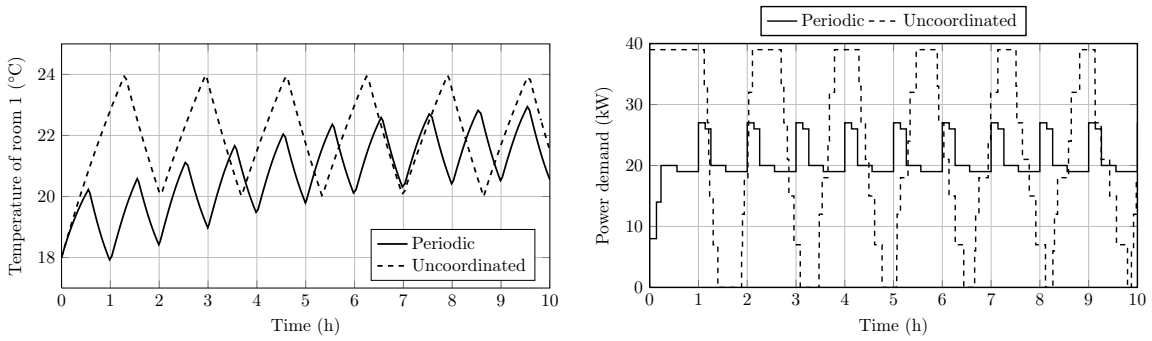
(a) Position trajectories of the two masses: the system is safe because the masses are driven to and maintained inside the ranges of desired positions (gray-filled regions).

(b) Periodic schedules $u_1(\cdot)$ and $u_2(\cdot)$: at any time, at most 1 of the control forces are activated.

Figure 3.4: Simulation results of the mass-spring-damper system with periodic scheduling in Example 3.2.

using Algorithm 3.3 and obtained $\delta^\star = 1.8525\,\mathrm{s}$. The computation took 154.3 ms to complete. A time period $\delta = 1.5\,\mathrm{s}$ was chosen and the system was simulated for 20 s from the initial state $x(0) = [\frac{2}{3}, 0, \frac{4}{3}, 0]^T$. Its results are reported in Figure 3.4 on this page. The positions of the masses, plotted in Figure 3.4a, were driven to the desired ranges (gray-filled regions in the figure) after about 4 s. As can be seen in Figure 3.4b which draws the periodic schedules $u_1(\cdot)$ and $u_2(\cdot)$, at any time, at most 1 of the control forces are activated. □

Example 3.3 We continue the room-heater running example in Example 3.1 on page 68 but consider the thermal interactions between adjacent rooms. Again, the ambient air temperature T_a is constant at 5 °C and there are no internal heat gains for the rooms, i.e., $Q_{g,i} = 0$ for all $i = 1, \ldots, n$. Applying the periodic scheduling synthesis in Algorithm 3.4 with peak constraint $k = 4$ yielded the utilization vector $\eta = [0.5597, 0.5790, 0.5525, 0.5395, 0.5128, 0.5111]^T$ and the maximal time period $\delta^\star = 95.15\,\mathrm{min}$. The computation took 188.6 ms to finish. We chose a time period $\delta = 60\,\mathrm{min}$ and simulated the synthesized periodic schedule for 10 hours, starting from the initial temperature 18 °C for all rooms. For comparison, we also simulated the uncoordinated two-position control rule (see Section 1.2.1 on page 5). Their results are reported in Figure 3.5 on the next page. For periodic green scheduling, the room temperatures converged more slowly to the desired temperature range than for uncoordinated scheduling. It is also obvious that the switching frequency induced by periodic scheduling is



(a) Temperature of room 1

(b) Aggregated power demand of 6 heaters

Figure 3.5: Simulation results of periodic scheduling for the room-heater system with inter-room thermal interactions in Example 3.3. Compared to uncoordinated scheduling, periodic green scheduling reduced the peak demand from 39 kW to 27 kW. However, the room temperatures converged more slowly to the desired temperature range and the switching frequency was higher.

higher than that induced by uncoordinated scheduling. However, green scheduling reduced the peak demand, as it is evident in Figure 3.5b, from 39 kW to 27 kW – a 30.77% saving, and reduced the total energy consumption from 223.62 kW h to 208.71 kW h – a 6.67% saving.

To demonstrate the scalability of the synthesis method, we scaled the room-heater system to 1000 rooms and 1000 heaters. The computation took 20.93 s on MATLAB™ to synthesize periodic schedules for the heaters. □

3.4. On Subinterval Sequencing for Periodic Scheduling

In the periodic scheduling synthesis methods presented in Sections 3.2 and 3.3, we deferred the discussion of the subinterval sequencing problem until this section. Given a utilization vector $\eta \in [0,1]^n$ and a peak constraint $k \geq \sum_{i=1}^{n} \eta_i$, the subinterval sequencing problem concerns with distributing n subintervals, each of length η_i for $i = 1, \ldots, n$, to the interval $[0, k]$ so that the resulted periodic schedule as defined in Equation (3.1) achieves "good" performance. The specific definition of "good" performance is subject to the individual system and application. For instance, good performance can mean that the actual trajectory $y(\cdot)$ stays as close as possible to the trajectory $\bar{y}(\cdot)$ of the average system. In systems where the control inputs u_i incur non-uniform demands (or costs) $c_i > 0$ for $i = 1, \ldots, n$, good

performance can mean the peak aggregated demand $\max_{t\geq 0}\sum_{i=1}^{n} c_i u_i(t)$ is minimized. Since Green Scheduling is motivated by the peak demand reduction problem, we will focus on subinterval sequencing for reducing the peak aggregated demand, stated below.

Problem 3.1 *Given utilization vector $\eta \in [0,1]^n$, peak constraint $k \geq \sum_{i=1}^{n}\eta_i$, and cost vector $c = [c_1,\ldots,c_n]^T$ where $c_i > 0$ for $i = 1,\ldots,n$. Find initial delays $r_1,\ldots,r_n$ (i.e., distribute n subintervals, each of length η_i for $i = 1,\ldots,n$, to the interval $[0,k]$) so that the peak aggregated demand of the resulted periodic schedule is minimized.* □

Apparently, if the demands of the control inputs are uniform, meaning that c_i are the same for all i, then the simple sequencing Algorithm 3.2 on page 69 will produce the least peak demand and will thus suffice. Therefore we only consider the non-uniform case.

Even though c_i are non-uniform, the n-choose-k constraint still holds for the system. Indeed, let σ_k denote the sum of the k smallest entries of c and Σ_k denote the sum of the k largest entries of c. Then from the schedulability results in Chapter 2, any safe periodic schedule, including those that incur the least peak demand, must have at least $k_{\min}$ control inputs simultaneously activated at some point, where $k_{\min}$ is the minimal peak constraint for the system. Therefore a global lower bound of the peak demand can be obtained as $\sigma_{k_{\min}}$. Furthermore, the peak demands of all safe periodic schedules which attain the peak constraint k are always bounded between σ_k and Σ_k. It follows that if the difference between σ_k and Σ_k is inconsiderable, the simple sequencing method in Algorithm 3.2 can be used without much suboptimality.

In the general case, the sequencing problem for peak demand minimization is an instance of the *two-dimensional strip packing problem* (2SP) (Lodi et al., 2002). Each subinterval i, for $i = 1,\ldots,n$, can be represented by a rectangle of width η_i and height c_i. The 2SP concerns with packing these n rectangles into a bin of width 1 and infinite height (called strip) so as to minimize the height to which the strip is used. It is well-known that 2SP is strongly NP-hard (Lodi et al., 2002), therefore the sequencing problem is also strongly NP-hard. A variety of approximation algorithms and metaheuristic algorithms for 2SP have

been proposed (for a good survey, see Lodi et al., 2002). These algorithms can be readily applied to the subinterval sequencing problem for periodic scheduling synthesis. In this dissertation, for simplicity, we only use the proposed simple sequencing algorithm and leave the application of more advanced algorithms for future research.

3.5. Conclusions and Related Work

In this chapter, we have developed periodic scheduling synthesis methods for two types of Green Scheduling systems: decoupled affine monotone dynamics (Section 3.2) and general affine dynamics without disturbances (Section 3.3). Both methods rely on the notions of average systems and limit cycles produced by periodic schedules. By finding the appropriate utilizations η and time period δ that put the limit cycle inside the safe set `Safe`, a safe periodic schedule can be constructed. The most prominent advantage of periodic scheduling, as demonstrated through several examples of the room-heater and the mass-spring-damper systems, is its simplicity and hence its scalability. Except for the NP-hard subinterval sequencing problem (Section 3.4), all calculations are simple and can be performed efficiently even for a large-scale system of 1000 rooms and 1000 heaters.

The problem of scheduling dynamical systems under peak constraint has received increasing attention recently. Lin (2005, Chapter 5) considered a similar problem but in the context of networked control systems and for stability, not safety. Furthermore, the systems were assumed to be decoupled, i.e., there were no inter-system interactions. Based on the same idea of utilizations (therein called attention rates), periodic scheduling strategies (therein called time-division scheduling) were developed to schedule the control systems so that they were stable and at most 1 of them could be activated simultaneously. Le Ny et al. (2011) proposed a periodic scheduling strategy for state feedback Linear Quadratic Regulation (LQR) problem with a constraint on the number of actuation signals that can be updated simultaneously. Their objective was to achieve the best performance, in terms of the LQR cost function, while satisfying the actuation constraint. Similarly, the idea of utilizations (therein called time averages) was used to construct the periodic schedules for the LQG

problem. In (Facchinetti and Vedova, 2011; Facchinetti et al., 2010; Vedova et al., 2010), the authors addressed a similar problem: schedule decoupled affine dynamical systems to reduce peak power such that each state variable is bounded in a given range and at most 1 actuator can be activated at any time. However, conventional real-time scheduling algorithms such as earliest deadline first (EDF) (Liu, 2000) were used in these works instead of pure periodic schedules. Synthesizing switching logic to minimize long run cost for a general hybrid system was studied in (Jha et al., 2010, 2011), in which the limit cycle of the system was utilized similarly to our synthesis method (Section 3.3). The periodic scheduling synthesis problem was formulated as a simulation-based (or black-box) nonlinear optimization and was solved by a generic solver such as `fminsearch` in MATLAB™. Although this approach is generic, it has two drawbacks: (a) simulation-based nonlinear optimization is not scalable; and (b) the hybrid system formulation is not scalable for a large number of discrete modes. As a consequence, the approach is only applicable to small-scale systems with a few state variables and a few discrete modes. Alur et al. (2012) overcame this scalability issue by considering simple constant-rate systems and studied the complexity and decidability of the scheduling problems. They also proposed polynomial-time scheduling algorithms for this class of systems. Finally, in (Subramanian et al., 2012), deferrable electric loads such as electric vehicles were abstracted similarly to computing tasks, thus real-time scheduling algorithms, e.g., EDF, could be applied.

Although periodic scheduling is simple and scalable, the lack of feedback makes it unsuitable for systems with significant disturbances, as we mentioned in the opening of this chapter. In the next chapter, state feedback green scheduling strategies will be developed to deal with the influence of constrained disturbances.

Chapter 4

Feedback Green Scheduling

The periodic scheduling synthesis presented in Chapter 3 is simple and scalable for large-scale systems, as shown in the examples therein. However, the lack of feedback from the system restricts its applicability to Green Scheduling systems without, or with insignificant, disturbances. The concept of feedback has been dominant in control theory and is studied in virtually all control courses and textbooks. By using feedback information, the controller can adjust its action to take into account unforeseen disturbances, therefore desired performance of the closed-loop system can still be achieved regardless of the disturbances. Other advantages of feedback control include the ability to stabilize unstable processes and the ability to handle model uncertainties (Franklin et al., 1998; Dorf and Bishop, 2008).

In this chapter we will study state feedback scheduling strategies for the Green Scheduling problem. Section 4.2 shows that the feedback scheduling strategy developed in Chapter 3 can be made robust to small disturbances. However, this robustness feature usually causes fast switching of the control inputs. Therefore we will use state feedback together with event-triggered and self-triggered techniques to reduce the switching rate of actuation while preserving the system's robustness to disturbances. In Section 4.3, another state feedback scheduling approach based on attracting sets (cf. Section 2.5.2 on page 47) is developed. Again, event-triggered and self-triggered techniques are utilized to decrease the frequency of actuation switching. Finally, state feedback discrete-time Green Scheduling is investigated in Section 4.4 by using backward reachability analysis.

4.1. Preliminaries

This section establishes important results about the bound on state variables of affine dynamical systems under unknown but constrained disturbances. These results will be used

throughout this chapter for deriving robust periodic schedules (Section 4.2) and self-triggered feedback scheduling strategies (Sections 4.2 and 4.3).

Consider an affine dynamical system

$$\dot{x}(t) = Ax(t) + B_0 + Wd(t) \tag{4.1}$$

where, as usual, $x \in \mathcal{X}$ is the state vector and $d \in \mathbb{R}^q$ is the disturbance vector; matrices A and W and vector B_0 have appropriate dimensions. We assume that the disturbances d is unknown but constrained in a known compact and convex set $\mathcal{D} \subset \mathbb{R}^q$. An admissible disturbance signal $d(\cdot)$ must satisfy $d(t) \in \mathcal{D}$ for all $t \geq 0$. The solution (or trajectory) of the dynamics' differential equation with initial condition $x(0) = x_0$ and admissible disturbance signal $d(\cdot)$ is denoted by $x(t; x_0, d(\cdot)) : \mathbb{R}^+ \to \mathbb{R}^n$. We employ the semicolon to distinguish between the time variable t and the parameters x_0 and $d(\cdot)$. When the initial state x_0 and the disturbance signal $d(\cdot)$ are clear from the context, we will omit them and denote the trajectory succinctly as $x(t)$. In the following result, a bound on $x(t)$ is characterized.

Theorem 4.1 *Let matrix $M \in \mathbb{R}^{n\times n}$ and number $\lambda \neq 0$ be such that*

$$M \succ 0, \tag{4.2a}$$

$$A^T M + MA + 2\lambda M \preceq 0. \tag{4.2b}$$

Define

$$\alpha = \frac{1}{\lambda} \sup_{x^T M x = 1} \sup_{d\in\mathcal{D}} x^T M \left(B_0 + Wd\right) = \frac{1}{\lambda} \sup_{d\in\mathcal{D}} \|B_0 + Wd\|_M \tag{4.3}$$

where $\|x\|_M := \sqrt{x^T M x}$ denotes the M-norm of vector x. Then for all initial states $x(0)$ and all admissible disturbance signals,

$$\|x(t)\|_M \leq (\|x(0)\|_M - \alpha)\mathrm{e}^{-\lambda t} + \alpha, \qquad \forall t \geq 0. \tag{4.4}$$

PROOF See Appendix A.2.1 on page 189. ∎

We remark that the first two conditions (4.2a) and (4.2b) of Theorem 4.1 constitute a GEVP[1] and are similar to those in Lemma 2.6 on page 53. However, here λ is not required to be positive. Hence Theorem 4.1 holds for all affine systems, not just those with Hurwitz state matrices A.

While Theorem 4.1 bounds the trajectory with respect to the origin (i.e., $\|x(t)\|_M$), we can bound it with respect to the initial state $x(0)$ (i.e., $\|x(t) - x(0)\|_M$) by the following result.

Corollary 4.1 *Let M and λ be as in Theorem 4.1, and let the initial state $x(0)$ be given. Define $\alpha = \frac{1}{\lambda} \sup_{d\in\mathcal{D}} \|Ax(0) + B_0 + Wd\|_M$. Then for all admissible disturbance signals,*

$$\|x(t) - x(0)\|_M \leq \alpha(1 - \mathrm{e}^{-\lambda t}), \qquad \forall t \geq 0. \tag{4.5}$$

PROOF Let $z(t) = x(t) - x(0)$ and derive $\dot{z}(t)$, then apply Theorem 4.1. ∎

For stable systems, i.e., A is Hurwitz, it is guaranteed that M and $\lambda > 0$ satisfying Theorem 4.1 exist (Proposition 2.1). Then it is obvious that for all initial state $x(0)$, the bound $b(t) := (\|x(0)\|_M - \alpha)\mathrm{e}^{-\lambda t} + \alpha$ converges to $\alpha > 0$ as $t \to \infty$. In particular, if $\|x(0)\|_M < \alpha$ then $b(t)$ is increasing and converges asymptotically to α; if $\|x(0)\|_M > \alpha$ then $b(t)$ is decreasing and also converges asymptotically to α. The following Corollary is a direct consequence of Theorem 4.1.

Corollary 4.2 *Suppose that the system also has control inputs $u \in \mathbb{R}^m$*

$$\dot{x}(t) = Ax(t) + B_0 + Bu(t) + Wd(t), \tag{4.6}$$

and A is Hurwitz. Consider a similar system under the same control input signal $u(\cdot)$, but without disturbances and with a possibly different affine vector $\tilde{B}_0$

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + \tilde{B}_0 + Bu(t). \tag{4.7}$$

Let M and $\lambda > 0$ be as in Theorem 4.1, and let $\alpha = \frac{1}{\lambda} \sup_{d\in\mathcal{D}} \|(B_0 - \tilde{B}_0) + Wd\|_M$. Then if

[1] See Section 2.5.3 on page 50.

$\|x(0) - \tilde{x}(0)\|_M \leq \alpha$, $\|x(t) - \tilde{x}(t)\|_M \leq \alpha$ *for all* $t \geq 0$, *regardless of the disturbances.* □

PROOF Let $z(t) = x(t) - \tilde{x}(t)$ and derive $\dot{z}(t)$, then apply Theorem 4.1. Note that $\|z(0)\|_M \leq \alpha$ and $\alpha > 0$. ■

The above result bounds the trajectories of two similar affine systems, one with disturbances (Equation (4.6)) and one without (Equation (4.7)). If they start α-close to each other then they will remain α-close at all time, regardless of the disturbances.

For systems with outputs $y(t) = Cx(t)$, all the above results still hold if the condition (4.2a) is replaced by $M \succeq C^T C$ and the obtained bounds are on $\|y(t)\|$, $\|y(t) - y(0)\|$, and $\|y(t) - \tilde{y}(t)\|$ respectively. We conclude this section with a remark on related results in the literature.

Remark 4.1 (On reachable sets and other bounds) The *reachable set* $\mathcal{R}_t(x_0)$ of system (4.1) at time $t \geq 0$ from x_0 is defined as the set of all states $x(t; x_0, d(\cdot))$ reachable at time t by the system, with initial condition $x(0) = x_0$ and with all admissible input signals:

$$\mathcal{R}_t(x_0) = \{z \in \mathbb{R}^n \,:\, \exists d(\cdot) \in \mathfrak{F}([0,t], \mathcal{D}), x(t; x_0, d(\cdot)) = z\}.$$

Corollary 4.1 gives us a simple over-approximation of $\mathcal{R}_t(x_0)$ as

$$\mathcal{R}_t(x_0) \subseteq \{z \in \mathbb{R}^n \,:\, \|z - x_0\|_M \leq \alpha(1 - \mathrm{e}^{-\lambda t})\}$$

where $\alpha = \frac{1}{\lambda} \sup_{d \in \mathcal{D}} \left\|Ax_0 + B_0 + Wd\right\|_M$. In many publications of the vast literature on computing reachable sets for dynamical systems (see, for example, Chutinan and Krogh, 2003; Asarin et al., 2003; Girard, 2005; Guernic and Girard, 2010), the bounds on state trajectories were derived from the Fundamental Inequality Theorem from the theory of dynamical systems (Hurewicz, 2002; Hubbard and West, 1995). For the affine system (4.1), this bound is

$$\|x(t) - x_0\|_M \leq \frac{\mu}{\|A\|_M} \left(\mathrm{e}^{\|A\|_M t} - 1\right), \qquad \forall t \geq 0 \tag{4.8}$$

where $\mu = \sup_{d \in \mathcal{D}} \|Ax_0 + B_0 + Wd\|_M = \alpha\lambda$ and $\|A\|_M$ is the induced M-norm of matrix A. Although this bound and the bound in Equation (4.5) look similar, their difference in the exponentials is notable. In particular for stable systems (i.e., A is Hurwitz), while the bound (4.8) always increases unboundedly, the bound (4.5) converges asymptotically to a finite constant $\alpha > 0$ as $t \to \infty$, hence it is strictly tighter than the first. See Appendix A.2.2 on page 190 for a formal proof of this result. In that Appendix, we also show that when A is not Hurwitz, the bound (4.5) is as tight as the bound (4.8).

We also remark that the bound obtained in Theorem 4.1 shares similarities with those presented in (Julius and Pappas, 2009, Propositions 1 & 2) for nonlinear dynamical systems. In particular, for affine systems, the authors also showed conditions similar to Equations (4.2a) and (4.2b). However, the dynamical systems considered in that work were not subject to disturbances. □

4.2. Feedback Scheduling Based on Periodic Scheduling

This section extends the periodic scheduling synthesis presented in Chapter 3 to affine systems with constrained disturbances:

$$\dot{x}(t) = Ax(t) + (B_0 + Bu(t)) + Wd(t) \tag{4.9}$$

in which variables x, u, d and parameters A, B_0, B and W have the usual interpretations. Again, we assume that the matrix A is Hurwitz, the safe set `Safe` is compact and convex, and the disturbances d are constrained in a known compact and convex set $\mathcal{D} \subset \mathbb{R}^q$. We consider the n-choose-k case where k is the peak constraint imposed on the binary controls u.

In this section we aim to develop feedback scheduling strategies based on periodic scheduling that can take account of the disturbances. To this end, in Section 4.2.1, we first improve the periodic scheduling synthesis so that the obtained schedules are robust to the disturbances.

However, robust periodic schedules often result in frequent switching, which is undesirable in practice. Event-triggered and self-triggered feedback scheduling strategies are presented in Sections 4.2.2 and 4.2.3 to improve upon robust periodic scheduling so that the switching frequency can be reduced.

4.2.1. Robust Periodic Scheduling

Consider a system similar to system (4.9) but without disturbances:

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + \tilde{B}_0 + Bu(t), \qquad \tilde{x}(0) = x(0) \tag{4.10}$$

in which the vector $\tilde{B}_0$ can be different from B_0. This system is called a *nominal system* of the system (4.9). Note that the control signal $u(\cdot)$ is the same for both system. It then follows from Corollary 4.2 that, since $x(0) - \tilde{x}(0) = 0$, we can bound the distance between $x(t)$ and $\tilde{x}(t)$ at all time $t \geq 0$ by $\|x(t) - \tilde{x}(t)\|_M \leq \alpha$. The matrix M and the constant $\lambda > 0$ can be found by solving the GEVP in Equations (4.2a) and (4.2b), while the radius α is calculated by solving the quadratic program (QP)

$$\alpha = \frac{1}{\lambda} \max_{d \in \mathcal{D}} \left\| (B_0 - \tilde{B}_0) + Wd \right\|_M. \tag{4.11}$$

We say that a trajectory $x(\cdot)$ is *robustly safe* with *robustness radius* $\alpha \geq 0$ if after some finite time, $x(t)$ not only is inside `Safe` but also keeps a distance of at least α from the exterior of `Safe`, in the metric induced by the M-norm. Mathematically, $x(\cdot)$ satisfies

$$\exists \tau \geq 0 \,:\, \forall t \geq \tau, \mathfrak{B}_M(x(t), \alpha) \subseteq \texttt{Safe}$$

where $\mathfrak{B}_M(x, \alpha) := \{z \,:\, \|z - x\|_M \leq \alpha\}$ denotes the ball with center x and radius α. It is obvious that if the control signal $u(\cdot)$ is such that the trajectory $\tilde{x}(\cdot)$ is robustly safe with radius α then the trajectory $x(\cdot)$ is safe because it is always α-close to $\tilde{x}(\cdot)$. This is illustrated in Figure 4.1 on the following page, in which the dash-dotted line represents the trajectory

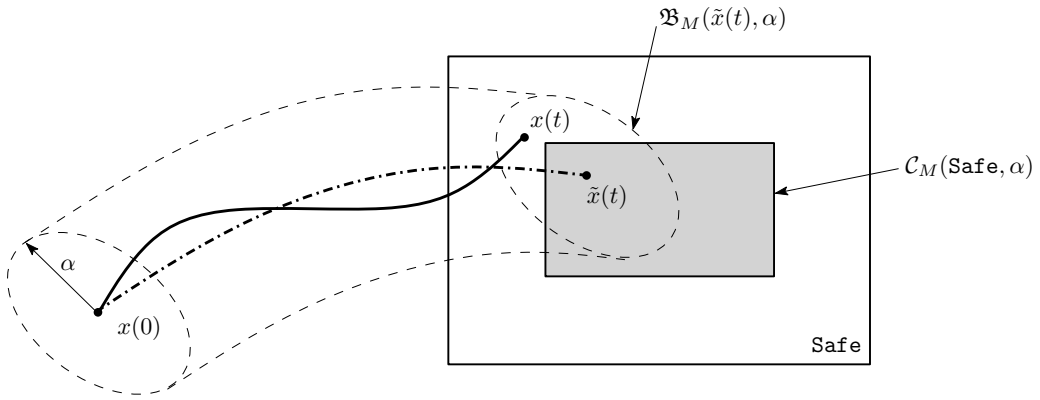


Figure 4.1: Illustration of robust periodic scheduling and the bound between trajectories of the nominal system and the actual system. The gray-filled rectangle is $\mathcal{C}_M(\texttt{Safe}, \alpha)$ – the α-contraction of Safe (the outer rectangle). A tube of radius α (dashed lines) around the nominal trajectory $\tilde{x}(\cdot)$ (dash-dotted line) encloses all possible trajectories of the actual system (solid line). If $\tilde{x}(t) \in \mathcal{C}_M(\texttt{Safe}, \alpha)$ at any t then $x(t) \in \texttt{Safe}$. Hence if $\tilde{x}(\cdot)$ is safe with respect to $\mathcal{C}_M(\texttt{Safe}, \alpha)$ then $x(\cdot)$ is safe with respect to Safe.

$\tilde{x}(\cdot)$ of the nominal system and the solid line represents the trajectory $x(\cdot)$ of the actual system. A tube of radius α (dashed lines) around the nominal trajectory encloses all possible trajectories of the actual system, i.e., at any time t, $x(t)$ always stays within distance α from $\tilde{x}(t)$. Therefore, if $\tilde{x}(t)$ is at least α away from the exterior of Safe, $x(t)$ must be inside Safe.

To characterize schedules $u(\cdot)$ that render $\tilde{x}(\cdot)$ robustly safe, we need the notion of set contraction by a given radius.

Definition 4.1 (α-Contraction of a Set) *Given a radius $\alpha \geq 0$ and a subset S of a set X which is equipped with a metric d, the α-contraction of S is defined as $\mathcal{C}_d(S, \alpha) := \{x \in X : \mathfrak{B}_d(x, \alpha) \subseteq S\}$.* □

The α-contraction of a set S can be computed as the Pontryagin difference[2] of S and the ball $\mathfrak{B}_d(0, \alpha)$: $\mathcal{C}_d(S, \alpha) = S \ominus \mathfrak{B}_d(0, \alpha)$. Employing this definition, a trajectory $\tilde{x}(\cdot)$ is robustly safe with respect to Safe and with radius α if and only if it is safe with respect to the set $\mathcal{C}_M(\texttt{Safe}, \alpha)$. In Figure 4.1, $\mathcal{C}_M(\texttt{Safe}, \alpha)$ is depicted by the smaller gray-filled rectangle inside Safe. Suppose that $\mathcal{C}_M(\texttt{Safe}, \alpha)$ is not empty and system (4.10) is schedulable with respect to the safe set $\mathcal{C}_M(\texttt{Safe}, \alpha)$ and under the peak constraint k. Then a safe schedule $u(\cdot)$ for system (4.10) is also safe for system (4.9) regardless of the disturbances, i.e., the

[2]See Page 55 for the definition of the Pontryagin difference of two sets.

schedule is *robustly safe* for system (4.9). In particular, the synthesis method presented in Section 3.3 can be used to derive a robustly safe periodic schedule $u(\cdot)$.

Choosing the Affine Vector

In the above development, we have not mentioned why the vector $\tilde{B}_0$ of system (4.10) can be different from B_0 and its role in the synthesis. From Equation (4.11), the robustness radius α depends on $\tilde{B}_0$. Intuitively, the smaller α is the better. Thus, it is desirable to choose $\tilde{B}_0$ so as to minimize α. We observe that for any $\tilde{B}_0$, $\alpha\lambda$ is the maximal distance from all points in the set $W\mathcal{D}$ to the point $(\tilde{B}_0 - B_0)$. Therefore, to minimize α, we can select $\tilde{B}_0 = w_c + B_0$ where w_c is the center of the smallest ball $\mathfrak{B}_M(w_c, r)$ that contains the set $W\mathcal{D}$. Usually $\mathcal{D}$ is symmetric about a point d_c, in which case $w_c = Wd_c$. For example, if $d := \{d \in \mathbb{R}^q : \|d\| \leq \beta\}$ for some $\beta > 0$ then $d_c = \mathbf{0}$ and hence $\tilde{B}_0 = B_0$.

In addition to minimizing α, we must choose $\tilde{B}_0$ so that the system (4.10) is schedulable with respect to $\mathcal{C}_M(\texttt{Safe}, \alpha)$ and the given peak constraint k. That is the schedulability condition in Theorem 2.6 on page 39 must be satisfied. If $\tilde{B}_0 = w_c + B_0$ does not satisfy this condition, we can find a different $\tilde{B}_0$ by solving the following optimization:

$$\begin{aligned}
\underset{\tilde{B}_0, \alpha, \eta}{\text{minimize}} \quad & \alpha \\
\text{subject to} \quad & \eta \in [0,1]^m, \quad \sum_{i=1}^{m} \eta_i \leq k \\
& -A^{-1}\left(\tilde{B}_0 + B\eta\right) \in \mathcal{C}_M(\texttt{Safe}, \alpha) \\
& \alpha = \frac{1}{\lambda} \max_{d \in \mathcal{D}} \left\| (B_0 - \tilde{B}_0) + Wd \right\|_M
\end{aligned}$$

The objective is to minimize α while ensuring that the system (4.10) is schedulable. Solving this optimization is difficult because of the set contraction in the second-to-last constraint. Furthermore, it is a bilevel optimization problem due to the last constraint. In the following, an iterative algorithm is proposed to search for $\tilde{B}_0$.

The basic idea of the algorithm is to divide the optimization problem into two sub-problems: one is to choose $\tilde{B}_0$ and calculate α, the other is to solve for the schedulability condition. Specifically, the following steps are performed:

1. Initially select $\tilde{B}_0 = w_c + B_0$ and calculate α.

2. Compute $\mathcal{C}_M(\texttt{Safe}, \alpha)$. If it is empty then report "No solution" and abort. If $-A^{-1}\left(\tilde{B}_0 + B\eta\right) \in \mathcal{C}_M(\texttt{Safe}, \alpha)$ then return $\tilde{B}_0$ and terminate.

3. Find a new $\tilde{B}_0'$ that is closest to $\tilde{B}_0$ while satisfying the schedulability condition with respect to $\mathcal{C}_M(\texttt{Safe}, \alpha)$; that is to solve the optimization

$$\begin{aligned} \underset{\tilde{B}_0', \eta}{\text{minimize}} \quad & \left\| \tilde{B}_0' - \tilde{B}_0 \right\|_M \\ \text{subject to} \quad & \eta \in [0,1]^m, \quad \sum_{i=1}^{m} \eta_i \leq k \\ & -A^{-1}\left(\tilde{B}_0' + B\eta\right) \in \mathcal{C}_M(\texttt{Safe}, \alpha) \end{aligned}$$

4. Calculate the robustness radius α' corresponding to $\tilde{B}_0'$. If $\alpha' \leq \alpha$ then return $\tilde{B}_0'$ and terminate[3].

5. If a given maximal number of iterations has been reached then terminate; otherwise, set $\tilde{B}_0$ to $\tilde{B}_0'$ and α to α', then repeat from step 2.

The Robust Periodic Scheduling Synthesis Algorithm

Putting everything together, the robust periodic scheduling synthesis algorithm consists of two steps:

1. Compute $\tilde{B}_0$, α and $\mathcal{C}_M(\texttt{Safe}, \alpha)$. If $\mathcal{C}_M(\texttt{Safe}, \alpha) = \emptyset$ then terminate.

2. Call Algorithm 3.4 to construct a periodic schedule $u(\cdot)$ for system (4.10) with the

[3]Note that if $\alpha' \leq \alpha$ then $\mathcal{C}_M(\texttt{Safe}, \alpha) \subseteq \mathcal{C}_M(\texttt{Safe}, \alpha')$, hence $-A^{-1}\left(\tilde{B}_0' + B\eta\right) \in \mathcal{C}_M(\texttt{Safe}, \alpha')$.

contracted safe set $\mathcal{C}_M(\texttt{Safe}, \alpha)$.

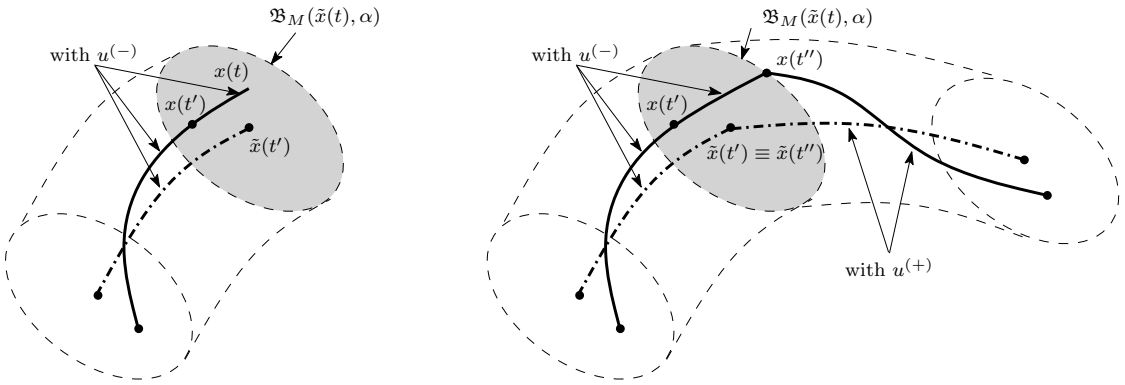
The obtained schedule $u(\cdot)$ will be robustly safe for the original system (4.9).

4.2.2. Event-triggered Feedback Scheduling

As we will see later in the illustrative example in Section 4.2.4, robust periodic scheduling usually induces frequent switching of the control inputs u_i. This is because in order for the system to be robustly safe, the trajectory of the nominal system (4.10) must be kept within a small subset of the safe set `Safe` which is α-away from the exterior of `Safe`. The distance α is therefore called the robustness radius. We showed in the previous subsection that α is not any value but depends on the dynamics of the system and the constraint of the disturbances (Equation (4.11)). Essentially, α represents the *worst-case bound* of the error between the trajectories of the original system (4.9) and the nominal system (4.10). This worst-case scenario means that normally the distance between the two trajectories at any time t is less than α, as illustrated in Figure 4.1 on page 89. This observation will be exploited to reduce the switching frequency of robust periodic scheduling.

Consider any time $t' \geq 0$ at which the robust periodic schedule switches from control vector $u^{(-)} \in \mathcal{U}$ to control vector $u^{(+)} \in \mathcal{U}$. At time t' we have $\|x(t') - \tilde{x}(t')\|_M \leq \alpha$. As discussed above, normally the distance between the trajectories is less than α. Also, we have shown in Section 4.2.1 that as long as $x(t)$ is within the distance of α from $\tilde{x}(t)$, the control input derived for the nominal system will be safe for the original system. It follows that, instead of switching to control vector $u^{(+)}$ immediately at time t' when $\|x(t') - \tilde{x}(t')\|_M < \alpha$, we can delay the switching until the first time that $\|x(t) - \tilde{x}(t')\|_M = \alpha$, for $t \geq t'$. In other words, *we "freeze" the time and evolution of the nominal system at t' and $\tilde{x}(t')$, and let the actual system continue to evolve with input $u^{(-)}$ until $x(t)$ is about to leave the robustness ball $\mathfrak{B}_M(\tilde{x}(t'), \alpha)$; at the time we "unfreeze" the nominal system and switch to input $u^{(+)}$.*

Since the new scheduling algorithm requires monitoring the state of the system, i.e., state feedback, to detect the event $\|x(t) - \tilde{x}(t')\|_M = \alpha$, we will call it an *event-triggered feedback*



(a) At time t', the nominal robust periodic schedule switches from $u^{(-)} \in \mathcal{U}$ to $u^{(+)} \in \mathcal{U}$. The nominal system ($\tilde{x}(\cdot)$) is "frozen" at time t' while the original system ($x(\cdot)$) continues.

(b) When $x(t)$ hits the boundary of $\mathfrak{B}_M(\tilde{x}(t'),\alpha)$ (gray-filled) at time $t'' \geq t'$, the nominal system is resumed and the control input for both systems is switched to $u^{(+)}$.

Figure 4.2: Event-triggered feedback scheduling based on robust periodic scheduling. The trajectory $\tilde{x}(\cdot)$ of the nominal system is drawn in dash-dotted lines, while the trajectory $x(\cdot)$ of the original system is drawn in solid lines.

scheduling algorithm. Figures 4.2a and 4.2b on the current page illustrate the new scheduling algorithm. In Figure 4.2a, both trajectories $\tilde{x}(\cdot)$ (dash-dotted line) and $x(\cdot)$ (solid line) evolve with control vector $u^{(-)}$. However, while $\tilde{x}(\cdot)$ is "frozen" at time instant t', $x(\cdot)$ continues for $t \geq t'$ as long as $x(t)$ is inside $\mathfrak{B}_M(\tilde{x}(t'),\alpha)$. In Figure 4.2b, when $x(t)$ hits the boundary of the robustness ball at time instant $t'' \geq t'$, the evolution of the nominal system is resumed at state $\tilde{x}(t'') \equiv \tilde{x}(t')$ and the control input (for both systems) is switched to $u^{(+)}$. It is obvious that the nominal trajectory $\tilde{x}(\cdot)$ does not change in shape but only in time. That is if we remove all the time intervals during which the nominal system is "frozen" we will recover the original nominal trajectory. Therefore $\tilde{x}(\cdot)$ is still robustly safe with robustness radius α. Because $x(t)$ is always maintained within distance α from $\tilde{x}(t)$, $x(\cdot)$ must be safe.

Let us formulate mathematically the resulted schedule. Any nominal δ-periodic schedule $\tilde{u}(\cdot)$ of the form in Equation (3.1) can be represented by an infinite sequence of pairs $(u^{(0)},\tau_0),(u^{(1)},\tau_1),\ldots$ that satisfy the following conditions:

- For all $i = 0, 1, \ldots$, $u^{(i)}$ is a valid control vector in $\mathcal{U}$ and $\tau_i > 0$ is the time duration that $u^{(i)}$ is applied;

- For all $i = 0, 1, \ldots$, $u^{(i)} \neq u^{(i+1)}$;
- There exists $s > 1$ such that $u^{(i)} = u^{(i+s)}$ and $\tau_i = \tau_{i+s}$ for all $i = 0, 1, \ldots$, and $\sum_{i=0}^{s-1} \tau_i = \delta$.

At any time $t \geq 0$, $\tilde{u}(t)$ is determined by

$$\tilde{u}(t) = \begin{cases} u^{(0)} & \text{if } 0 \leq t < \tau_0 \\ u^{(i)} & \text{if } \sum_{j=0}^{i-1} \tau_j \leq t < \sum_{j=0}^{i} \tau_j,\ i \geq 1. \end{cases}$$

With the event-triggered feedback scheduling strategy, the resulted schedule $u(\cdot)$ goes through the same sequence of control vectors as the nominal periodic schedule $\tilde{u}(\cdot)$ but with extended time durations. In particular, it is represented by a sequence $(u^{(0)}, \tau_0'), (u^{(1)}, \tau_1'), \ldots$ with $\tau_i' \geq \tau_i$ for all $i = 0, 1, \ldots$ Note that $u(\cdot)$ is no longer periodic in time as $\tilde{u}(\cdot)$ but is still periodic in the sequence of control vectors (i.e., $u^{(i)} = u^{(i+s)}$ for all i).

The flowcharts in Figure 4.3 on page 96 compare the algorithms for periodic scheduling and event-triggered feedback scheduling based on robust periodic scheduling. On the far left, the basic periodic scheduling algorithm has a main loop which simply applies each control vector $u^{(i)}$ for a fixed duration τ_i in the correct sequence. In the middle, the event-triggered scheduling algorithm adds two new steps (gray-filled blocks) to the basic algorithm:

- The first block calculates the state of the nominal system after the delay τ_i, based on Equation (4.10);
- The second block monitors the state of the actual system and detects the event when $\|x(t) - \tilde{x}(t')\|_M = \alpha$.

We note that there is a possibility that the event-triggered scheduling algorithm will result in an unsafe schedule $u(\cdot)$. Suppose at time t', $\tilde{x}(t') \notin \mathcal{C}_M(\texttt{Safe}, \alpha)$ and the nominal system is "frozen." Also, the disturbances on the system are such that $x(t)$ never reaches the boundary of the robustness ball around $\tilde{x}(t')$, hence the event $\|x(t) - \tilde{x}(t')\|_M = \alpha$ never

Algorithm 4.1 Event-triggered Feedback Scheduling Based on Robust Periodic Scheduling

Input: the sequence $(u^{(0)}, \tau_0), (u^{(1)}, \tau_1), \ldots$ of the nominal periodic schedule, the nominal dynamics (4.10), M, λ, maximal waiting time $\tau_{\max}$

1: $i \leftarrow 0$
2: $\tilde{x} \leftarrow x(0)$
3: **while** `true` **do** ▷ main loop repeats indefinitely
4: Apply $u^{(i)}$
5: Delay for τ_i
6: $\tilde{x} \leftarrow \mathrm{e}^{A\tau_i}\tilde{x} + \left(\mathrm{e}^{A\tau_i} - \mathrm{I}\right) A^{-1}(\tilde{B}_0 + Bu^{(i)})$ ▷ update $\tilde{x}$
7: $t' \leftarrow t$ ▷ mark the current time
8: **repeat** ▷ monitor state and detect event
9: Measure (monitor) $x(t)$
10: **until** $(\|x(t) - \tilde{x}\|_M = \alpha) \vee ((\tilde{x} \notin \mathcal{C}_M(\mathtt{Safe}, \alpha)) \wedge (t - t' \geq \tau_{\max}))$
11: $i \leftarrow i + 1$
12: **end while**

occurs. Consequently, the nominal system is stuck at time t' and state $\tilde{x}(t')$ indefinitely and therefore the system is not safe. This situation can be avoided by restricting the waiting time for the event to a finite maximal value, denoted by $\tau_{\max}$. A detailed algorithm of the event-triggered scheduling strategy is listed in Algorithm 4.1 on the current page, in which the symbol t denotes the current time when the corresponding statement is executed. An illustrative example will be given in Section 4.2.4.

4.2.3. Self-triggered Feedback Scheduling

Although the proposed event-triggered feedback scheduling strategy is able to reduce the switching frequency of the control inputs, it requires continuously monitoring the state of the system to ascertain the time instant at which it switches to the next control vector. As a consequence, it necessitates the use of dedicated hardware to track the system's state. To avoid this requirement we develop in this subsection a self-triggered scheduling strategy[4] based on robust periodic scheduling.

We have shown that in event-triggered scheduling, after the nominal delay τ_i of the nominal periodic schedule, we can delay switching the control input as long as $x(t)$ is inside the

[4]In Section 4.5 we will discuss related work on self-triggered control.

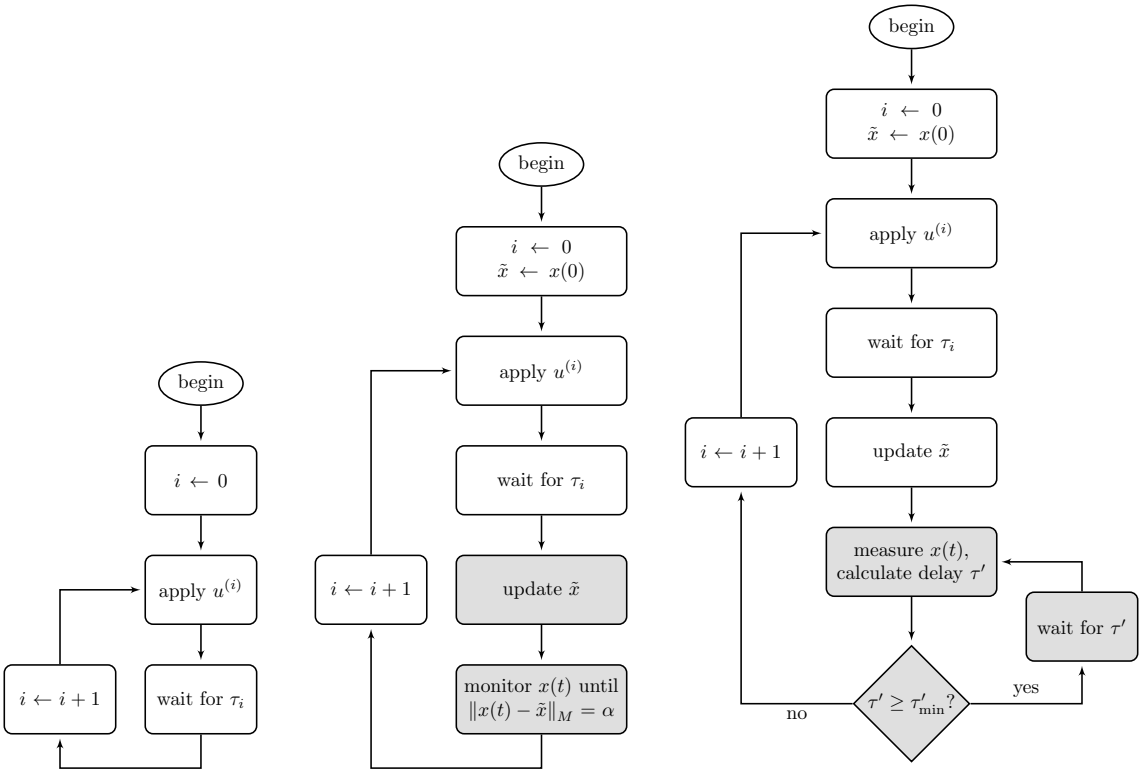


Figure 4.3: Flowcharts comparing the basic periodic scheduling algorithm (left), the event-triggered feedback scheduling algorithm based on robust periodic scheduling (middle), and the self-triggered feedback scheduling algorithm based on robust periodic scheduling (right).

robustness ball. The fundamental idea of self-triggering is to obtain a lower bound estimate of this time delay, and only measure $x(t)$ and decide whether to switch the control input after that minimal time delay. Indeed, let us again use Figure 4.2 on page 93 to explain. Using Theorem 4.1 on page 84, we can attain an upper bound of the distance $\|x(t) - \tilde{x}(t')\|_M$ as a function of time $t \geq t'$

$$\left\|x(t) - \tilde{x}(t')\right\|_M \leq (\left\|x(t') - \tilde{x}(t')\right\|_M - \alpha')\mathrm{e}^{-\lambda t} + \alpha', \qquad \forall t \geq t'$$

where

$$\alpha' = \frac{1}{\lambda} \sup_{d \in \mathcal{D}} \left\|A\tilde{x}(t') + B_0 + Bu^{(-)} + Wd\right\|_M. \tag{4.12}$$

Note that α' depends on $u^{(-)}$ and the state $\tilde{x}(t')$ of the nominal system at time t', and it is not the same as the robustness radius α of the nominal robust periodic schedule. This

bound allows us to determine a time interval $[t', t' + \tau']$ during which $x(t)$ is guaranteed to be in the robustness ball, where $\tau' \geq 0$ is calculated by

$$\tau' = \begin{cases} \frac{1}{\lambda}(\log(\alpha' - \|x(t') - \tilde{x}(t')\|_M) - \log(\alpha' - \alpha)) & \text{if } \alpha < \alpha' \\ \infty & \text{otherwise} \end{cases} \tag{4.13}$$

Similar to event-triggered scheduling, if $\tilde{x}(t') \notin \mathcal{C}_M(\texttt{Safe}, \alpha)$ and the delay τ' exceeds the maximal delay time $\tau_{\max}$, we only delay for $\tau_{\max}$ then switch to the next control vector. On the other hand, for practical purposes, if it is shorter than a predefined minimal time delay $\tau'_{\min} > 0$ then we switch the control input immediately. Otherwise, we delay for τ', after which a new time delay is computed by replacing $x(t')$ with $x(t' + \tau')$ in Equation (4.13). The self-triggered scheduling algorithm is detailed in Algorithm 4.2 on the following page, where the symbol t denotes the current time when the corresponding statement is executed. The flowchart on the far right in Figure 4.3 describes a simplified version of the algorithm with some technical details omitted. It is similar to the event-triggered algorithm in the middle flowchart except the event-triggering mechanism (middle flowchart, bottom block) being replaced by the self-triggering mechanism (right flowchart, gray-filled blocks).

4.2.4. An Illustrative Example

Consider the room-heater running example with 6 rooms/heaters and with disturbances (cf. Section 1.2.1 on page 5). The ambient air temperature T_a, which corresponds to the disturbance variable d_a, varies between $T_{a,\min} = 5\,^\circ\text{C}$ and $T_{a,\max} = 6\,^\circ\text{C}$. For each room $i = 1, \ldots, 6$, the internal heat gain $Q_{g,i}$ (corresponding to disturbance variable $d_{g,i}$) can take any value in $[0, 0.05]$ (kW). Therefore the disturbance set is

$$\mathcal{D} = \left\{ d \in \mathbb{R}^7 \,:\, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5 \end{bmatrix} \preceq d \preceq \begin{bmatrix} 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \\ 6 \end{bmatrix} \right\}.$$

The desired temperature bounds are the same for all rooms: $l_i = 20\,^\circ\text{C}$ and $h_i = 24\,^\circ\text{C}$, $\forall i = 1, \ldots, 6$. Solving the GEVP in Equations (4.2a) and (4.2b) yields $\lambda = 8.9126 \times 10^{-5}$

Algorithm 4.2 Self-triggered Feedback Scheduling Based on Robust Periodic Scheduling

Input: the sequence $(u^{(0)}, \tau_0), (u^{(1)}, \tau_1), \ldots$ of the nominal periodic schedule, the nominal dynamics (4.10), M, λ, maximal waiting time $\tau_{\max}$, minimal waiting time $\tau'_{\min}$

$i \leftarrow 0$
$\tilde{x} \leftarrow x(0)$
while `true` **do** ▷ main loop repeats indefinitely
  Apply $u^{(i)}$
  Delay for τ_i
  $\tilde{x} \leftarrow \mathrm{e}^{A\tau_i}\tilde{x} + \left(\mathrm{e}^{A\tau_i} - \mathrm{I}\right) A^{-1}(\tilde{B}_0 + Bu^{(i)})$ ▷ update $\tilde{x}$
  α' (Equation (4.12))
  $t' \leftarrow t$ ▷ mark the current time
  repeat ▷ self-triggered switching
    Measure $x(t)$
    Calculate τ' (Equation (4.13))
    `switch` $\leftarrow (\tau' < \tau'_{\min}) \vee ((\tilde{x} \notin \mathcal{C}_M(\texttt{Safe}, \alpha)) \wedge (t' + \tau_{\max} - t < \tau'_{\min}))$
    if not `switch` **then**
      if $\tilde{x} \in \mathcal{C}_M(\texttt{Safe}, \alpha)$ **then**
        Delay for τ'
      else
        Delay for $\min(\tau', t' + \tau_{\max} - t)$
      end if
    end if
  until `switch`
  $i \leftarrow i + 1$
end while

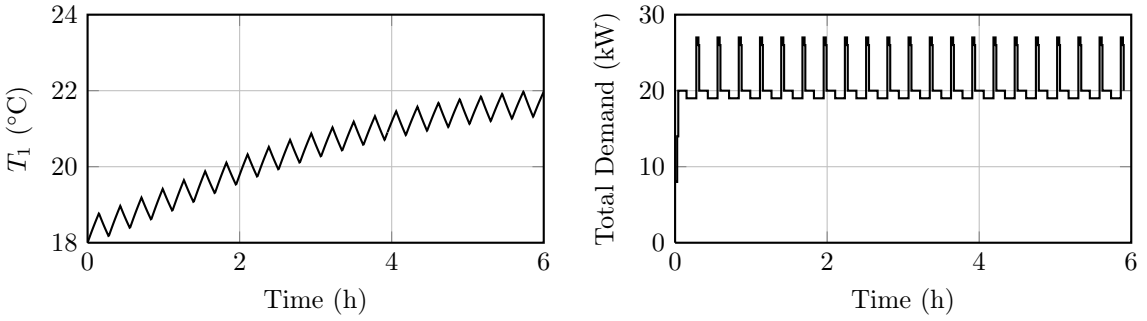
and

$$M = \begin{bmatrix} 1.206 & 0.1404 & 0.007843 & 0.009916 & -0.00688 & 0.04046 \\ 0.1404 & 1.096 & 0.005356 & 0.006772 & -0.004699 & 0.02763 \\ 0.007843 & 0.005356 & 1 & 0.0003783 & -0.0002625 & 0.001544 \\ 0.009916 & 0.006772 & 0.0003783 & 1 & -0.0003319 & 0.001952 \\ -0.00688 & -0.004699 & -0.0002625 & -0.0003319 & 1 & -0.001354 \\ 0.04046 & 0.02763 & 0.001544 & 0.001952 & -0.001354 & 1.008 \end{bmatrix}.$$

We first synthesize a robust periodic schedule for the heaters, following the procedure in Section 4.2.1 on page 91. To find the affine vector $\tilde{B}_0$ for the nominal system (4.10), we observe that $\mathcal{D}$ is symmetric about $d_c = [0.025, \ldots, 0.025, 5.5]^T$. Thus, we choose $\tilde{B}_0 = w_c + B_0 =$ where $w_c = Wd_c$, i.e.,

$$\tilde{B}_0 = 10^{-4} \times \begin{bmatrix} 5.0348 \\ 4.9877 \\ 5.2917 \\ 5.1797 \\ 5.0746 \\ 4.5676 \end{bmatrix}.$$

The robustness radius α is calculated by solving the QP in Equation (4.11) on page 88:



(a) Temperature of room 1.

(b) Total energy demand of all heaters is capped at 27 kW.

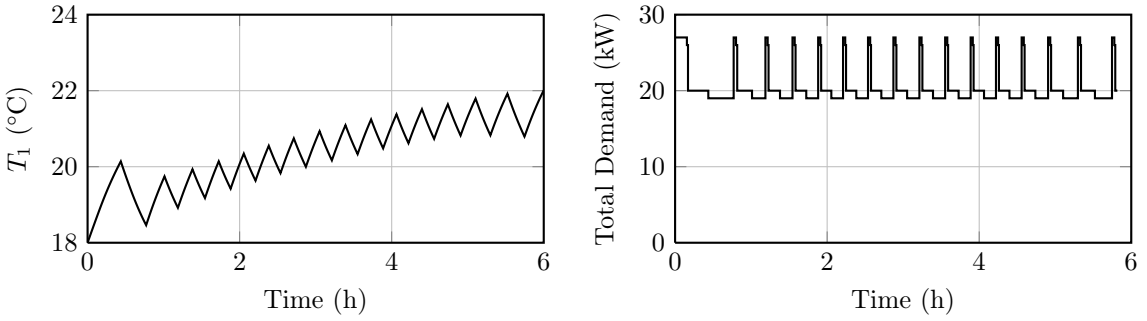
Figure 4.4: Simulation results of the robust periodic schedule for the room-heater example in Section 4.2.4

$\alpha = 1.6184$. Then, the α-contraction of `Safe` is obtained by computing the Pontryagin difference

$$\mathcal{C}_M(\texttt{Safe}, \alpha) = \texttt{Safe} \ominus \mathfrak{B}_M(0, \alpha) = \left\{ x \in \mathbb{R}^6 \ : \ \begin{bmatrix} 21.4860 \\ 21.5581 \\ 21.6182 \\ 21.6181 \\ 21.6182 \\ 21.6135 \end{bmatrix} \preceq x \preceq \begin{bmatrix} 22.5140 \\ 22.4419 \\ 22.3818 \\ 22.3819 \\ 22.3818 \\ 22.3865 \end{bmatrix} \right\}.$$

Because $\mathcal{C}_M(\texttt{Safe}, \alpha)$ is non-empty, we can now synthesize a robust periodic schedule for the heaters by constructing a periodic schedule for the nominal system with the contracted safe set. Using Algorithm 3.4 on page 77, we obtain the utilization vector $\eta = [0.5401, 0.5584, 0.5321, 0.5194, 0.4936, 0.4919]^T$. Since $\sum_{i=1}^{6} \eta_i = 3.1355$, we choose a peak constraint $k = 4$. The simple subinterval sequencing method in Algorithm 3.2 on page 69 is used to compute the initial delays of the heaters as $r_1 = 0$, $r_2 = 0.5401$, $r_3 = 0.0985$, $r_4 = 0.6306$, $r_5 = 0.1500$, $r_6 = 0.6436$. A maximal time period $\delta_{\max} = 1012.5\,\text{s}$ is found by Algorithm 3.3 on page 77 with a tolerance of 30 s. Thus we can choose a time period $\delta = 1005\,\text{s} = 16.75\,\text{min}$. A simulation of the room-heater system for 6 hours with the constructed robust periodic schedule and with randomly generated disturbances was carried out. In Figure 4.4a on the current page, the temperature trajectory of room 1 is plotted; while in Figure 4.4b, the total energy demand of all heaters is shown.

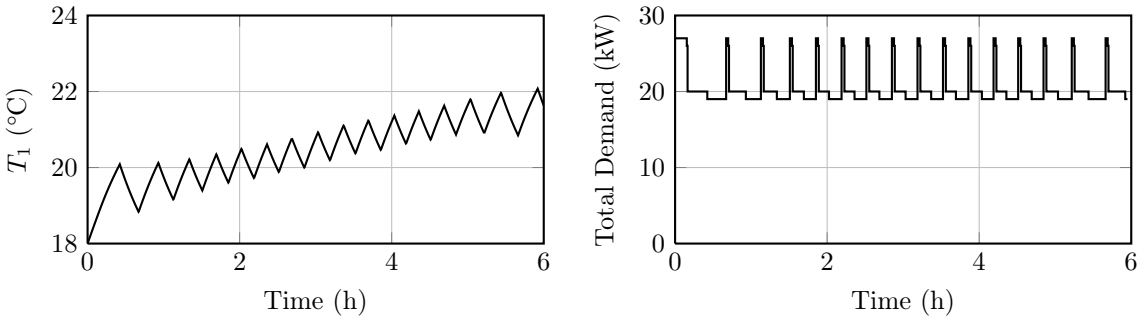
Based on the robust periodic schedule, event- and self-triggered feedback scheduling strategies are derived using Algorithm 4.1 and Algorithm 4.2 respectively. The temperature of room



(a) Temperature of room 1.

(b) Total energy demand of all heaters is capped at 27 kW.

Figure 4.5: Simulation results of the event-triggered feedback scheduling strategy for the room-heater example in Section 4.2.4



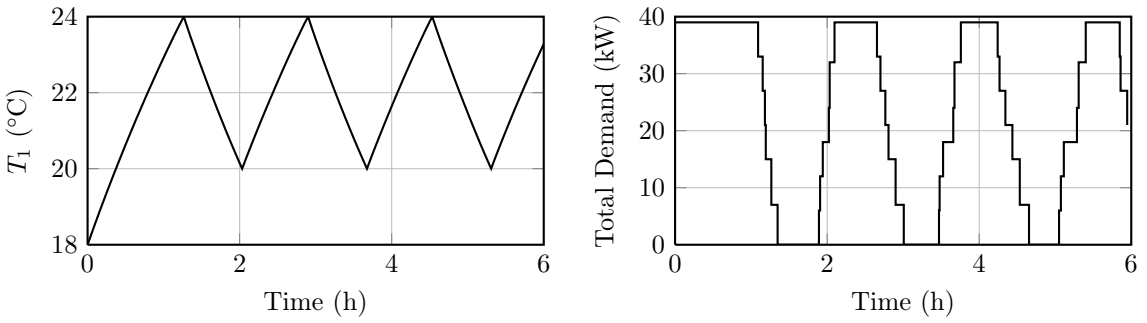
(a) Temperature of room 1.

(b) Total energy demand of all heaters is capped at 27 kW.

Figure 4.6: Simulation results of the self-triggered feedback scheduling strategy for the room-heater example in Section 4.2.4

1 and the total energy demand of all heaters for both cases are displayed in Figures 4.5 and 4.6 on this page respectively. For comparison, we also simulated the system with the uncoordinated scheduling strategy (cf. Section 1.2.1 on page 5). Its results are plotted in Figure 4.7. The energy demand curves for uncoordinated scheduling and for self-triggered scheduling are compared in Figure 4.8.

It is clear that in all cases, the temperature of room 1 was driven to and maintained inside the range 20 °C to 24 °C, although the uncoordinated scheduling strategy takes shorter time to bring T_1 to the desired range. As can be seen in the total energy demand plots, the demand curve for uncoordinated scheduling fluctuates between high peaks (39 kW) and low



(a) Temperature of room 1.

(b) Total energy demand of all heaters.

Figure 4.7: Simulation results of the uncoordinated scheduling strategy for the room-heater example in Section 4.2.4. The demand curve fluctuates from 0 kW to 39 kW and incurs a high peak demand.

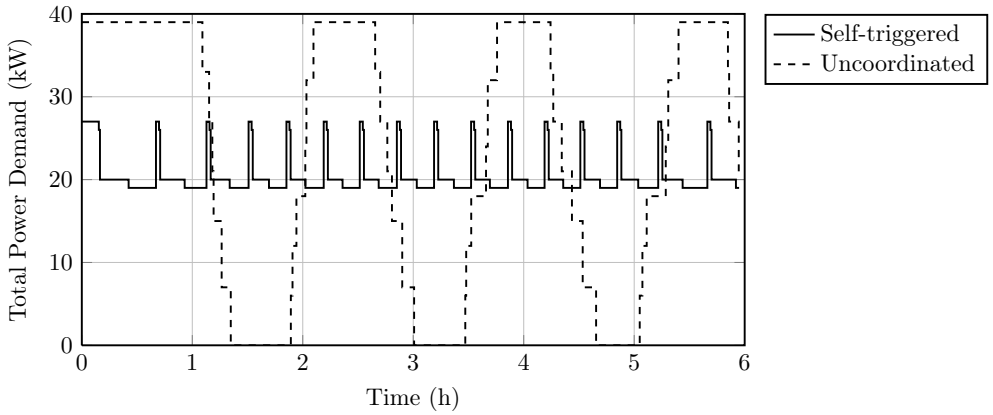


Figure 4.8: Comparison of the energy demand curves of uncoordinated scheduling and of self-triggered scheduling for the room-heater example in Section 4.2.4: the curve for uncoordinated scheduling (dashed line) fluctuates between 0 kW and 39 kW, while that for self-triggered scheduling (solid line) only varies from 19 kW to 27 kW.

valleys (0 kW), while those for the green scheduling strategies are more flat and only vary from 19 kW to 27 kW. This observation is confirmed in Table 4.1 on the following page, where the peak demand and total energy consumption for each scheduling strategy are listed. All green scheduling strategies reduce the peak demand by 30.77% compared to the uncoordinated strategy (from 39 kW down to 27 kW). There are also decreases in the total energy consumption, between 12.78% and 15.17%, since the green schedulers tend to operate at lower mean temperatures than the uncoordinated scheduler.

Although the green scheduling strategies lowered the peak demand as well as the total energy

Table 4.1: Peak demands and total energy consumption for the example in Section 4.2.4. The percentage numbers in parentheses are reductions in peak demands and total energy consumption compared to the uncoordinated scheduling strategy.

	Peak power demand (kW)	Total energy consumption (kW h)
Uncoordinated	39	140.2
Robust	27 (−30.77%)	122.3 (−12.78%)
Event-triggered	27 (−30.77%)	118.9 (−15.17%)
Self-triggered	27 (−30.77%)	121.3 (−13.45%)

Table 4.2: Switching frequency results for the example in Section 4.2.4.

	Total No. of switching	Interval between switching of u_1 (min)		
		Max	Min	Average
Uncoordinated	20	122.00	98.01	106.16
Robust	127	16.75	16.75	16.75
Event-triggered	92	46.23	19.65	22.99
Self-triggered	95	40.28	19.43	22.65

consumption, they caused frequent switching of the heaters as it is obvious from Figures 4.4a to 4.7a. The total numbers of switching (from off state to on state) of all six heaters for the four scheduling strategies are reported in the second column of Table 4.2 on the current page. The last three columns of this table list the maximal, minimal and averaged time intervals between consecutive switching of heater 1 (i.e., u_1) for the corresponding cases. Evidently, the event-triggered and self-triggered scheduling strategies improved upon the basic robust periodic schedule as they reduced the total number of switching (from 127 to 92 and 95 respectively) and extended the time intervals between consecutive switching of the heaters. However, compared to the uncoordinated schedule, all the green scheduling strategies resulted in significantly higher switching frequencies.

4.2.5. Discussion

Although periodic scheduling and its variants (e.g., robust periodic scheduling, event-triggered and self-triggered feedback scheduling based on periodic scheduling) are scalable and simple to implement, they have several drawbacks. As we have seen in the previous example, they usually induce high switching frequencies of the control inputs. This drawback can

be explained by the periodic nature and the worst-case robustness characteristics (i.e., the robustness radius α) of these schedules as they must maintain the nominal trajectory inside a small contracted safe set. Furthermore, these scheduling strategies can only handle small disturbance sets because the robustness radius α increases, usually quickly, with the size of $\mathcal{D}$. For instance, if the ambient air temperature in the previous example could rise to 7 °C instead of just 6 °C then α would exceed the size of `Safe`, resulting in an empty $\mathcal{C}_M(\texttt{Safe}, \alpha)$, hence a robust periodic schedule would not be available. In the next section, we will take a different approach and develop green scheduling strategies based on attracting sets, which can overcome these issues.

4.3. Feedback Scheduling Based on Attracting Sets

This section presents feedback scheduling strategies based on the notion of *attracting sets* of control systems. As in Section 4.2, we consider an affine system with constrained disturbances:

$$\dot{x}(t) = Ax(t) + (B_0 + Bu(t)) + Wd(t). \tag{4.14}$$

Variables x, u, d and parameters A, B_0, B and W have the usual interpretations. We also assume that the state matrix A is Hurwitz, the safe set `Safe` is compact and convex, and the disturbances d are constrained in a known compact and convex set $\mathcal{D} \subset \mathbb{R}^q$. However, we will not restrict the control inputs u to the n-choose-k case, so the valid control set $\mathcal{U}$ can be any subset of the finite, discrete set $\{0,1\}^m$.

4.3.1. Robust Attracting Sets of Feedback Control Systems

In Definition 2.5 on page 48, we defined a (finite-time) attracting set $\mathcal{A}$ of control system (4.14) as a subset of the state space such that there exists an admissible feedback control law $u(t) = \kappa(x(t))$ that, from any initial state $x(0)$ and for any admissible disturbance signal $d(\cdot)$, can drive $x(t)$ to $\mathcal{A}$ in finite time and maintains it inside $\mathcal{A}$ indefinitely. Our aim is to design the feedback law $\kappa(\cdot)$. To this end, consider Theorem 2.9 on page 48 which characterizes

the attracting sets by sublevel sets of a robust control Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}^+$. In Section 2.5.3 on page 50, we showed that for affine systems (4.14), V could have a quadratic form $V(x) := (x - x_c)^T M(x - x_c)$ where $x_c \in \mathcal{X}$ and $M \succeq 0$. The following result allows us to determine an attracting set of affine control system (4.14) and the associated feedback control law. Its proof is given in Appendix A.2.3 on page 191.

Theorem 4.2 *Consider the control system in Equation* (4.14)*. Given any $x_c \in \mathcal{X}$. If there exist $M \in \mathbb{R}^{n\times n}$, $\lambda > 0$ and $\alpha > 0$ such that*

$$M \succeq 0, \tag{4.15a}$$

$$A^T M + MA \preceq -2\lambda M, \tag{4.15b}$$

$$\alpha > \frac{1}{\lambda} \max_{z^T M z = 1} \left(\min_{u\in\mathcal{U}} \max_{d\in\mathcal{D}} z^T M \left(Ax_c + B_0 + Bu + Wd \right) \right) \tag{4.15c}$$

then $\mathcal{A} := \{x \in \mathbb{R}^n \,:\, (x - x_c)^T M (x - x_c) \leq \alpha^2\}$ is an attracting set with state feedback control law $\kappa(\cdot)$ given by

$$\kappa(x) = \arg\min_{u\in\mathcal{U}} (x - x_c)^T MBu, \qquad \forall x \in \mathcal{X}. \tag{4.16}$$

Because A is Hurwitz, M, λ and α always exist, as verified in the following result.

Proposition 4.1 *If the state matrix A is Hurwitz then there exist $M \in \mathbb{R}^{n\times n}$, $\lambda > 0$ and $\alpha > 0$ that satisfy the conditions in Theorem 4.2.* □

PROOF See Appendix A.2.4 on page 192. ■

Matrix M and number $\lambda > 0$ satisfying Equations (4.15a) and (4.15b) can be computed by solving the GEVP

$$\begin{aligned} \underset{\lambda, M}{\text{maximize}} \quad & \lambda \\ \text{subject to} \quad & M \succeq 0 \\ & A^T M + MA \preceq -2\lambda M \end{aligned} \tag{4.17}$$

with scientific computation software such as MATLAB™. To calculate α, we need to solve the optimization in Equation (4.15c). Obviously, this optimization is multilevel, non-convex (due to the constraint $z^T Mz = 1$), bilinear (due to $z^T MBu$ and $z^T MWd$) and combinatorial (because $\mathcal{U}$ is finite), hence it is difficult. In the next subsection, we will discuss how it can be relaxed and α can be computed more efficiently.

Computing α

First, observe that the optimization variables u and d in Equation (4.15c) are separable. Furthermore, as shown in Section 2.5.3 on page 50, under the assumptions of the Green Scheduling problem, we can equivalently replace $\mathcal{U}$ with its convex hull $\text{co}(\mathcal{U})$. Therefore the optimization can be rewritten as

$$\max_{z^T Mz=1} \left(z^T M \left(Ax_c + B_0 \right) + \min_{u \in \text{co}(\mathcal{U})} z^T MBu + \max_{d \in \mathcal{D}} z^T MWd \right) \tag{4.18}$$

which is non-convex, bilevel and bilinear, hence still difficult. However, if x_c is such that for all $d \in \mathcal{D}$, there exists $u \in \text{co}(\mathcal{U})$ satisfying $Ax_c + B_0 + Bu + Wd = \mathbf{0}$, or equivalently if $-W\mathcal{D} \subseteq (Ax_c + B_0) + B\,\text{co}(\mathcal{U})$, then we have (see also Lemma 2.6 on page 53)

$$z^T M \left(Ax_c + B_0 \right) + \min_{u \in \text{co}(\mathcal{U})} z^T MBu + \max_{d \in \mathcal{D}} z^T MWd \leq 0$$

for all $z \in \mathbb{R}^n$. Therefore

$$\max_{z^T Mz=1} \left(z^T M \left(Ax_c + B_0 \right) + \min_{u \in \text{co}(\mathcal{U})} z^T MBu + \max_{d \in \mathcal{D}} z^T MWd \right) \leq 0$$

and any $\alpha > 0$ will suffice the inequality (4.15c) without solving the optimization problem.

If $-W\mathcal{D} \not\subseteq (Ax_c + B_0) + B\,\text{co}(\mathcal{U})$ then we will have to solve Equation (4.18). Consider the optimization

$$\max_{z^T Mz \leq 1} \left(z^T M \left(Ax_c + B_0 \right) + \min_{u \in \text{co}(\mathcal{U})} z^T MBu + \max_{d \in \mathcal{D}} z^T MWd \right) \tag{4.19}$$

which is similar to Equation (4.18) but the constraint $z^T Mz = 1$ is relaxed to $z^T Mz \leq 1$. Obviously the optimal value of Equation (4.18) does not exceed the optimal value of Equation (4.19). Let $(z^\star, u^\star, d^\star)$ be an optimal solution of Equation (4.19). Then $u^\star = \arg\min_{u \in \mathrm{co}(\mathcal{U})} z^{\star T} MBu$ and $d^\star = \arg\max_{d \in \mathcal{D}} z^{\star T} MWd$. Let $J^\star = z^{\star T} M(Ax_c + B_0) + z^{\star T} MBu^\star + z^{\star T} MWd^\star$ be the optimal value. Evidently $J^\star \geq 0$ because with $z = \mathbf{0}$ the value is 0. There are only two cases:

1. $\|z^\star\|_M > 0$: Define $\hat{z} = \frac{z^\star}{\|z^\star\|_M}$ which satisfies the constraint $\hat{z}^T M\hat{z} = 1$. Since $z^\star$ is scaled by the positive number $1/\|z^\star\|_M$ to obtain $\hat{z}$, we have $u^\star = \arg\min_{u \in \mathrm{co}(\mathcal{U})} \hat{z}^T MBu$ and $d^\star = \arg\max_{d \in \mathcal{D}} \hat{z}^T MWd$. We also have

$$\hat{z}^T M(Ax_c + B_0) + \hat{z}^T MBu^\star + \hat{z}^T MWd^\star = \frac{1}{\|z^\star\|_M} J^\star \geq J^\star$$

   where the inequality follows from $\|z^\star\|_M \leq 1$ and $J^\star \geq 0$. Therefore $(\hat{z}, u^\star, d^\star)$ is optimal for both optimizations (4.18) and (4.19). Hence their optimal values are equal.

2. $\|z^\star\|_M = 0$: in this case $J^\star = 0$ and the optimal value of Equation (4.18) is non-positive. Since we are only interested in positive values of α, any $\alpha > 0 = J^\star$ will satisfy the condition (4.15c).

It follows that, for the purpose of calculating $\alpha > 0$, we can solve the relaxed problem (4.19). Note that the relaxed constraint $z^T Mz \leq 1$ is convex.

Let us fix the outer variable z of the bilevel problem (4.19) and consider the convex inner-level optimization $\min_{u \in \mathrm{co}(\mathcal{U})} z^T MBu$. We then obtain the Karush-Kuhn-Tucker (KKT) optimality conditions (see Boyd and Vandenberghe, 2006, sec. 5.5.3) for this problem, which are sufficient because the optimization is convex. Similarly, the KKT conditions are obtained for the convex inner-level problem $\max_{d \in \mathcal{D}} z^T MWd$. These conditions are then collected and incorporated into the constraints of the outer-level problem. In addition, the optimal values of the inner-level problems, in terms of their dual variables, are included in the objective function of the outer-level problem. Eventually, we attain an optimization problem which is

equivalent to Equation (4.19) but is no longer bilevel. As an example, suppose $\mathrm{co}(\mathcal{U})$ and $\mathcal{D}$ are compact polytopes represented by $\{u \in \mathbb{R}^m : H_u u \preceq K_u\}$ and $\{d \in \mathbb{R}^q : H_d d \preceq K_d\}$ respectively. Then the final optimization problem is

$$
\begin{aligned}
\underset{z,u,d,\nu_u,\nu_d}{\text{maximize}} \quad & z^T M(Ax_c + B_0) - K_u^T \nu_u + K_d^T \nu_d \\
\text{subject to} \quad & \nu_u \succeq \mathbf{0}, \quad \nu_d \succeq \mathbf{0}, \quad z^T M z \leq 1 \\
& \left.\begin{aligned} & H_u u \preceq K_u \\ & \nu_u^T (H_u u - K_u) = 0 \\ & B^T M z + H_u^T \nu_u = \mathbf{0} \end{aligned}\right\} \text{KKT conditions of } \min_{u \in \mathrm{co}(\mathcal{U})} z^T M B u \\
& \left.\begin{aligned} & H_d d \preceq K_d \\ & \nu_d^T (H_d u - K_d) = 0 \\ & -W^T M z + H_d^T \nu_d = \mathbf{0} \end{aligned}\right\} \text{KKT conditions of } \max_{d \in \mathcal{D}} z^T M W d
\end{aligned}
\tag{4.20}
$$

where ν_u and ν_d are the dual variables of u and d respectively. We remark that the final optimization (4.20) is still difficult due to the bilinear constraints in the KKT conditions. However, solving this problem is easier than the original problem in Equation (4.15c) because it is no longer multilevel and combinatorial, and because the constraint $z^T M z = 1$ has been relaxed to $z^T M z \leq 1$. This type of optimization problems is readily solved by nonlinear programming solvers such as IPOPT (Wächter and Biegler, 2006) and `fmincon` (of MATLAB™), or by global programming solvers such as BMIBNB in YALMIP (Löfberg, 2012).

In summary, α is computed as follows:

1. if $-W\mathcal{D} \subseteq (Ax_c + B_0) + B\,\mathrm{co}(\mathcal{U})$ then choose any $\alpha > 0$;
2. otherwise, obtain the KKT conditions for inner-level problems $\min_{u \in \mathrm{co}(\mathcal{U})} z^T M B u$ and $\max_{d \in \mathcal{D}} z^T M W d$, then construct the final optimization problem (e.g., Equation (4.20)) and solve it with a nonlinear programming solver or a global programming solver.

4.3.2. Basic Feedback Scheduling

Theorem 4.2 immediately gives us a feedback scheduling strategy for the system in Equation (4.14). Suppose that $\mathcal{A} = \mathfrak{B}_M(x_c, \alpha) \subseteq \texttt{Safe}$. Then by the definition of attracting sets, the feedback control law in Equation (4.16) will drive the state to $\mathcal{A}$ in finite time and maintain it inside $\mathcal{A}$ indefinitely. Because $\mathcal{A}$ is a subset of $\texttt{Safe}$, the system will be safe. Note that we only need to re-compute the control inputs u when x is not inside the attracting ball $\mathcal{A}$. Once x is inside $\mathcal{A}$, we can simply keep the current control vector u until x hits the boundary of $\mathcal{A}$. Therefore, the basic feedback scheduling strategy is given by

$$u(t) = \kappa(x(t)) = \begin{cases} \arg\min_{u \in \mathcal{U}} (x(t) - x_c)^T MBu & \text{if } \|x(t) - x_c\|_M \geq \alpha \\ u(t^-) & \text{otherwise} \end{cases} \tag{4.21}$$

in which $u(t^-)$ denotes the currently used control vector. The minimization in the feedback law can be solved offline. Indeed, let the finite control input set be $\mathcal{U} = \{u^{(1)}, \ldots, u^{(N)}\}$ where each $u^{(i)} \in \{0,1\}^q$ is a valid control vector. Then we partition the state space into $\mathcal{P}_1, \ldots, \mathcal{P}_N$ where each partition $\mathcal{P}_i \subset \mathcal{X}$ is defined as

$$\mathcal{P}_i := \{x \in \mathbb{R}^n \,:\, (x - x_c)^T MB(u^{(i)} - u^{(j)}) \leq 0 \; \forall j \neq i\}.$$

It is simple to verify that for each $i = 1, \ldots, N$, if $x \in \mathcal{P}_i$ then $u^{(i)}$ is an optimal solution of Equation (4.21). By pre-computing and storing the partitions $\mathcal{P}_1, \ldots, \mathcal{P}_N$, the feedback law $\kappa(x)$ can be calculated by checking the set membership of x.

Although the basic feedback scheduling strategy is straightforward, it has several limitations:

1. It requires continuous monitoring of the state.

2. It requires solving the minimization (4.21) in continuous time, which is impractical.

3. It usually results in a sliding mode control signal (Edwards and Spurgeon, 1998):

when x is on a common surface between two partitions $\mathcal{P}_i$ and $\mathcal{P}_j$, $i \neq j$, the control signal will switch rapidly and repeatedly between $u^{(i)}$ and $u^{(j)}$, causing the state x to slide along the common surface. Therefore, the switching frequency is very high, as illustrated in the next example. This effect is often undesirable in practice.

In the next subsections, we will develop event- and self-triggered feedback scheduling strategies to overcome these limitations of the basic strategy.

Example 4.1 Consider again the mass-spring-damper example in Section 1.2.2 on page 8. The disturbance forces d_1 and d_2 have the maximal magnitude $d_{\max} = 0.1$, i.e., $|d_1| \leq 0.1$ and $|d_2| \leq 0.1$, or $\mathcal{D} = [-0.1, 0.1] \times [-0.1, 0.1]$. The desired safe set of positions y_1 and y_2 is $\texttt{Safe} = [0.75, 0.85] \times [1.15, 1.25]$, that is we want to maintain $0.75 \leq y_1 \leq 0.85$ and $1.15 \leq y_2 \leq 1.25$. Using the sufficient schedulability condition in Theorem 2.12 on page 57, we verified that the mass-spring-damper system is schedulable with peak constraint $k = 1$. In this example, we will apply the basic feedback scheduling strategy in Equation (4.21) to safely control the masses.

Recall that the state vector is defined as $x = [y_1, v_1, y_2, v_2]$ where $v_1 = \dot{y}_1$ and $v_2 = \dot{y}_2$ are the velocities of the masses. Choose $x_c = [0.8, 0, 1.2, 0]^T$. Solving the GEVP (4.17) yields $\lambda = 0.3621$ and

$$M = \begin{bmatrix} 1.2933 & 0.6137 & 0.0651 & 0.5326 \\ 0.6137 & 1.4109 & 0.1477 & 0.9040 \\ 0.0651 & 0.1477 & 1.3931 & 0.5938 \\ 0.5326 & 0.9040 & 0.5938 & 1.9657 \end{bmatrix}.$$

We can verify that $-W\mathcal{D} \subseteq (Ax_c + B_0) + B\,\mathrm{co}(\mathcal{U})$, hence we do not need to solve the optimization (4.15c) and can choose any $\alpha > 0$. Let us select α such that $\mathfrak{B}_M(x_c, \alpha)$ is the largest ball satisfying $C\mathfrak{B}_M(x_c, \alpha) \subseteq \texttt{Safe}$. Using MATLAB™ α is computed to be 0.0424.

The basic feedback law is implemented and simulated in MATLAB™ for 10 s with randomly generated disturbances and with initial state $x(0) = [\frac{2}{3}, 0, \frac{4}{3}, 0]^T$. The disturbance signals $d_1(\cdot)$ and $d_2(\cdot)$ are plotted in Figure 4.9 on the next page. In Figure 4.10a on the following page, the position trajectories $y_1(\cdot)$ and $y_2(\cdot)$ are depicted. The resulted control signal $u_1(\cdot)$ is plotted in Figure 4.10b, which shows a very fast switching frequency during the first 4.3 s.

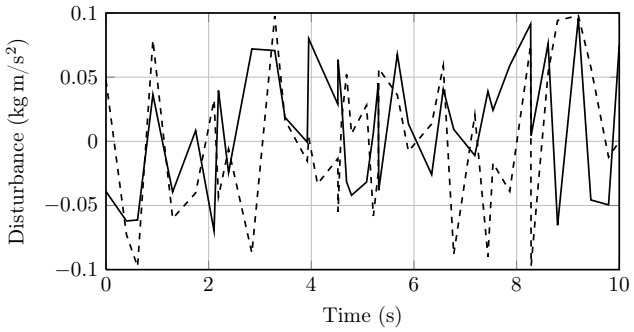
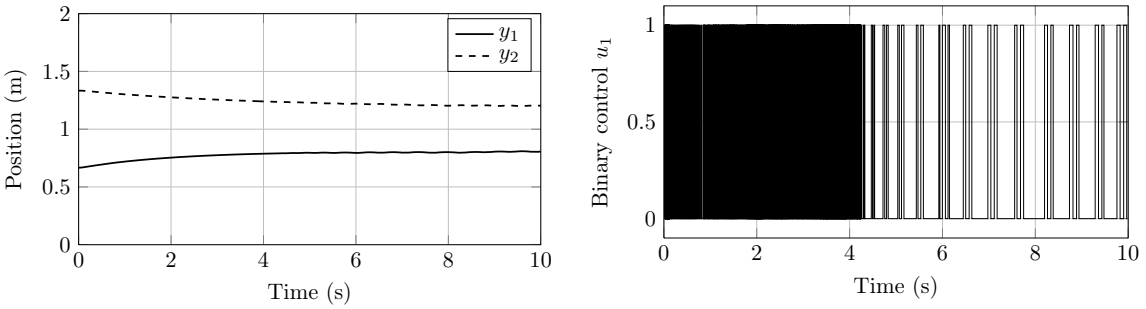


Figure 4.9: Disturbance signals for Example 4.1: both $d_1(\cdot)$ (solid line) and $d_2(\cdot)$ (dashed line) have maximal magnitude $d_{\max} = 0.1$.



(a) The position trajectories.

(b) The binary control signal $u_1(\cdot)$.

Figure 4.10: Simulation results of Example 4.1 with the basic feedback scheduling strategy: the sliding mode control effect is clearly visible during the first $4.3\,\mathrm{s}$ where u_1 switches very fast.

This is explained by the sliding mode control effect as discussed above in limitation number 3 on page 108. Figure 4.11 on the following page illustrates the phase plot of the system, projected to the output space (y_1, y_2). The gray-filled square is the safe set `Safe` and the ellipsoid inside it is the attracting ball $\mathfrak{B}_M(x_c, \alpha)$. Clearly, the system is safe. □

4.3.3. Event-triggered Feedback Scheduling

As illustrated in the previous example, the basic feedback scheduling strategy usually causes very fast switching frequencies due to the sliding mode control effect. Similar to robust periodic scheduling (which also usually induces frequent switching), an event-triggered scheduling algorithm can be used to alleviate this problem. Furthermore, it also improves upon the basic strategy as it does not require solving the control law in Equation (4.21) in continuous time.

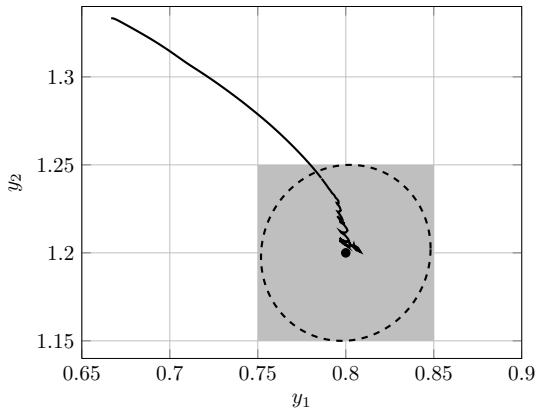


Figure 4.11: Phase plot for Example 4.1: projected to the output space (y_1, y_2), the trajectory (solid line) slides to the attracting ball (ellipsoid drawn in dashed line) inside the safe set `Safe` (gray-filled square).

Consider Theorem 4.2 on page 104 – the main theorem behind the basic scheduling strategy. Essentially, the conditions in Equations (4.15a) to (4.15c) ensure that whenever $x(t)$ is outside the attracting set $\mathcal{A}$, there exists a control such that the function $V(x(t)) = (x(t) - x_c)^T M(x(t) - x_c)$ always decays, along the flow of the system, with a rate at least $-\gamma$, where $\gamma > 0$ is the minimal decaying rate. The feedback law in Equation (4.21) simply finds a control that always reduces $V(x(t))$ at the fastest rate possible. However, it is certainly sufficient to use a control that reduces $V(x(t))$ at a rate not slower than γ. That is, as long as the currently used control, denoted $u^\star$, satisfies

$$\sup_{d \in \mathcal{D}} \dot{V}(x(t)) = \max_{d \in \mathcal{D}} 2(x(t) - x_c)^T M(Ax(t) + B_0 + Bu^\star + Wd) \leq -\gamma \tag{4.22}$$

then we do not need to compute and switch to a new control. This observation leads to an event-triggered scheme where we only switch the control when it does not satisfy inequality (4.22) at the current state. Let t be the current time and $x^\star = x(t)$ be the current state. Instead of solving the optimization (4.22) in continuous time, we find a ball $\mathfrak{B}_M(x^\star, r)$ with radius $r > 0$ around the current state so that for all $x \in \mathfrak{B}_M(x^\star, r)$,

$$\max_{d \in \mathcal{D}} 2(x - x_c)^T M(Ax + B_0 + Bu^\star + Wd) \leq -\gamma. \tag{4.23}$$

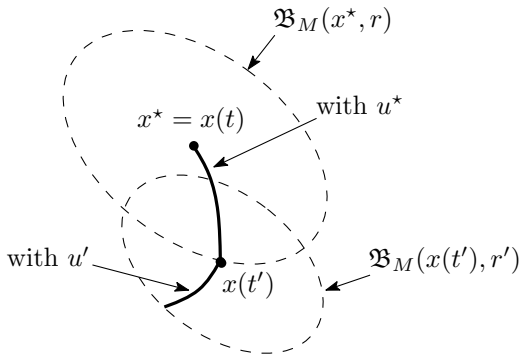


Figure 4.12: Illustration of the event-triggered scheduling strategy based on attracting sets. The solid lines represent the system's trajectory. At time t, the ball $\mathfrak{B}_M(x^\star, r)$ is constructed and the current control $u^\star$ is used until time $t' \geq t$ when the event $\|x(t') - x^\star\|_M = r$ is detected. At this time, a new control u' is determined and a new ball $\mathfrak{B}_M(x(t'), r')$ is constructed. The event-triggered scheme is then repeated.

Once r is determined, we can keep the current control $u(t') = u^\star$ for $t' \geq t$ as long as $x(t') \in \mathfrak{B}_M(x^\star, r)$ and only compute a new control when the event $\|x(t') - x^\star\|_M = r$ is detected. This event-triggered scheme is illustrated in Figure 4.12 on this page. The only remaining step is to calculate the radius r. In the rest of this subsection, we will present two methods to estimate r.

First Method to Estimate Radius r

The inequality (4.23) is equivalent to

$$2(x - x_c)^T MA(x - x_c) + \max_{d \in \mathcal{D}} 2(x - x_c)^T M(Ax_c + B_0 + Bu^\star + Wd) \leq -\gamma, \forall x \in \mathfrak{B}_M(x^\star, r).$$

It follows from Equation (4.15b) that the inequality, for all $x \in \mathfrak{B}_M(x^\star, r)$,

$$-2\lambda(x - x_c)^T M(x - x_c) + 2\max_{d \in \mathcal{D}}(x - x_c)^T M(Ax_c + B_0 + Bu^\star + Wd) \leq -\gamma \qquad (4.24)$$

implies the inequality (4.23). Recall that $x^\star$ is the current state as well as the center of the ball $\mathfrak{B}_M(x^\star, r)$ we are looking for. Let $\beta := \|x^\star - x_c\|_M$ be the distance from $x^\star$ to x_c. Because our goal is to drive the state towards an attracting ball centered at x_c, we are only interested in $r \leq \beta$. Since $\|x - x^\star\|_M \leq r$, we have $(x - x_c)^T M(x - x_c) = \|x - x_c\|_M^2 \geq (\beta - r)^2$.

By writing $x - x_c = (x - x^\star) - (x^\star - x_c)$ in the maximization in Equation (4.24), we obtain

$$\begin{aligned}\max_{d\in\mathcal{D}}(x - x_c)^T M(Ax_c + B_0 + Bu^\star + Wd) \leq &\max_{d\in\mathcal{D}}(x - x^\star)^T M(Ax_c + B_0 + Bu^\star + Wd) \\ &+ \max_{d\in\mathcal{D}}(x^\star - x_c)^T M(Ax_c + B_0 + Bu^\star + Wd).\end{aligned}$$

Denote $\theta = \max_{d\in\mathcal{D}}(x^\star - x_c)^T M(Ax_c + B_0 + Bu^\star + Wd)$ which can be computed by solving a linear program (LP). We then have

$$\begin{aligned}&- 2\lambda(x - x_c)^T M(x - x_c) + 2\max_{d\in\mathcal{D}}(x - x_c)^T M(Ax_c + B_0 + Bu^\star + Wd) \\ \leq &- 2\lambda(\beta - r)^2 + 2\theta + 2\max_{d\in\mathcal{D}}(x - x^\star)^T M(Ax_c + B_0 + Bu^\star + Wd).\end{aligned}$$

It follows that if r satisfies

$$\max_{d\in\mathcal{D}}(x - x^\star)^T M(Ax_c + B_0 + Bu^\star + Wd) \leq \lambda(\beta - r)^2 - \theta - \frac{\gamma}{2} \tag{4.25}$$

then the inequality (4.24) is justified. Define

$$\xi = \frac{1}{\lambda}\max_{\|z\|_M=1}\max_{d\in\mathcal{D}} z^T M(Ax_c + B_0 + Bu^\star + Wd) = \frac{1}{\lambda}\max_{d\in\mathcal{D}}\|Ax_c + B_0 + Bu^\star + Wd\|_M$$

which can be calculated by solving a QP. We then have that for all $x \in \mathbb{R}^n$

$$\max_{d\in\mathcal{D}}(x - x^\star)^T M(Ax_c + B_0 + Bu^\star + Wd) \leq \lambda\xi\|x - x^\star\|_M \leq \lambda\xi r.$$

Using the obtained inequality in Equation (4.25), we deduce that the inequality (4.24) is implied by the inequality $\lambda\xi r \leq \lambda(\beta - r)^2 - \theta - \frac{\gamma}{2}$, or equivalently $r^2 - (2\beta + \xi)r + (\beta^2 - \frac{\theta}{\lambda} - \frac{\gamma}{2\lambda}) \geq 0$. By calculating the roots of the quadratic equation in r, it is straightforward to verify that

$$r = \frac{1}{2}\left(2\beta + \xi - \sqrt{\xi^2 + 4\beta\xi + \frac{4\theta}{\lambda} + \frac{2\gamma}{\lambda}}\right) = \beta + \frac{1}{2}\left(\xi - \sqrt{\xi^2 + 4\beta\xi + \frac{4\theta}{\lambda} + \frac{2\gamma}{\lambda}}\right) \tag{4.26}$$

satisfies the previous inequality, hence it satisfies the main inequality (4.23). Note that obviously $r \leq \beta$.

Second Method to Estimate Radius r

The left hand side of Equation (4.23) can be written as

$$\begin{aligned}
&\max_{d\in\mathcal{D}} 2(x-x_c)^T M(Ax+B_0+Bu^\star+Wd) \\
=&2\max_{d\in\mathcal{D}}((x-x^\star)+(x^\star-x_c))^T M(A(x-x^\star)+Ax^\star+B_0+Bu^\star+Wd) \\
\leq& -2\lambda(x-x^\star)^T M(x-x^\star)+2(x^\star-x_c)^T MA(x-x^\star) \\
&+2\max_{d\in\mathcal{D}}(x^\star-x_c)^T M(Ax^\star+B_0+Bu^\star+Wd) \\
&+2\max_{d\in\mathcal{D}}(x-x^\star)^T M(Ax^\star+B_0+Bu^\star+Wd).
\end{aligned} \tag{4.27}$$

We have that $\lambda(x-x^\star)^T M(x-x^\star) \geq 0$ and

$$\begin{aligned}
(x^\star-x_c)^T MA(x-x^\star) &\leq \left\|(R^{-1})^T A^T M(x^\star-x_c)\right\|_2 \|R(x-x^\star)\|_2 \\
&\leq r\left\|(R^{-1})^T A^T M(x^\star-x_c)\right\|_2
\end{aligned}$$

in which non-singular matrix R satisfying $R^T R = M$ is determined by the Cholesky decomposition of M (see Strang, 2006). Define $\zeta = \max_{d\in\mathcal{D}}(x^\star-x_c)^T M(Ax^\star+B_0+Bu^\star+Wd)$ and

$$\chi = \max_{\|z\|_M=1}\max_{d\in\mathcal{D}} z^T M(Ax^\star+B_0+Bu^\star+Wd) = \max_{d\in\mathcal{D}}\|Ax^\star+B_0+Bu^\star+Wd\|_M \tag{4.28}$$

which can be computed by solving a LP and a QP respectively. We can bound

$$\max_{d\in\mathcal{D}}(x-x^\star)^T M(Ax^\star+B_0+Bu^\star+Wd) \leq \chi\|x-x^\star\|_M \leq \chi r.$$

Using these inequalities in Equation (4.27), it follows that if r is such that

$$r \left\| (R^{-1})^T A^T M (x^\star - x_c) \right\|_2 + \chi r + \zeta \leq -\frac{\gamma}{2}$$

then the inequality (4.23) holds. Therefore we can estimate r as

$$r = -\frac{\frac{\gamma}{2} + \zeta}{\|(R^{-1})^T A^T M (x^\star - x_c)\|_2 + \chi}. \tag{4.29}$$

Note that although the expression of r has a negation sign, it is usually that $\zeta < -\frac{\gamma}{2}$, hence r is usually positive.

The Event-triggered Scheduling Algorithm

Having calculated the radius r of the event $\|x(t') - x^\star\|_M = r$, we can now present the event-triggered scheduling algorithm, which is given in Algorithm 4.3 on the next page. The main loop of the algorithm determines the control $u^\star$ only when the state $x(t)$ is outside the attracting set $\mathcal{A}$. Inside the loop, a new control is computed if the current control does not robustly reduce $V(x(t))$, i.e., $u^\star$ does not satisfy the inequality in Equation (4.22); otherwise the current control is re-used. Then the radius r is calculated, either by Equation (4.26) or Equation (4.29), or by taking the maximum of them. For practical purposes, we use a lower bound $r_{\min} > 0$ for r, which can be chosen very small. If the calculated r is less than $r_{\min}$, we set r to $r_{\min}$. Finally, the state $x(t)$ is monitored continuously until the event $\|x(t) - x^\star\|_M = r$ is detected, after which the steps are repeated.

4.3.4. Self-triggered Feedback Scheduling

Similar to the feedback scheduling strategies based on periodic scheduling (Section 4.2), we can derive a self-triggered scheduling scheme from the event-triggered scheduling algorithm in Section 4.3.3 to avoid continuously monitoring the state. As we discussed in Section 4.2.3 on page 95, dedicated hardware is required to track the system's state so that the event detection can be carried out. A self-triggered scheduling implementation does not require

Algorithm 4.3 Event-triggered Feedback Scheduling Based on Attracting Sets

Obtain $x(0)$
$u^\star \leftarrow \arg\min_{u\in\mathcal{U}}(x(0)-x_c)^T MBu$
while $\|x(t)-x_c\|_M \geq \alpha$ **do** ▷ when outside $\mathcal{A}$
    if $u^\star$ does not satisfy Equation (4.22) **then**
        $u^\star \leftarrow \arg\min_{u\in\mathcal{U}}(x(t)-x_c)^T MBu$ ▷ new control
    end if
    $x^\star \leftarrow x(t)$ ▷ center of event ball $\mathfrak{B}_M(x^\star, r)$
    Calculate r (by Equations (4.26) or (4.29) or max of them)
    if $r < r_{\min}$ **then**
        $r \leftarrow r_{\min}$
    end if
    repeat ▷ monitor state and detect event
        Monitor $x(t)$
    until $\|x(t)-x^\star\|_M = r$
end while

such hardware and is therefore more attractive in practice.

Recall that we denoted the current time by t and the current state by $x^\star = x(t)$. The fundamental idea is the same as in Section 4.2.3: we obtain a lower bound estimate of the time delay until the event $\|x(t')-x^\star\|_M = r$ might happen, and only measure $x(t')$ and decide whether to re-compute the control after that minimal time delay. From Theorem 4.1, we can bound the distance $\|x(t')-x^\star\|_M$ for all $t' \geq t$, under the current control $u^\star$, as

$$\left\|x(t')-x^\star\right\|_M \leq \frac{\chi}{\lambda}\left(1-\mathrm{e}^{-\lambda t}\right)$$

where χ is defined in Equation (4.28). It follows that during the time interval $t' \in [t, t+\tau]$, where $\tau = -\frac{1}{\lambda}\log\left(1-\frac{r\lambda}{\chi}\right)$, the system's state $x(t')$ is guaranteed to stay inside the ball $\mathfrak{B}_M(x^\star, r)$. For practical purposes, if τ is shorter than a predefined minimal delay $\tau_{\min}$ (e.g., the timing resolution of the implementation platform) then we delay for $\tau_{\min}$ instead of τ.

Putting everything together, the self-triggered scheduling algorithm is detailed in Algorithm 4.4 on the next page. It is similar to Algorithm 4.3, except that the event detection mechanism is replaced by the calculation of τ and the time delay.

Algorithm 4.4 Self-triggered Feedback Scheduling Based on Attracting Sets

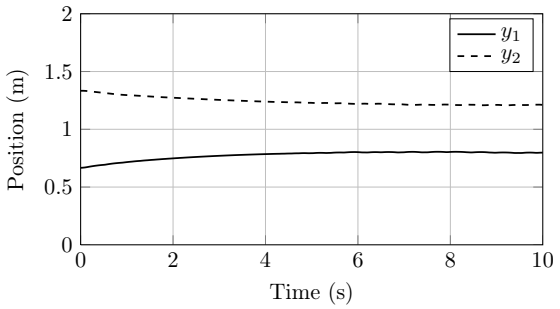
Obtain $x(0)$
$u^\star \leftarrow \arg\min_{u\in\mathcal{U}}(x(0)-x_c)^T MBu$
while $\|x(t)-x_c\|_M \geq \alpha$ **do** ▷ when outside $\mathcal{A}$
  if $u^\star$ does not satisfy Equation (4.22) **then**
    $u^\star \leftarrow \arg\min_{u\in\mathcal{U}}(x(t)-x_c)^T MBu$ ▷ new control
  end if
  $x^\star \leftarrow x(t)$ ▷ center of event ball $\mathfrak{B}_M(x^\star, r)$
  Calculate r (by Equations (4.26) or (4.29) or max of them)
  if $r < r_{\min}$ **then**
    $r \leftarrow r_{\min}$
  end if
  $\tau \leftarrow \max\Big(-\frac{1}{\lambda}\log\Big(1-\frac{r\lambda}{\chi}\Big), \tau_{\min}\Big)$ ▷ time delay for self-triggered mechanism
  Delay for τ time units
end while

4.3.5. Illustrative Examples

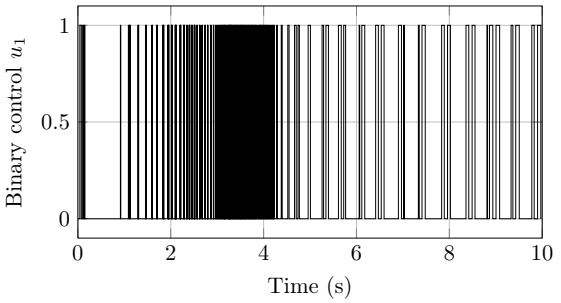
Example 4.2 In this example, we continue Example 4.1 on page 109 and implement the event-triggered and self-triggered scheduling strategies for the mass-spring-damper system. The position trajectories $y_1(\cdot)$ and $y_2(\cdot)$ of the two masses are plotted in Figure 4.13a on the next page, while the control signal $u_1(\cdot)$ is plotted in Figure 4.13b. It is clear that with the event-triggered scheduling scheme, the switching frequency of u_1 is reduced significantly compared to the basic scheduling strategy (cf. Figure 4.10 on page 110). In Figure 4.14 on the next page are the position trajectories and the control signal $u_1(\cdot)$ resulted from simulation of the self-triggered scheduling algorithm. They are comparable to those of the event-triggered scheduling strategy because the main difference between the two schemes is the replacement of the event detection mechanism by the time delay mechanism. Figure 4.15 on page 119 displays the time delay values τ calculated by the self-triggered scheduling algorithm over time. As can be seen in the graph, the time delay is as long as approximately 350 ms and is over 20 ms for the most part. Considering that on average the scheduling computation took about 2.5 ms to complete[5], it is reasonably practicable. □

Example 4.3 Consider the room-heater example with 6 rooms and 6 heaters (cf. Sec-

[5] On MATLAB™ with YALMIP for solving optimizations.

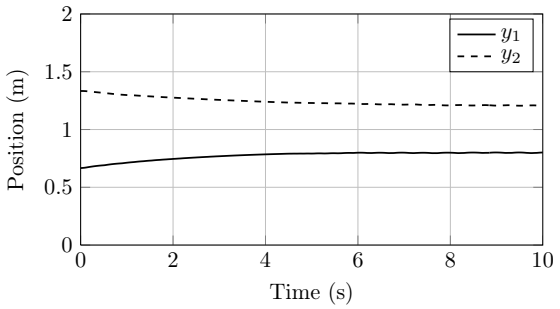


(a) The position trajectories.

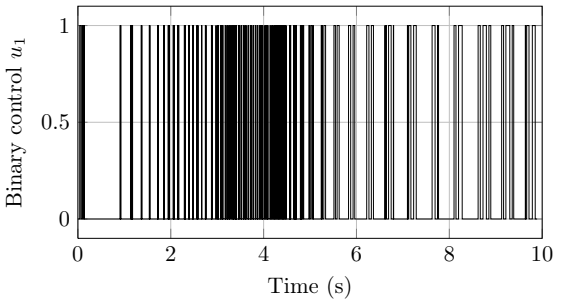


(b) The binary control signal $u_1(\cdot)$.

Figure 4.13: Simulation results for the mass-spring-damper system in Example 4.2 with event-triggered feedback scheduling. Compared to the basic feedback scheduling strategy (cf. Figure 4.10 on page 110), it significantly reduces the switching frequencies of the control inputs.



(a) The position trajectories.



(b) The binary control signal $u_1(\cdot)$.

Figure 4.14: Simulation results for the mass-spring-damper system in Example 4.2 with self-triggered feedback scheduling. Similar to the event-triggered scheduling strategy, it significantly reduces the switching frequencies of the control inputs compared to the basic strategy.

tion 1.2.1). Unlike the example in Section 4.2.4 for feedback scheduling based on periodic scheduling, in this example we allow the disturbances to vary in much larger ranges. Specifically, the ambient air temperature T_a can vary between $T_{a,\min} = 2\,^\circ\mathrm{C}$ and $T_{a,\max} = 12\,^\circ\mathrm{C}$ (compared to $5\,^\circ\mathrm{C}$ to $6\,^\circ\mathrm{C}$ in Section 4.2.4). For each room $i = 1, \ldots, 6$, the room's heat gain $Q_{g,i}$ can take any value in $0\,\mathrm{kW}$ to $0.5\,\mathrm{kW}$ (compared to $0\,\mathrm{kW}$ to $0.05\,\mathrm{kW}$ in Section 4.2.4). The desired temperature bounds are $l_i = 20\,^\circ\mathrm{C}$ and $h_i = 24\,^\circ\mathrm{C}$ for all $i = 1, \ldots, 6$.

In this example, we implemented the self-triggered scheduling strategy for the heaters. The ambient air temperature profile used for simulation is depicted in Figure 4.16 on page 120. Similar to the example in Section 4.2.4, the rooms' heat gains are generated randomly. The resulted temperature of room 1 (T_1) and the schedule of heater 1 (u_1) are displayed in

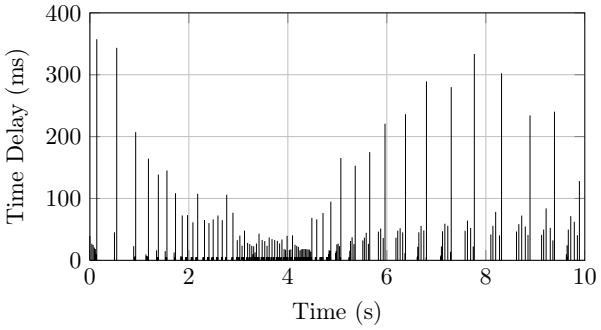


Figure 4.15: Time delays calculated by the self-triggered scheduling algorithm for the mass-spring-damper system in Example 4.2. They could be as long as about 350 ms.

Figures 4.17a and 4.17b on the following page. Evidently T_1 was safe.

For comparison, we also simulated the system with the uncoordinated scheduling strategy (see Section 1.2.1). Figures 4.18a and 4.18b on page 121 plot the temperature of room 1 and the schedule of heater 1 from the simulation. The switching frequencies of the two scheduling strategies are summarized and compared in Table 4.3 on the following page. The second column reports the total numbers of switching of all six heaters. The last three columns list the maximal, minimal and averaged time intervals between consecutive switching of heater 1. Observe that heater 1 was switched less frequently by the uncoordinated scheduler than by the self-triggered scheduler. This is because the self-triggered scheduler had to maintain the peak constraint $k = 4$ as well as the robustness of the system (with respect to disturbances) while it did not continuously monitor the system's state. The peak constraint is reflected clearly in Figure 4.19 on page 121, which compares the energy demand curves of both scheduling strategies. The peak power demand of the self-triggered scheduler is capped at 27 kW while it is 39 kW for the uncoordinated case. Their total energy consumption are almost the same: 94.03 kW for uncoordinated scheduling and 93.01 kW for self-triggered scheduling. □

Example 4.4 To demonstrate the scalability of the scheduling algorithms developed in this section, we consider the room-heater system as in Example 4.3 but scale it to 50 rooms and 50 heaters. We simulated the system with self-triggered scheduling and with uncoordinated scheduling. Their energy demand curves are plotted in Figure 4.20 on page 122. Compared

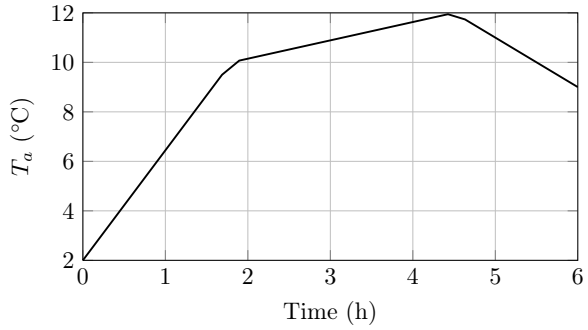
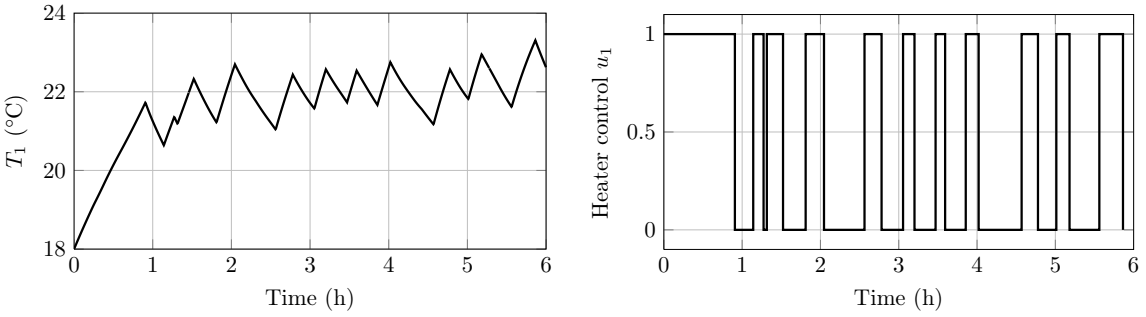


Figure 4.16: Ambient air temperature profile for Example 4.3



(a) Temperature of room 1.

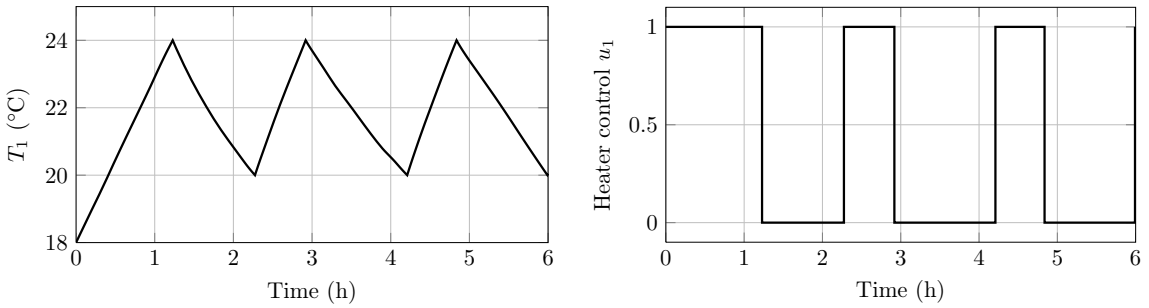
(b) Schedule $u_1(\cdot)$ of heater 1.

Figure 4.17: Simulation results of the self-triggered feedback scheduling strategy for the room-heater system in Example 4.3. The temperature of room 1 (left) is driven to and maintained inside the desired range $[20, 24]$°C. The schedule of heater 1 (right) does not switch too frequently.

to uncoordinated scheduling, self-triggered scheduling reduced the peak power demand by 17.66%, from 351 kW down to 289 kW. The computation of the self-triggered scheduling algorithm took only 29.24 ms on average to complete. Therefore, we can conclude that it is scalable, at least for medium-scale systems. □

Table 4.3: Switching frequency results for the room-heater system in Example 4.3.

	Total No.	Interval between switching of u_1 (min)		
	of switching	Max	Min	Average
Uncoordinated	17	136.46	106.95	119.83
Self-triggered	60	68.49	10.43	33.37



(a) Temperature of room 1.

(b) Schedule $u_1(\cdot)$ of heater 1.

Figure 4.18: Simulation results of uncoordinated scheduling for the room-heater system in Example 4.3. Compared to the self-triggered scheduling strategy (Figure 4.17), heater 1 (b) was switched less frequently.

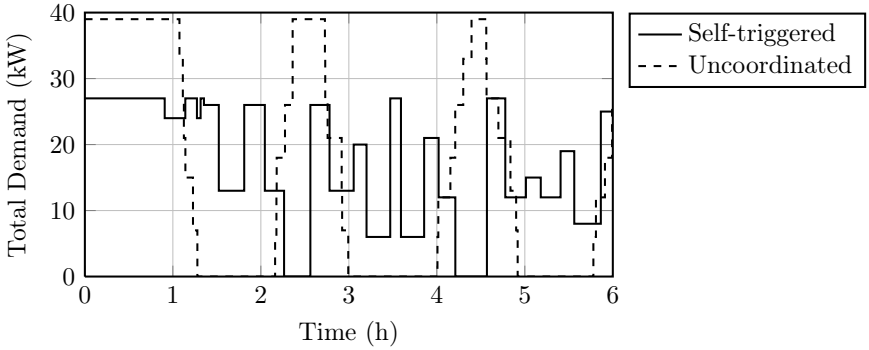


Figure 4.19: Comparison of the energy demand curves of uncoordinated scheduling and self-triggered scheduling for Example 4.3: the curve for uncoordinated scheduling (dashed line) fluctuates between 0 kW and 39 kW, while that for self-triggered scheduling (solid line) is curbed at 27 kW.

4.3.6. Improve Feedback Scheduling with Disturbance Prediction

Let us examine Example 4.3 on page 117 of the room-heater system more closely. From Figure 4.16 on the preceding page, we observe that after about 1.5 h, the ambient air temperature T_a rises significantly to about 9 °C from 2 °C at the beginning, and then only varies between 9 °C and 12 °C for more than 6 hours. Hence, heat loss from the rooms to the ambient environment decreases, which reduces the heating requirement from the heaters. This change in T_a, as a disturbance to the system, has two consequences: (a) the peak constraint k can be reduced for the duration when T_a is high; and (b) the event- and self-triggered scheduling algorithms can result in longer delays (hence fewer switching) considering a smaller constraint set of T_a for the last 6 hours. The first consequence can

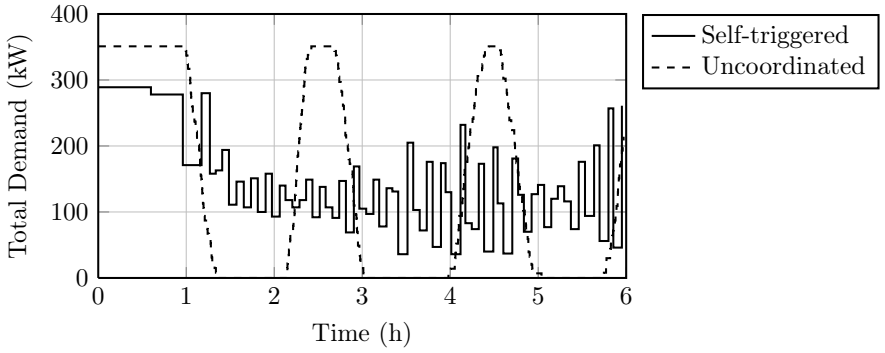


Figure 4.20: Comparison of the energy demand curves of uncoordinated scheduling and self-triggered scheduling for the large-scale room-heater example in Example 4.4: the peak for uncoordinated scheduling (dashed line) is 351 kW, while the peak for self-triggered scheduling (solid line) is 289 kW.

be seen clearly in the demand curve plot in Figure 4.19 on the previous page: during the last 6 hours, the power demand of self-triggered scheduling fluctuated between 0 kW and 27 kW. If the peak constraint k were reduced, e.g., $k = 3$, then the peak demand during these hours would have been reduced and the demand curve would have not fluctuated too much. The same observation can be made of Figure 4.20 for the large-scale room-heater system in Example 4.4.

It follows from the above observations that if short-term predictions of the disturbances are available, they can be incorporated into the scheduling algorithms to improve Green Scheduling. Specifically, suppose that at any time $t \geq 0$, a prediction of the disturbance constraint set can be obtained for a finite time horizon $h > 0$, that is $d(\tau) \in \mathcal{D}_{[t,t+h]} \subseteq \mathbb{R}^q$ $\forall \tau \in [t, t+h]$, where $\mathcal{D}_{[t,t+h]}$ is known. Then it is straightforward to modify the scheduling algorithms (Algorithms 4.3 and 4.4) to exploit this new information as follows:

- Replace $\mathcal{D}$ by $\mathcal{D}_{[t,t+h]}$ in all calculations; and
- Recompute the peak constraint k at regular intervals which are shorter than the prediction horizon h, for instance every half hour in the room-heater example.

Example 4.5 We continue Example 4.3 but use the above scheme with disturbance prediction. We assume that a one-hour prediction (i.e., $h = 1\,\text{h}$) is available at any time t. Furthermore, every 30 minutes, the peak constraint k is recalculated, taking into account the

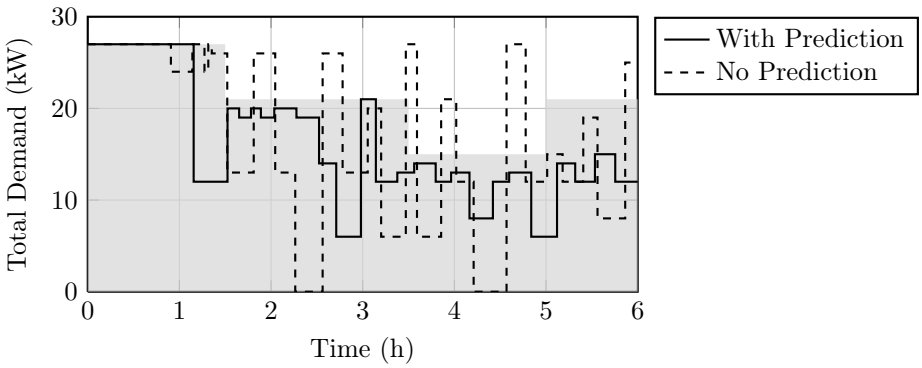


Figure 4.21: Comparison of the energy demand curves of self-triggered scheduling with and without disturbance prediction for Example 4.5: the improved scheduling algorithm with disturbance prediction achieved lower peak demand after 1.5 h.

disturbance prediction. It turned out that the peak constraint k is 4 between 0 h and 1.5 h, 3 between 1.5 h and 3.5 h, 2 between 3.5 h and 5 h, and 3 between 5 h and 6 h. A MATLAB™ simulation was carried out with the same ambient air temperature profile (Figure 4.16) and the same heat gain profile for each room. The demand curves of self-triggered scheduling with and without disturbance prediction are plotted in Figure 4.21 on this page, where the time-varying peak constraint for the case with prediction is illustrated by the gray-filled region in the background. Evidently, with disturbance prediction, the peak energy demand was reduced after 1.5 h and the demand curve became less fluctuating. In Table 4.4 on the next page we report the resulted switching frequencies of the two cases. The second column reports the total numbers of switching of all six heaters. The last three columns list the maximal, minimal and averaged time intervals between consecutive switching of heater 1. It can clearly be seen that actuation switching of the improved scheduling algorithm with disturbance prediction was less frequent as the delays between consecutive switching increased. □

4.4. Feedback Scheduling Based on Backward Reachability

In Chapter 3 and Sections 4.2 and 4.3, we studied scheduling strategies which are based on the techniques used for the schedulability analysis in Chapter 2, namely periodic scheduling and attracting sets. In this section, we will take a different approach and develop

Table 4.4: Switching frequency results for the room-heater system in Example 4.5 where disturbance prediction is utilized.

	Total No. of switching	Interval between switching of u_1 (min)		
		Max	Min	Average
Without prediction	60	68.49	10.43	33.37
With prediction	50	91.50	15.25	36.90

a feedback scheduling strategy for discrete-time systems based on backward reachability analysis. Reachability analysis is extensively used in the literature for safety verification and control synthesis for dynamical systems, especially for switched and hybrid systems (for introductions on this subject see (Maler, 2008; Mitchell, 2007)). In the next subsection, we redefine the green scheduling problem for discrete-time systems and introduce the notion of backward reachability. We will sketch our method for synthesizing discrete-time green scheduling strategies in Section 4.4.2. Section 4.4.3 studies robust periodic invariant sets – an important tool in our synthesis. The green scheduling synthesis is presented in Section 4.4.4 and an illustrative example is given in Section 4.4.5. Finally, we discuss the advantages and limitations of this synthesis method in Section 4.4.6.

4.4.1. Discrete-time Green Scheduling and Backward Reachability

System's Dynamics and Green Scheduling Problem

In this section, we will work with discrete time, hence the time variable t can only receive integer values, i.e., $t \in \mathbb{N} = \{0, 1, 2, \dots\}$. Consider a discrete-time control system with disturbances

$$x(t+1) = f(x(t), u(t), d(t)) = Ax(t) + B_0 + Bu(t) + Wd(t), \qquad \forall t \in \mathbb{N} \tag{4.30}$$

where variables x, u, d and parameters A, B_0, B, W have the usual interpretations (cf. Chapter 1 on page 1) For example, the discrete-time dynamics (4.30) can be the result of discretization of the continuous-time control system in Equation (1.9) on page 12.

We generalize the constraints of the Green Scheduling problem by assuming that at any time step $t \in \mathbb{N}$, the disturbances d belong to a time-varying bounded set $\mathcal{D}_t \subset \mathbb{R}^q$ and the control inputs u are constrained to a time-varying set $\mathcal{U}_t \subset \{0,1\}^m$ (which is finite because u is binary). This generalization allows us to consider disturbance prediction (e.g., forecast of time-varying ambient air temperature) and time-varying peak constraint on the control inputs. We assume that these constraints are periodic with time period $\delta \in \{1, 2, \dots\}$, meaning that $\mathcal{D}_{t+\delta} = \mathcal{D}_t$ and $\mathcal{U}_{t+\delta} = \mathcal{U}_t$ for all t. Similarly, instead of a constant desired safe set `Safe`, we also allow it to be time-varying and periodic: the desired safe set at time $t \in \mathbb{N}$ is $\texttt{Safe}_t \subset \mathbb{R}^n$ and $\texttt{Safe}_{t+\delta} = \texttt{Safe}_t$, $\forall t$. Obviously, the case with fixed disturbance set $\mathcal{D}$, fixed valid control set $\mathcal{U}$ and fixed safe set `Safe` is a special case with $\delta = 1$.

Because we are now working in discrete time, all the involved signals are discrete-time. An admissible disturbance signal $d(\cdot)$ is a sequence of values $d(0), d(1), \dots$ where $d(t) \in \mathcal{D}_t$ for $t = 0, 1, \dots$. An admissible control signal (or schedule) $u(\cdot)$ is a sequence $u(0), u(1), \dots$ where $u(t) \in \mathcal{U}_t$ for $t \in \mathbb{N}$. The resulted state trajectory $x(\cdot)$ is a sequence of states $x(0), x(1), \dots$ that satisfy the difference equation (4.30).

Analogously to continuous-time systems (see Chapter 1), we define the notions of safe trajectories and safe feedback scheduling strategies for discrete-time dynamics. A state trajectory is *safe* if after some finite time steps, it is driven to and maintained inside the (time-varying) desired safe set. Specifically, there exists a finite value $\tau \in \mathbb{N}$ such that $x(t) \in \texttt{Safe}_t$ for all $t \geq \tau$. A feedback scheduling strategy $u(t) = \kappa(t, x(t))$, where function $\kappa : \mathbb{N} \times \mathbb{R}^n \to \{0,1\}^m$ specifies the feedback law, is safe (and valid) if the closed-loop system

$$x(t+1) = f(x(t), \kappa(t, x(t)), d(t), \quad \forall t \in \mathbb{N}$$

satisfies, for every initial state $x(0) \in X_0$,

- the resulted state trajectory $x(\cdot)$ is safe; and
- $u(t) = \kappa(t, x(t)) \in \mathcal{U}_t$ for all $t \in \mathbb{N}$.

Our objective is to synthesize safe feedback scheduling strategies for the system (4.30).

Backward Reachability

Backward reachability refers to the type of analysis of a dynamical system in which, starting from a set of states, we follow all possible trajectories of the system backwards in time. Although it seems that reachability is similar to and can be performed by simulation of the system's dynamics, here we are interested in the set of all possible trajectories of the system, not an individual trajectory. Backward reachability will play an important role in the scheduling synthesis presented later, when we determine a set of states from which the system can reach a given set of states (the desired safe set in our case) under an admissible control signal.

Because the system (4.30) is subject to unknown but constrained disturbances d, it must be able to *robustly* reach the desired set of states, i.e., regardless of the disturbances. To this end, we define the **one-step robust backward reachability operator** for time step $t \in \mathbb{N}$ from a set $X \subset \mathbb{R}^n$ as

$$\mathcal{R}_t^{-1}(X) := \{x \in \mathbb{R}^n \,:\, \exists u \in \mathcal{U}_t, f(x,u,d) \in X \;\forall d \in \mathcal{D}_t\}\,. \tag{4.31}$$

Essentially, $\mathcal{R}_t^{-1}(X)$ is the set of all states at time step t from which the system can reach X by a valid control input regardless of the disturbances. We can extend this definition for an interval $[t, t']$ of time steps, $t' \geq t$, as the composition of the operators for each individual time step from t to t'

$$\mathcal{R}_{[t,t']}^{-1}(X) := \left(\mathcal{R}_t^{-1} \circ \cdots \circ \mathcal{R}_{t'}^{-1}\right)(X) \tag{4.32}$$

where the symbol $\circ$ denotes the composition of two operators or functions: $(g \circ h)(x) = g(h(x))$. The operator $\mathcal{R}_t^{-1}$ is obviously a special case: $\mathcal{R}_t^{-1} \equiv \mathcal{R}_{[t,t]}^{-1}$. It is straightforward to verify that we can compose these operators as

$$\mathcal{R}_{[t,t']}^{-1} \equiv \mathcal{R}_{[t,t'']}^{-1} \circ \mathcal{R}_{[t''+1,t']}^{-1} \tag{4.33}$$

where t, t', t'' are integers and $t \leq t'' < t'$. Furthermore, because of the periodicity of the control input set $\mathcal{U}_t$ and the disturbance set $\mathcal{D}_t$, robust backward reachability operators are also periodic, meaning that $\mathcal{R}_t^{-1} \equiv \mathcal{R}_{t+\delta}^{-1}$ and $\mathcal{R}_{[t,t']}^{-1} \equiv \mathcal{R}_{[t+\delta,t'+\delta]}^{-1}$ for all time steps $t \leq t'$.

For wide classes of system dynamics (both discrete-time and continuous-time) and common types of sets, robust backward reachability operators can be computed or approximated. Computational methods for reachability analysis were developed for polytopic sets in (Chutinan and Krogh, 1999, 2003), for ellipsoids in (Kurzhanskiy and Varaiya, 2007), and for zonotopes in (Girard, 2005). Other computational techniques were also proposed, for example using support functions (Guernic and Girard, 2010) and level set methods (Mitchell and Tomlin, 2000; Mitchell et al., 2005). For affine systems being considered in green scheduling, robust backward reachability operators can be computed by

$$\mathcal{R}_t^{-1}(X) = A^{-1}\left(((X \ominus W\mathcal{D}_t) \oplus (-B\mathcal{U}_t)) - B_0\right)$$

in which the symbol $\ominus$ denotes the Pontryagin difference of two sets and the symbol $\oplus$ denotes the Minkowski sum of two sets (cf. page 55). For sets represented by polytopes, these geometric operations can be computed efficiently using available scientific software, e.g., the Multi-Parametric Toolbox in MATLAB™ (Kvasnica et al., 2004) and the Parma Polyhedra Library (BUGSENG, 2012).

4.4.2. Overview of the Scheduling Synthesis

According to the definitions of safety in Section 4.4.1, a safe trajectory typically has two phases:

1. **Convergence:** From the initial state $x(0)$, after a finite number $\tau \in \mathbb{N}$ of time steps, $x(t)$ converges to a set $\mathbb{C}_\tau \subseteq \texttt{Safe}_\tau$, where the set $\mathbb{C}_\tau$ will be determined in the second phase. Not all safe trajectories go through this phase, in which case $\tau = 0$ and $x(0) \in \mathbb{C}_0$. For practical purposes, we assume that a maximal number $T \in \mathbb{N}$ of time steps for this phase is given, and we only consider safe trajectories that complete the

convergence phase in at most T time steps. In scheduling synthesis, robust backward reachability is used for this phase to compute the set of initial states that can reach $\mathbb{C}_\tau$, $\tau \leq T$, as well as the feedback control law to achieve that convergence.

2. **Invariance:** After the state is driven to a desired subset $\mathbb{C}_\tau \subset \texttt{Safe}_\tau$ in the convergence phase, it is maintained in a sequence of sets $\{\mathbb{C}_t\}_{t=\tau}^{\infty}$ where $\mathbb{C}_t \subset \texttt{Safe}_t$, $\forall t \geq \tau$. Because $\mathcal{U}_t$, $\mathcal{D}_t$ and $\texttt{Safe}_t$ are δ-periodic, we will look for a sequence of sets that is also δ-periodic, i.e., $\mathbb{C}_{t+\delta} = \mathbb{C}_t$ for all $t \geq \tau$. Therefore any subsequence of length δ of this sequence is invariant, hence the name of this phase. In scheduling synthesis, for this phase we aim to find the maximal periodic invariant sequence of sets for the system, and derive a feedback control law that maintains the state inside these sets.

In Section 4.4.3 we will study in details the invariance phase, then the convergence phase and the overall scheduling synthesis will be presented in Section 4.4.4.

4.4.3. Robust Periodic Invariance

The idea of periodically invariant sets for finding a control strategy for a periodic linear discrete-time system was proposed by Blanchin and Ukovich (1993). Recently, there has been research work on applying this idea to controlling periodic linear systems (Gondhalekar, 2011; Gondhalekar and Jones, 2011; Zhou et al., 2011), especially Model Predictive Control (MPC) for building systems due to the periodic nature of the disturbances (Ma et al., 2009; Gondhalekar et al., 2010; Ma et al., 2012). In this subsection, we will employ this notion and its computation to determine the sequence $\{\mathbb{C}_t\}_{t=\tau}^{\infty}$ as well as a safe feedback control law for the invariance phase as discussed above. To emphasize the robustness of the control law with respect to disturbances, we will use the term **robust periodic invariant**[6].

We first review the basic definitions and results for robust periodic invariant sequence of sets for constrained systems.

Definition 4.2 *A finite sequence* $\mathbb{S} = \{\mathbb{C}_0, \ldots, \mathbb{C}_{\delta-1}\}$ *of sets* $\mathbb{C}_t \subseteq \texttt{Safe}_t$, $\forall t = 0, \ldots, \delta - 1$,

[6]The terms *periodically invariant* and *periodic controlled invariant* are also used by other authors.

is a ***robust periodic invariant sequence*** *of sets for constrained system* (4.30) *if for every* $t = 0, \ldots, \delta - 1$,

$$\forall x \in \mathbb{C}_t \; \exists u \in \mathcal{U}_t \; : \; f(x, u, d) \in \mathbb{C}_{(t+1) \bmod \delta} \; \forall d \in \mathcal{D}_t.$$

A sequence $\mathbb{S}^\star = \{\mathbb{C}_0^\star, \ldots, \mathbb{C}_{\delta-1}^\star\}$ *is said to be the* ***maximal robust periodic invariant sequence*** *if it is robust periodic invariant and contains all robust periodic invariant sequences* $\{\mathbb{C}_0, \ldots, \mathbb{C}_{\delta-1}\}$ *for the system, meaning that* $\mathbb{C}_t \subseteq \mathbb{C}_t^\star \; \forall t = 0, \ldots, \delta - 1$. □

Using the robust backward reachability operator $\mathcal{R}^{-1}$ defined in Equation (4.31), the maximal robust periodic invariant sequence, if it exists, can be computed by the procedure in Algorithm 4.5 on the following page (cf. Procedure 4.1 in Blanchin and Ukovich, 1993). Starting from the desired safe sets $\{\texttt{Safe}_t\}_{t=0}^{\delta-1}$ for one period, the algorithm essentially compute the one-step backward reachable sets repeatedly until it detects one of the two terminating conditions: (1) a fixed point is found (line 5) in which case the obtained sequence $\{\Omega_0, \ldots, \Omega_{\delta-1}\}$ is maximal; or (2) an empty backward reachable set is found (line 8) in which case the maximal robust periodic invariant sequence does not exist.

One important question regarding Algorithm 4.5 is its termination, that is whether the algorithm will terminate after a finite number of iterations. In general, there is no guarantee that Algorithm 4.5 will terminate. However, it is shown in (Bertsekas, 1972) and in (Blanchin and Ukovich, 1993) that under certain compactness and continuity conditions, convergence of the sequence $\{\Omega_0, \ldots, \Omega_{\delta-1}\}$ to the maximal one can be guaranteed. Specifically, for the affine dynamics in Equation (4.30) and assuming that the sets $\mathcal{U}_t$, $\mathcal{D}_t$, $\texttt{Safe}_t$ are convex and compact polytopes, we can guarantee that (Blanchin and Ukovich, 1993)

- If $\mathbb{S}^\star$ does not exist then Algorithm 4.5 will terminate (by the condition in line 8);
- If $\mathbb{S}^\star$ exists then the sequence $\{\Omega_0, \ldots, \Omega_{\delta-1}\}$ will converge to it, thus we can obtain an arbitrarily close over-approximation of $\mathbb{S}^\star$ by executing the algorithm for a sufficiently large number of iterations.

Algorithm 4.5 Computation of Maximal Robust Periodic Invariant Sequence

Initialize a sequence $\Omega_t \leftarrow \texttt{Safe}_t$, $t = 0, \ldots, \delta - 1$
$t \leftarrow 0$
while true do ▷ main iteration
    Compute $R \leftarrow \mathcal{R}^{-1}_{(t-1) \bmod \delta}(\Omega_{t \bmod \delta}) \bigcap \texttt{Safe}_{(t-1) \bmod \delta}$
    if $t \leq -\delta$ and $\Omega_{(t-1) \bmod \delta} = R$ **then**
        return Found $\mathbb{S}^\star = \{\Omega_0, \ldots, \Omega_{\delta-1}\}$
    end if
    if $R = \emptyset$ **then**
        return "Maximal robust periodic invariant sequence does not exist"
    end if
    $\Omega_{(t-1) \bmod \delta} \leftarrow R$
    $t \leftarrow t - 1$
end while

Suppose that $\mathbb{S}^\star$ exists and we can compute it. Then for any initial state $x(0) \in \mathbb{C}_0^\star$ a feedback control law $u(t) = \kappa(t, x(t))$ that can maintain the system safe indefinitely (i.e., $x(t) \in \texttt{Safe}_t$ for all $t \in \mathbb{N}$) must keep $x(t)$ in $\mathbb{S}^\star$, as verified by the following result.

Proposition 4.2 *(adapted from Blanchin and Ukovich, 1993, Corollary 3.1) A control strategy $u(t) = \kappa(t, x(t))$ maintains the system (4.30) safe indefinitely with the initial set $\mathbb{C}_0^\star$ if and only if, for $x(t) \in \mathbb{C}^\star_{t \bmod \delta}$, it satisfies the conditions*

$$f(x(t), \kappa(t, x(t)), d) \in \mathbb{C}^\star_{(t+1) \bmod \delta}, \qquad \forall d \in \mathcal{D}_t$$
$$\kappa(t, x(t)) \in \mathcal{U}_t$$

for all $t \geq 0$. □

A proof of the result can be found in (Blanchin and Ukovich, 1993).

Once $\mathbb{S}^\star$ is obtained, it is straightforward to derive a control law that satisfies Proposition 4.2 as, for any $t \in \mathbb{N}$ and $x(t) \in \mathbb{C}^\star_{t \bmod \delta}$,

$$\kappa(t, x(t)) = \text{any } u \in \mathcal{U}_t \text{ such that } f(x(t), u, d) \in \mathbb{C}^\star_{(t+1) \bmod \delta} \quad \forall d \in \mathcal{D}_t. \tag{4.34}$$

In practice, the control at each time step is usually chosen so as to optimize some objective

function as suggested by Blanchin and Ukovich (1993), e.g., to reduce the total energy consumption. An MPC strategy can also be formulated for determining the control as

$$\begin{aligned} \underset{u(t),\ldots,u(t+N-1)}{\text{minimize}} \quad & \sum_{i=t}^{t+N-1} c_i(x(i),u(i)) + c_f(x(t+N)) \\ \text{subject to} \quad & u(i) \in \mathcal{U}_i \\ & x(i+1) = f(x(i),u(i),\bar{d}(i)) \\ & f(x(i),u(i),d) \in \mathbb{C}^{\star}_{i \bmod \delta}, \quad \forall d \in \mathcal{D}_i \end{aligned}$$

in which the constraints are satisfied for all $i = t, \ldots, t+N-1$ and

- $t \in \mathbb{N}$ is the current time step;
- $N \in \mathbb{N}$ is a given finite horizon, $N \geq 1$;
- $\bar{d}_i \in \mathcal{D}_i$ is the nominal disturbance at time i, $i = t, \ldots, t+N-1$;
- $c_i(x(i),u(i))$ is the (scalar) cost at time i with respect to state $x(i)$ and control $u(i)$, $i = t, \ldots, t+N-1$;
- $c_f(x(t+N))$ is the (scalar) terminal cost depending on $x(t+N)$.

The optimization is solved for an optimal sequence of controls $u(t), \ldots, u(t+N-1)$ but only $u(t)$ is applied. At the next time step, $t+1$, the optimization is re-formulated and the process is repeated.

Recall that in the *invariance phase* of safe green scheduling (cf. Section 4.4.2), we aim to maintain the system's state in a periodic sequence of subsets of $\texttt{Safe}_t$, $\forall t \geq \tau$. Obviously $\mathbb{S}^{\star}$ is the sequence of sets that we are looking for. Once the state x has been driven to $\mathbb{C}^{\star}_{\tau \bmod \delta}$ at time step τ (in the *convergence phase*), the control law derived above is used to guarantee infinite-horizon safety of the system.

4.4.4. Feedback Scheduling Strategy

In the previous subsection, we have presented a method to determine the desired sequence of sets $\mathbb{S}^\star = \{\mathbb{C}_0^\star, \ldots, \mathbb{C}_{\delta-1}^\star\}$ for the *invariance phase*. In the *convergence phase*, we control the system to drive its state to one of the sets in the sequence. Suppose that at time $\tau \leq T$, where $T \geq 1$ is a given maximal number of time steps for the convergence phase, $x(\tau)$ is guaranteed to be in $\mathbb{C}_{\tau \bmod \delta}^\star$ by an admissible control sequence $u(0), \ldots, u(\tau-1)$ regardless of the disturbances. Evidently, to achieve this guarantee, we must have $x(t) \in \mathcal{R}_{[t,\tau-1]}^{-1}(\mathbb{C}_{\tau \bmod \delta}^\star)$ for every $t = 0, \ldots, \tau-1$. Specifically, the initial state must satisfy $x(0) \in \mathcal{R}_{[0,\tau-1]}^{-1}(\mathbb{C}_{\tau \bmod \delta}^\star)$. Then, similarly to Section 4.4.3, a control law $u(t) = \kappa(t, x(t))$ can be obtained to ensure that x converges to $\mathbb{C}_{\tau \bmod \delta}^\star$ at time τ.

Let us define the sets, for all $\tau = 1, \ldots, T$ and all $t = 0, \ldots, \tau$,

$$\Lambda_t^\tau := \begin{cases} \mathbb{C}_{\tau \bmod \delta}^\star & \text{if } t = \tau \\ \mathcal{R}_{[t,\tau-1]}^{-1}(\mathbb{C}_{\tau \bmod \delta}^\star) & \text{if } t < \tau \end{cases} \tag{4.35}$$

If $x(0) \in \Lambda_0^\tau$ for some τ then the system can be controlled to converge to $\mathbb{S}^\star$ in at most τ time steps by following the sequence of sets $\Lambda_0^\tau, \ldots, \Lambda_{\tau-1}^\tau, \mathbb{C}_{\tau \bmod \delta}^\star$. Furthermore, if the set X_0 of initial states satisfies $X_0 \subseteq \left(\bigcup_{\tau=1}^T \Lambda_0^\tau\right) \cup \mathbb{C}_0^\star$ then for any $x(0) \in X_0$, the convergence phase will take at most T time steps. The scheduling strategy for the convergence phase is summarized in Algorithm 4.6 on the next page. It is guaranteed to terminate after at most T time steps as long as the initial state satisfies the above condition. In line 3, we try to reduce the length of the convergence phase by always looking for the smallest τ such that $x(t) \in \Lambda_t^\tau$. As a consequence, the closed-loop system may start by following the path $\{\Lambda_t^\tau\}_{t=0}^{t'}$ but switch to a shorter path $\{\Lambda_t^{\tau'}\}_{t=t'}^{\tau'-1}$, $\tau' < \tau$, at time step t' where $x(t') \in \Lambda_{t'}^\tau \cap \Lambda_{t'}^{\tau'}$.

The overall feedback scheduling algorithm consists of the algorithms for the two phases and is presented in Algorithm 4.7 on page 134. Note that in lines 6 to 16 we find the smallest number of time steps for the convergence phase.

Algorithm 4.6 Scheduling Algorithm for the Convergence Phase

Require: $x(0) \in \left(\bigcup_{\tau=1}^{T} \Lambda_0^\tau\right) \cup \mathbb{C}_0^\star$

1: $t \leftarrow 0$
2: **while** $x(t) \notin \mathbb{C}^\star_{t \bmod \delta}$ **do**
3: Find minimal τ such that $x(t) \in \Lambda_t^\tau$
4: Determine a control $u(t) \in \mathcal{U}_t$ satisfying $f(x(t), u(t), d) \in \Lambda_{t+1}^\tau$, $\forall d \in \mathcal{D}_t$
5: Apply $u(t)$
6: $t \leftarrow t + 1$
7: **end while**

4.4.5. An Illustrative Example

We illustrate the presented feedback scheduling algorithm through the room-heater running example. Consider the room-heater system in Section 1.2.1 on page 5. However, due to the expensive computational requirement of the synthesis method (about which we will discuss in Section 4.4.6), we downscale the system to 3 rooms and 3 heaters. The parameters of the system are (see Section 1.2.1 for their descriptions):

$$C_1 = 2984, \quad C_2 = 2023, \quad C_3 = 2801, \quad Q_1 = 8, \quad Q_2 = 6, \quad Q_3 = 8$$

$$K = \begin{bmatrix} 0.2447 & 0.1380 & 0.2550 \\ 0.1380 & 0.1861 & 0.1140 \\ 0.2550 & 0.1140 & 0.2521 \end{bmatrix}$$

The continuous-time dynamics is discretized with a sampling time of 15 min to obtain a discrete-time system, $\forall t \in \mathbb{N}$,

$$x(t+1) = \begin{bmatrix} 0.8287 & 0.0355 & 0.0642 \\ 0.0524 & 0.8248 & 0.0437 \\ 0.0684 & 0.0315 & 0.8225 \end{bmatrix} x(t) + \begin{bmatrix} 2.1981 & 0.0500 & 0.0877 \\ 0.0667 & 2.4270 & 0.0590 \\ 0.0877 & 0.0443 & 2.3333 \end{bmatrix} u(t) + \begin{bmatrix} 0.2748 & 0.0083 & 0.0110 & 0.0715 \\ 0.0083 & 0.4045 & 0.0074 & 0.0792 \\ 0.0110 & 0.0074 & 0.2917 & 0.0776 \end{bmatrix} d(t).$$

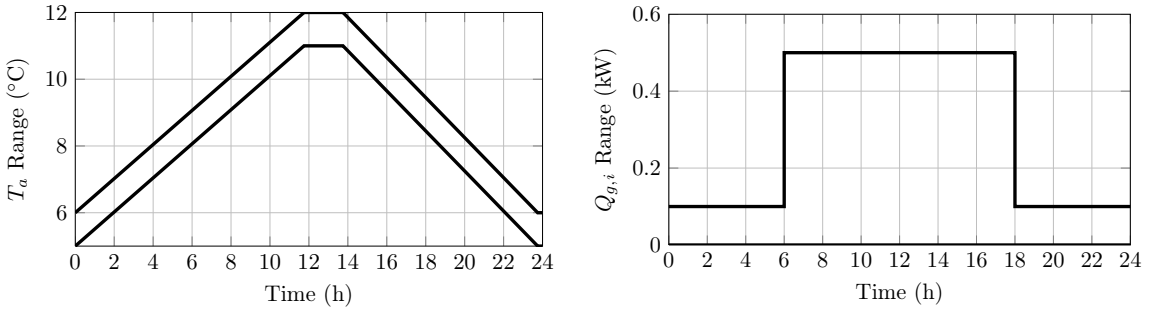
The desired temperature bounds are $l_i = 20\,^\circ\text{C}$ and $h_i = 24\,^\circ\text{C}$ for all rooms and at all time. We assume that the constraints of the disturbances (ambient air temperature T_a and heat gains $Q_{g,i}$) are time-varying and periodic every 24 hours (1 day). The varying range of T_a for 24 hours is provided in Figure 4.22a on the following page. The range of $Q_{g,i}$, for every room

Algorithm 4.7 Feedback Scheduling Algorithm Based on Backward Reachability

Call Algorithm 4.5 to compute $\mathbb{S}^\star$
if $\mathbb{S}^\star$ does not exist **then**
    Stop
end if
if $x(0) \notin \mathbb{C}_0^\star$ **then** ▷ convergence phase
    $\tau \leftarrow 1$
    while $\tau < T$ **do**
        Compute Λ_t^τ, $\forall t = 0, \ldots, \tau$, by Equation (4.35)
        if $x(0) \in \Lambda_0^\tau$ **then**
            Break
        end if
        $\tau \leftarrow \tau + 1$
    end while
    if $\tau = T$ **then**
        return "Cannot converge in at most T time steps"
    end if
    Call Algorithm 4.6
end if
Call scheduling algorithm in the sequence $\mathbb{S}^\star$

$i = 1, 2, 3$, is varied by the level of activity inside the room: $Q_{g,i} \in [0, 0.5]$ (kW) between 6 a.m. and 6 p.m., and $Q_{g,i} \in [0, 0.1]$ (kW) otherwise (see Figure 4.22b for an illustration).

The peak constraint on the control inputs is chosen to be $k = 2$ at all time. Because the sampling time is 15 min, there are 96 time steps in a 24-hour period ($\delta = 96$). The procedure in Algorithm 4.5 to compute the maximal robust periodic invariant sequence $\mathbb{S}^\star$ was implemented in MATLAB™. It took almost 534 s (8 min 54 s) to complete 97 iterations,



(a) Ambient air temperature T_a (b) Heat gain $Q_{g,i}$ for each room i

Figure 4.22: Time-varying and 24-hour-periodic ranges of the disturbances in the example in Section 4.4.5.

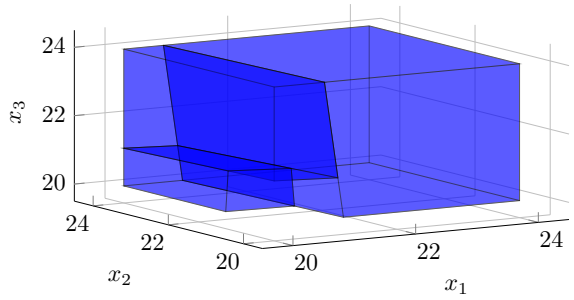


Figure 4.23: The first invariant set $\mathbb{C}_0^\star$ in the maximal robust periodic invariant sequence $\mathbb{S}^\star$ for the example in Section 4.4.5.

at which it successfully computed $\mathbb{S}^\star$. Figure 4.23 on the current page plots the first invariant set $\mathbb{C}_0^\star$ in $\mathbb{S}^\star$.

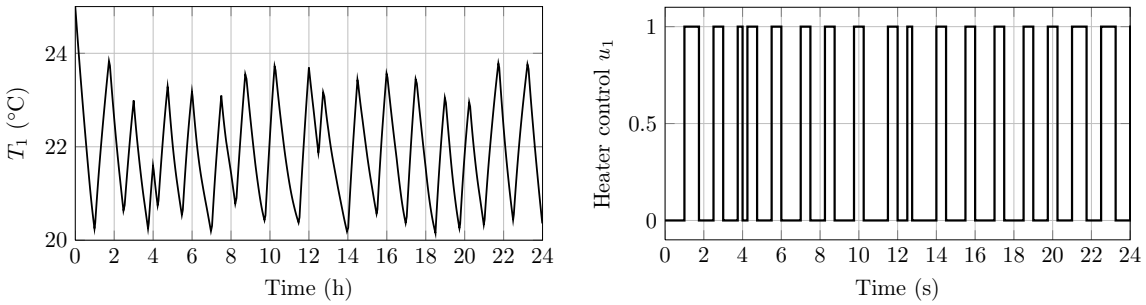
A simulation was carried out for 24 hours with randomly generated ambient air temperature T_a and heat gains $Q_{g,i}$ at each time step. In the scheduling algorithm, at each time step $t \in \mathbb{N}$, a control vector $u(t)$ is selected according to three criteria (in the order of priority):

(1) It must satisfy the conditions in Proposition 4.2 on page 130;

(2) Among all control vectors that satisfy (1), select those that induce the minimal number of switching from the previous time step, i.e., $\|u(t) - u(t-1)\|_1$ is minimized;

(3) Finally, if there are multiple control vectors that meet criteria (1) and (2), select one that demands the least energy, i.e., $[Q_1, Q_2, Q_3] \cdot u(t)$ is minimized.

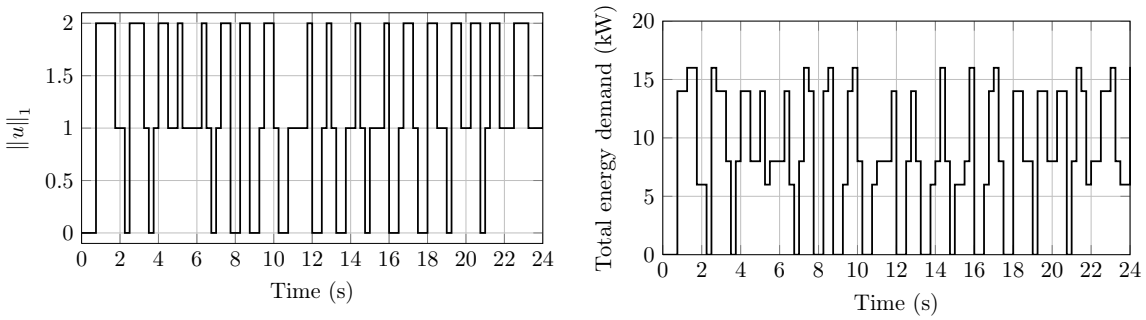
The resulted temperature trajectory of room 1 (T_1) and the control signal of heater 1 (u_1) are displayed in Figure 4.24 on the next page. Evidently T_1 is driven to and maintained in the desired range $[20, 24]$°C. The number of activated heaters at each time step (i.e., $\|u(t)\|_1$) is reported in Figure 4.25a on the following page, while their total energy demand is plotted in Figure 4.25b. It can be seen that at any time step, the number of activated heaters is capped at $k = 2$. The peak energy demand is 16 kW and the total energy consumption is 204 kW h. If the criteria (2) and (3) above are switched, i.e., we prioritize reducing energy demand more important than reducing switching rates, then the number of switching will be



(a) Temperature of room 1

(b) Schedule $u_1(\cdot)$ of heater 1

Figure 4.24: Simulation results of the discrete-time scheduling strategy based on backward reachability for the room-heater example in Section 4.4.5. The temperature of room 1 (left) is maintained inside the desired range $[20, 24]$°C.



(a) Number of activated heaters

(b) Total energy demand curve

Figure 4.25: Energy demand results of the discrete-time scheduling strategy based on backward reachability for the room-heater example in Section 4.4.5. At any time step, the number of activated heaters (left) is capped at $k = 2$.

almost doubled (from 18 to 33 for u_1) while the peak demand will be the same and the total energy consumption will be reduced by only 5.15% (from 204 kW h to 193.5 kW h).

4.4.6. Discussion

As we have shown in the above room-heater example, the feedback scheduling strategy based on backward reachability can take into account time-varying disturbance constraints instead of large ultimate bounds of the disturbances for the entire day. Furthermore, time-varying safe set $\texttt{Safe}_t$ and time-varying control input set $\mathcal{U}_t$ can also be specified to achieve better performance, for example:

- During the on-peak hours, when energy price can be significantly higher than during

the off-peak hours, we can change the safe set $\texttt{Safe}_t$ to allow for more discomfort but less energy demand; we can also change $\mathcal{U}_t$ by reducing the peak constraint k;

- When the rooms are unoccupied at night, we can turn off the system by setting $k = 0$ and changing the safe set $\texttt{Safe}_t$ accordingly, so that the total energy consumption will be reduced.

More sophisticated optimization, for example MPC, can also be implemented on top of this scheduling strategy as discussed at the end of Section 4.4.3. Finally, although we only presented the method for an affine dynamical system in Equation (4.30), it can readily be applied to more complex dynamical systems, e.g., nonlinear systems, as long as the robust backward reachability operator $\mathcal{R}_t^{-1}$ in Equation (4.31) can be computed or approximated (Maler, 2008; Mitchell, 2007).

On the other hand, the previous example also highlights a major drawback of the scheduling approach, namely its highly expensive computational requirement. It can be explained by two factors, assuming $\mathcal{U}_t = \mathcal{U}\ \forall t \in \mathbb{N}$,

- Since the control input set $\mathcal{U}$ is finite, we must enumerate all the valid control vectors in $\mathcal{U}$ in calculating the one-step backward reachability operator $\mathcal{R}_t^{-1}$ and in each iteration of the scheduling algorithms (Equation (4.34) and Algorithm 4.6). Therefore, the computational complexity of these operations increases exponentially with the size of the system, as the cardinality of $\mathcal{U}$ increases exponentially with the number m of control variables.

- More adversely, the discrete nature of $\mathcal{U}$ also causes high complexity in the representation of sets. Consider for example the operator $\mathcal{R}_t^{-1}$ being applied on a set X at time step $t \in \mathbb{N}$. Assume that X is a single convex and compact polytope. Since $\mathcal{R}_t^{-1}(X) = \bigcup_{u \in \mathcal{U}} \{x \in \mathbb{R}^n : f(x, u, d) \in X\ \forall d \in \mathcal{D}_t\}$, it is a (generally non-convex) union of up to $|\mathcal{U}|$ individual polytopes, where $|\mathcal{U}|$ denotes the number of elements (the cardinality) of $\mathcal{U}$. Then $\mathcal{R}_{[t-1,t]}^{-1}(X)$ may contain up to $|\mathcal{U}|^2$ individual polytopes. Although it

is possible to merge[7] the individual polytopes to reduce the size of the non-convex unions, they are still blown up in size quickly. This effect makes set operations, e.g., the Pontryagin difference, significantly more expensive in terms of computation and memory. More details on set operations for computation of reachable and invariant sets and their complexity can be found in (Kerrigan, 2000).

As reported in the room-heater example, the computation of $\mathbb{S}^\star$ took almost 9 min to complete. When we increased the number of rooms and heaters to 6, it did not finish in reasonable time and had to be terminated. Although good programming techniques and libraries may alleviate this issue, e.g., using the C/C++ languages and the Parma Polyhedra Library (BUGSENG, 2012) instead of MATLAB™, the approach is still only applicable to small-scale systems.

4.5. Conclusions

In this chapter, we have developed three state feedback Green Scheduling strategies.

First, the feedforward periodic scheduling strategy studied in Chapter 3 was improved in Section 4.2 to be robust to small disturbances. Event- and self-triggered techniques were then employed to alleviate an issue of robust periodic scheduling, namely the fast switching of actuation. Robust periodic scheduling relies on the existence of a robust tube around the nominal trajectory that bounds the system's trajectory under disturbances (see Figure 4.1 on page 89). This idea is closely related to the notions of approximate simulation and bisimulation relations of dynamical systems, which were proposed and developed in depth by Girard and Pappas (2005, 2007a,b). These notions have been used extensively in verification of control systems and safe controller synthesis (Girard and Pappas, 2006; Fainekos et al., 2006; Julius et al., 2007; Girard, 2012). For an excellent review of approximate bisimulation and its applications in computer science and control theory, we refer the reader to (Girard and Pappas, 2011).

[7] Merging polytopes is a computationally expensive operation.

Robust periodic scheduling strategies can only handle significantly small disturbances, as we showed in the room-heater example in Section 4.2.4. To avoid this drawback, we took another approach in Section 4.3 and developed feedback scheduling strategies based on attracting sets of the control system. Similarly, event- and self-triggered techniques were utilized to reduce the actuation switching rate. As we have mentioned, the results on attracting sets of control systems developed in Section 2.5.2 on page 47 and later in Section 4.3.1 on page 103 are based on the notion of robust control Lyapunov functions, an important analysis tool in control theory (Artstein, 1983; Freeman and Kokotovic, 1996; Liberzon et al., 2002). In addition, the conditions on matrix M and numbers λ and α in Theorem 4.2 on page 104 were inspired by a similar result in Theorem 3 in (Girard and Pappas, 2007a) for approximate bisimulation for constrained linear systems.

In the first two feedback scheduling strategies, event- and self-triggered techniques were used to reduce the rate of switching of the control inputs. Event-triggering has been used in control theory since the end of the nineties for efficient implementations of control laws in situations where limited resources, such as actuation and network communication, are shared among several subsystems (Åström and Bernhardsson, 1999; Otanez et al., 2002; Tabuada, 2007; Heemels et al., 2008; Lunze and Lehmann, 2010). Similar to our results, Lyapunov functions were usually used in the literature for deriving the event-triggering conditions (e.g., Tabuada, 2007; Seuret and Prieur, 2011). However, event-triggered control often requires dedicated hardware to continuously monitor the system's state for event detection. To reduce resource utilization even further, Velasco et al. (2003) proposed self-triggered control in which the controller decides its next execution time based on the current state of the system. Since then, considerable development has been done in the analysis and synthesis of self-triggered control (Lemmon et al., 2007; Mazo et al., 2009; Anta and Tabuada, 2010; Postoyan et al., 2011) and its applications (Dimarogonas et al., 2010; Camacho et al., 2010; Nowzari and Cortés, 2011). Unlike Green Scheduling, most of these work focused on stability of the control system instead of safety.

The third feedback scheduling strategy was developed in Section 4.4 where we considered the Green Scheduling problem in discrete time and investigated a discrete-time scheduling strategy based on backward reachability analysis. This scheduling strategy is capable of handling a large class of system's dynamics as well as time-varying system's specifications such as time-varying disturbance constraint $\mathcal{D}_t$ and time-varying safe set $\texttt{Safe}_t$. In addition, more sophisticated control methods such as MPC can be implemented on top of this strategy. Despite these advantages, it has scalability issue because of its high computational requirement and is thus only applicable to small-scale systems.

This section ends the presentation of all theoretical results developed in this dissertation. We have provided answers to the two important questions raised in Section 1.3.3 on page 16:

1. **Schedulability analysis:** Does there exist a safe schedule or a safe scheduling strategy for a Green Scheduling system?

2. **Scheduling synthesis:** If there does, then how to synthesize a safe schedule or a safe scheduling strategy for the system?

In the next chapter, an application of these theoretical results will be presented for radiant heating and cooling systems in buildings.

Chapter 5

Application in Radiant Heating and Cooling Systems

In Chapters 2 to 4 we have studied the schedulability analysis and scheduling synthesis for the Green Scheduling problem. In this chapter, the theoretical results developed in those chapters will be applied to radiant heating and cooling systems in buildings for peak demand reduction. We first give a brief overview and a model of the radiant heating and cooling systems in Section 5.1, followed by a discussion on the control techniques for these systems in Section 5.2. Application of the Green Scheduling theory to radiant heating and cooling systems will be presented in Section 5.3. Two case studies, one for a hydronic radiant cooling system model in MATLAB™ and one for an electric radiant heating system model in EnergyPlus, will be described in Sections 5.4 and 5.5 respectively. We conclude the chapter with a discussion in Section 5.6.

5.1. Radiant Heating and Cooling Systems

A conventional forced-air heating, ventilating and air conditioning (HVAC) system uses the flow of air to provide thermal comfort within a conditioned space. HVAC systems rely on ductwork, vents, etc. as means of air distribution and use air handlers, filters, blowers, heat exchangers, and various controls to regulate the temperature and flow of air entering a space (ASHRAE, 2009). Radiant heating and cooling systems, henceforward called *radiants* or *radiant systems* for short, serve as an alternative to the conventional forced-air HVAC systems for buildings. In radiants, heat is supplied to or removed from building elements such as floors, ceilings and walls by circulating water, air or electric current through a circuit embedded in or attached to the structure (Olesen, 2002; Watson and Chapman, 2002). When

the radiant heating system is located underneath the floor, it is often called underfloor heating or simply floor heating. Although radiants depend largely on radiant heat transfer between the thermally controlled building elements and the conditioned space, hence their name, they also depend on convection. For instance, in a floor heating application, the natural circulation of heat within a room is caused by heat rising.

The benefits of radiants over forced-air HVAC systems for residential and commercial buildings has been studied well in the literature of energy and buildings (Stetiu, 1999; Watson and Chapman, 2002; Lehmann et al., 2007; Saelens et al., 2011). Essentially, there are three major benefits: human comfort, reduced heat loss, and peak energy demand reduction. Consider the radiant heating system for example.

- **Human comfort.** Human thermal comfort is affected by a number of parameters, as described in the ASHRAE/ANSI Standard 55-2004 (ASHRAE, 2004). An important factor affecting thermal comfort is thermal stratification. It refers to the vertical air temperature difference that results due to the tendency of warm air rising to the ceiling and cool air settling down near the floor. In radiant heating systems, the air mass in the conditioned space is heated to a lower temperature than with a convection heating system as long as the occupants are radiantly heated. Thus, stratification is normally less with radiant systems than convection systems. As illustrated in Figure 5.1 on the following page, in some situations the difference in the air temperature from floor to ceiling can exceed $20°F$ for a forced-air HVAC system, while it is much less in a radiant system. Furthermore, radiants also minimize drafts and dust movements, thereby providing a clean, odorless and quiet operation (Siegenthaler, 2011). Therefore, human comfort is much improved with radiant systems than with conventional HVAC systems.

- **Reduced heat loss.** Stratification also greatly affects building energy use. A high degree of temperature stratification leads to significantly higher air temperatures near the ceiling. This in turn increases the heat loss through the ceiling and hence increases

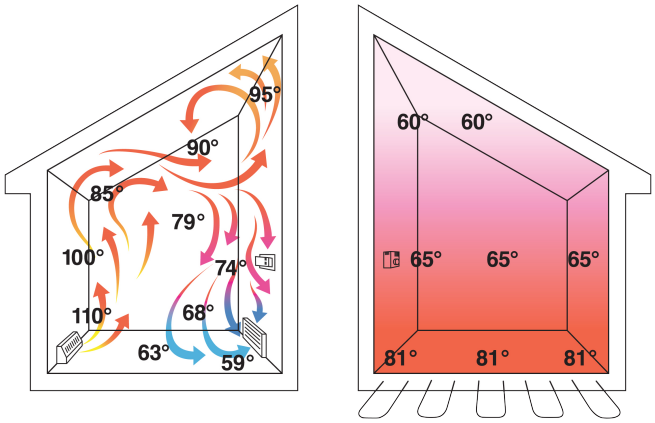


Figure 5.1: Comparison of air temperatures from floor to ceiling for forced-air heating (left) and radiant floor heating (right). *Source: National Association of Home Builders*.

the heating load of the space. As a consequence, radiant systems have much less heat loss than convection systems (Watson and Chapman, 2002).

- **Reduced peak energy demand.** Because the building elements used in radiant systems have high thermal mass, they serve as an energy storage whose slow thermal behaviour is exploited to provide cooling or heating. Hence, the building's thermal mass can be utilized to flatten out peaks in energy demand.

Nowadays, radiant systems are widely used in both commercial and residential buildings in Korea, Germany, Austria, Denmark (Olesen, 2002) and in some parts of the United States (Doebber et al., 2010).

5.1.1. Simple Thermal Model for Radiant Systems

Modern radiant systems use either electrical resistance elements (*electric radiant systems*) or fluid flowing in pipes (*hydronic radiant systems*) to heat or cool the building elements (ASHRAE, 2008; Siegenthaler, 2011). In hydronic radiant systems, hot or cold supply water is pumped through a system of tubes laid in a pattern inside the radiant building elements. Electric radiants typically consist of electric cables built into the slab. Whether electric cables or hydronic tubing, the operation of electric and hydronic radiant systems is similar. In this subsection, we describe a simple thermal model for a hydronic radiant system. The

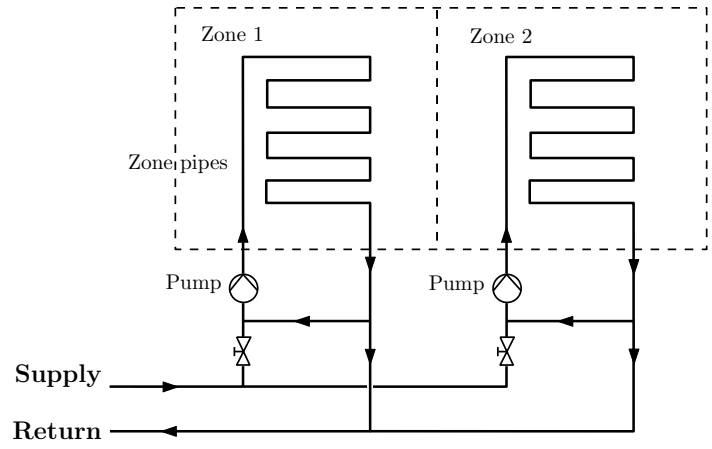


Figure 5.2: Diagram of a hydronic radiant system for two zones.

model will then be used for designing control methods for the radiant system.

Consider a building of two zones[1] equipped with a hydronic radiant system (Figure 5.2). Embedded in a slab under the floor or above the ceiling of each zone is a piping system that carries water from a supply source, such as a boiler system for heating or a chiller system for cooling. The radiant piping systems for the zones are separate, meaning that their water distribution pipes are separate from each other. As illustrated in Figure 5.2, each zone has its own supply and return pipes as well as a circulation pump. This hydronic circuit topology is similar to that used in (Gwerder et al., 2009) and is one of the topologies proposed in (Lehmann et al., 2011).

Koschenz and Dorer (1999) developed a mathematical model for the thermal dynamics of hydronic radiant systems as in Figure 5.2. It is assumed that the slab is uniformly heated and there is no lateral temperature difference or heat transfer. As proposed in (Seem, 1987) and shown in Gwerder et al. (2008), the 3-dimensional heat transfer model in the slab can be reduced to a 1-dimensional model by establishing a correlation between supply water temperature, core temperature (i.e., mean slab temperature in the plane of the piping system) and zone air temperature. For each zone, the thermal dynamics from the supply water temperature to the zone temperature is then modeled using a Resistance–Capacitance (RC)

[1]A zone is defined as a conditioned space controlled by a thermostat.

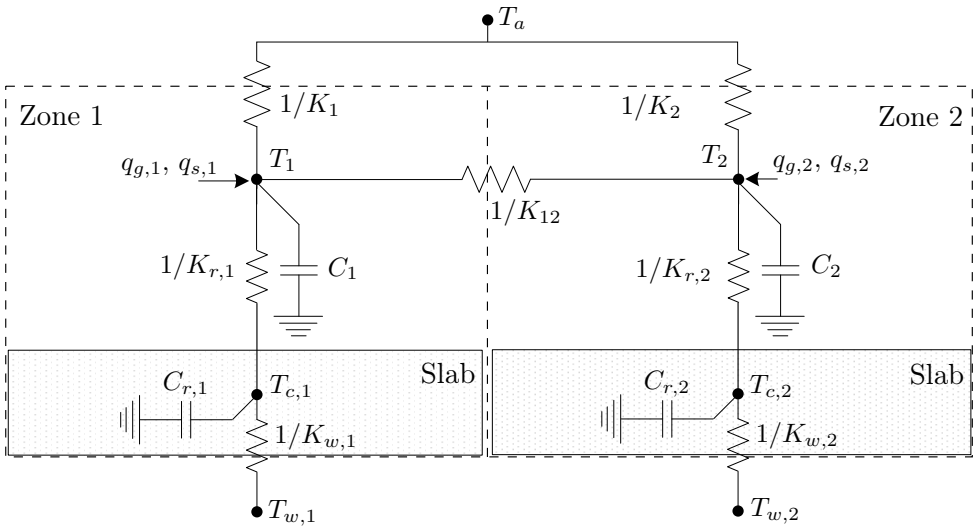


Figure 5.3: RC network model of a hydronic radiant system for two zones.

network model as shown in Figure 5.3 on the current page. The model for each zone i, $i = 1, 2$, has 4 nodes: T_a is the (common) ambient air temperature, T_i is the zone temperature, $T_{c,i}$ is the core temperature of the slab, and $T_{w,i}$ is the supply water temperature. The parameters and variables of the RC network model are summarized in Table 5.1 on the following page.

Given the RC network model, we can write the differential equations for the dynamic thermal model of the zones and their radiant systems. When the pump of zone i is circulating water in the piping system of zone i, the first-order differential equation for the slab's core temperature node $T_{c,i}$ is:

$$C_{r,i}\frac{\mathrm{d}T_{c,i}}{\mathrm{d}t} = K_{r,i}(T_i - T_{c,i}) + K_{w,i}(T_{w,i} - T_{c,i}) = -(K_{r,i} + K_{w,i})T_{c,i} + K_{r,i}T_i + K_{w,i}T_{w,i}. \quad (5.1)$$

When the pump is not running, equivalently the supply water temperature node $T_{w,i}$ is removed, the differential equation for $T_{c,i}$ becomes:

$$C_{r,i}\frac{\mathrm{d}T_{c,i}}{\mathrm{d}t} = -K_{r,i}T_{c,i} + K_{r,i}T_i. \quad (5.2)$$

Table 5.1: List of parameters of the RC network model in Figure 5.3 for a hydronic radiant system.

T_a	ambient air temperature (°C)
T_i	air temperature of zone i, $i = 1, 2$ (°C)
$T_{c,i}$	core temperature of the slab of zone i, $i = 1, 2$ (°C)
$T_{w,i}$	supply water temperature for zone i, $i = 1, 2$ (°C)
$q_{g,i}$	internal heat gain of zone i, $i = 1, 2$ (W/m^2)
$q_{s,i}$	heat gain due to solar radiation of zone i, $i = 1, 2$ (W/m^2)
K_i	thermal conductance between zone i, $i = 1, 2$, and outside air (W/(K m^2))
$K_{r,i}$	thermal conductance between core and air temperatures of zone i, $i = 1, 2$ (W/(K m^2))
$K_{w,i}$	thermal conductance of the piping system of zone i, $i = 1, 2$ (W/(K m^2))
K_{ij}	thermal conductance between zone i and zone j (W/(K m^2))
C_i	thermal capacitance of zone i, $i = 1, 2$ (J/K)
$C_{r,i}$	thermal capacitance of the slab of zone i, $i = 1, 2$ (J/K)

The first-order differential equation for the zone air temperature node T_i can be written as:

$$\begin{aligned} C_i \frac{\mathrm{d}T_i}{\mathrm{d}t} &= K_{r,i}(T_{c,i} - T_i) + K_i(T_a - T_i) + \sum_{j \neq i} K_{ij}(T_j - T_i) + q_{g,i} + q_{s,i} \\ &= -(K_{r,i} + K_i + \sum_{j \neq i} K_{ij})T_i + K_{r,i}T_{c,i} + \sum_{j \neq i} K_{ij}T_j + q_{g,i} + q_{s,i}. \end{aligned} \tag{5.3}$$

The model for each zone i has two state variables T_i and $T_{c,i}$. It also has three disturbance variables T_a, $q_{g,i}$ and $q_{s,i}$. The supply water temperature $T_{w,i}$ can be regulated (controlled) or fixed, depending on the control method for the radiant system.

Let us consider $n > 1$ zones instead of two zones and suppose that $T_{w,i}$ are constant for all i. Define the state vector $x = [T_1, T_{c,1}, \ldots, T_n, T_{c,n}]^T \in \mathbb{R}^{2n}$ and the disturbance vector $d = [q_{g,1}, q_{s,1}, \ldots, q_{g,n}, q_{s,n}, T_a]^T \in \mathbb{R}^{2n+1}$. Let binary variable $u_i \in \{0, 1\}$ denote the running status of the pump of zone i: $u_i = 1$ if the pump is running and $u_i = 0$ otherwise. Define the binary vector $u = [u_1, \ldots, u_n] \in \{0, 1\}^n$. Then the differential equations (5.1) to (5.3) for all zones can be combined to give a state-space model of the entire system:

$$\dot{x}(t) = (A_0 + \sum_{i=1}^{n} A_i u_i(t))x(t) + Bu(t) + Wd(t). \tag{5.4}$$

We note that:

- The state matrix of the model depends on u (i.e., switching state matrix) because the differential equation of the core temperature $T_{c,i}$ changes with respect to the pump's status u_i (Equations (5.1) and (5.2)).
- For every value of u, the state matrix is strictly diagonally dominant with negative diagonal entries, hence it is Hurwitz (Horn and Johnson, 1990).

5.2. Control of Radiant Heating and Cooling Systems

For a hydronic radiant system as in Figure 5.2, the supply water temperature $T_{w,i}$ and the mass flow rate (by the pump) are the two manipulatable variables for low level control. The water temperature can be regulated by using mixing valves, while the mass flow rate can be changed by variable speed control of the circulation pump. In practice, typically one of these two controllable variables is fixed, or only changed infrequently, and the other is manipulated. That is, either the supply water temperature is fixed and variable speed control is used for the pump, or the pump runs at constant speed and the supply water temperature is varied. However, both options require continuous operation of the circulation pump, which can result in high operating cost of the radiant system (i.e., electricity cost of the pump).

5.2.1. Intermittent Operation of Radiant Systems

Intermittent operation of the circulation pump was studied in (Gwerder et al., 2009) for reducing the energy consumption of radiant systems. Essentially, the supply water temperature $T_{w,i}$ is fixed and the circulation pump is either switched off or operated at a constant speed. Because of the high thermal inertia of the radiant systems, this simple control method is appropriate for regulating the zone temperature. Furthermore, as the pump no longer runs continuously, the electricity consumption and electricity cost of the system can be reduced. In the rest of this chapter, we will use this intermittent control method of the circulation pump for the radiant system.

5.2.2. Other Control Methods for Radiant Systems

We briefly review other control methods studied in the literature for radiant systems.

Two-position control is the simplest type of control for radiant systems, in which the system is switched on or off when the zone temperature reaches certain thresholds. Outdoor reset control (ORC) is a feedforward control strategy which sets the supply water temperature accordingly to the ambient air temperature by a predetermined rule (called heating curves). PID control can be used to set the supply water temperature based on the zone temperature error and its history.

More advanced control methods have been proposed to control radiant systems to achieve better performance. Model predictive control was studied in (Lee et al., 2002; Chen, 2002; Sakellariou, 2011) and was shown to improve the comfort and the energy consumption of radiant systems. In Beghi et al. (2011), a controller called *comforstat* was developed to improve the energy performance of a hydronic radiant heating/cooling system by regulating the water temperature. A two-parameter switching control strategy was described in Cho and Zaheer-Uddin (1999) and was shown to achieve better temperature regulation than simple on-off control.

5.3. Green Scheduling for Peak Demand Reduction in Radiant Heating and Cooling Systems

As we mentioned in Section 5.1, one benefit of radiant systems is their capability to flatten out peaks in energy demand. However, intermittent operation of the radiant system causes the electricity demand of the pump to fluctuate when it is switched on and off. Therefore, the peak demand reduction benefit of radiant systems is neutralized for a single zone. Furthermore, in a system of multiple zones, temporally correlated spikes in the energy demand of the system can occur when multiple circulation pumps are activated simultaneously, as we will demonstrate in a case study in Section 5.4. This type of situations is exactly the motivation

of Green Scheduling, about which we discussed back in Section 1.1 in Chapter 1. Therefore, the Green Scheduling strategies we have developed in this dissertation can be applied to coordinate the intermittent operation of the pumps so as to reduce the peak energy demand of a radiant system for multiple zones.

We will present two case studies of Green Scheduling being applied to radiant systems. In the first case study in Section 5.4, Green Scheduling is implemented for a hydronic radiant cooling system similar to that described in Section 5.1.1 and illustrated in Figure 5.2. Simulation results in MATLAB™ will be presented to demonstrate the effective peak demand reduction achieved by Green Scheduling. Section 5.5 investigates a more realistic case study of an electric radiant heating system, whose simulation was performed by the high-fidelity building simulator EnergyPlus. Again, the Green Scheduling approach helped reduce the peak electricity demand of the radiant system.

5.4. Case Study 1: Hydronic Radiant Cooling System

5.4.1. Description of the Building and Radiant System

In this case study, we consider a building which consists of 10 zones of equal size as illustrated in Figure 5.4. Five of the zones face north while the other five face south. A hydronic radiant cooling system is used to satisfy the cooling need of the building. The radiant system has the same configuration as described in Section 5.1.1 and illustrated in Figure 5.2. That is each zone is served by a separate piping circuit with its own circulation pump. The supply water temperature is the same for all zones and is fixed at a predetermined value T_w. Each pump is operated intermittently, i.e., it can only be either switched off, when the mass flow rate is 0, or switched on, when the mass flow rate is constant.

For each zone, we used the parameter values from (Gwerder et al., 2008), which are summarized in Table 5.2 on page 151. These are the characteristics of a zone in a typical office building. Because the parameters in (Gwerder et al., 2008) are for a single zone, while the

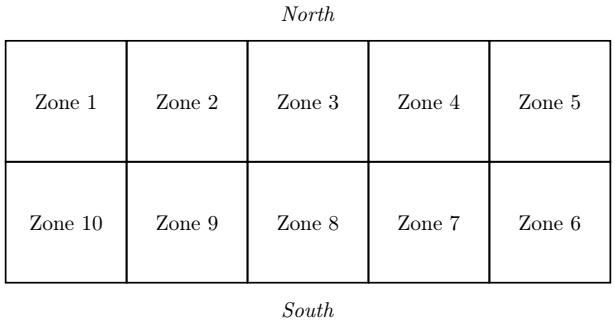


Figure 5.4: Layout of the building considered in the case study in Section 5.4. There are 10 zones: five of them face north, the other five face south.

considered building has 10 zones, we varied several parameter values for each zone randomly around the nominal values given in the reference. The range for each of these parameters is also reported in Table 5.2. For the four corner zones that have two external walls (zones 1, 5, 6 and 10), we increased the thermal conductance K_i accordingly. For thermal comfort of the occupants, the zone temperatures should stay in the desired range from 22 °C to 26 °C.

5.4.2. Disturbances

As we presented in Section 5.1.1, there are three disturbance variables affecting each zone, namely the ambient air temperature T_a, the internal heat gain $q_{g,i}$ (e.g., from occupants, lights and equipment) and the heat gain $q_{s,i}$ due to solar radiation into the zone. These disturbances are constrained and their time-varying constraint sets are assumed to be known. In particular, weather forecast and historical data can give us the range of T_a and the range of $q_{s,i}$ for each zone i at any given time during the day. The constraint of $q_{g,i}$ for each zone i can be calculated based on the power rates of the equipment and lights in the zone as well as its occupants' schedules. In this case study, we assume the time-varying constraints of T_a, $q_{g,i}$ and $q_{s,i}$, for every $i \in \{1, \ldots, 10\}$, as given in Figure 5.5 on page 152. Note that between the north zones (zones 1 to 5) and the south zones (zones 6 to 10), their solar radiation heat gain profiles are different, as clearly displayed in Figure 5.5c. In the simulations that follow, each disturbance signal was generated randomly within its given time-varying constraint.

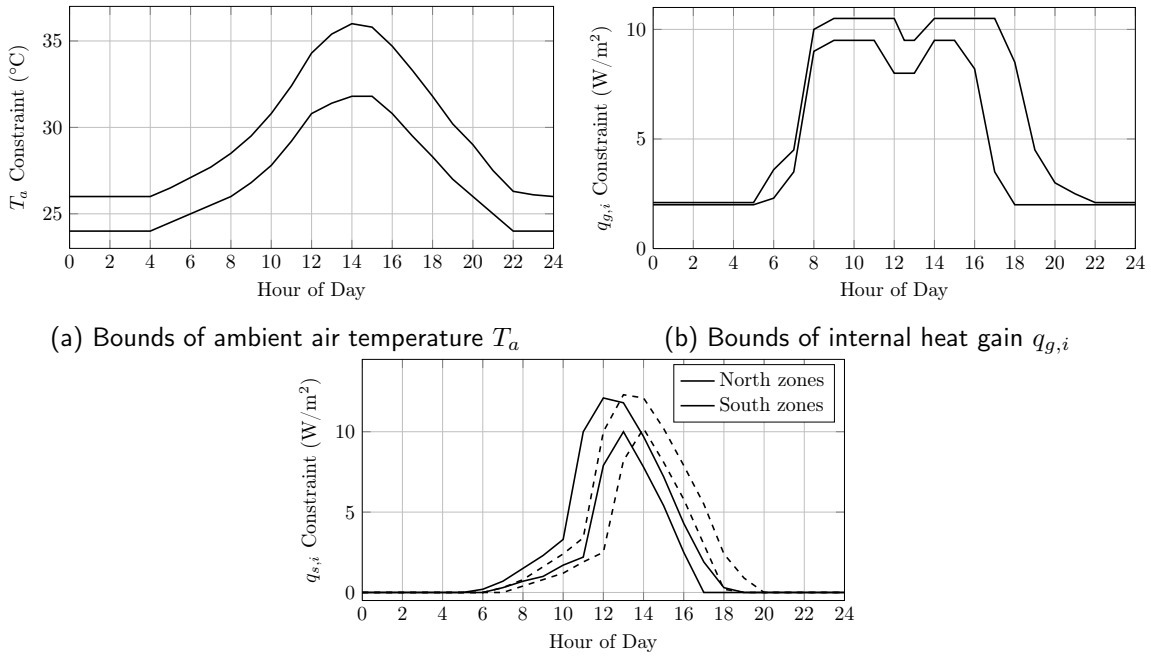
Table 5.2: Parameter values for a zone in the case study in Section 5.4.

Space length, width, height	$6\,\mathrm{m} \times 6\,\mathrm{m} \times 3\,\mathrm{m}$
Façade area	$18\,\mathrm{m}^2$
Internal-wall area	$36\,\mathrm{m}^2$
Thickness of concrete slab	$250\,\mathrm{mm}$
Pipe spacing	$200\,\mathrm{mm}$
External/internal pipe diameter	$20/15\,\mathrm{mm}$
Mass flow rate (per slab area)	$15\,\mathrm{kg}/(\mathrm{h}\,\mathrm{m}^2)$
$T_{w,i}$	$18\,^\circ\mathrm{C}$
$1/K_i$	[2.1, 2.2] $(\mathrm{K}\,\mathrm{m}^2/\mathrm{W})$
$1/K_{r,i}$	[0.124, 0.130] $(\mathrm{K}\,\mathrm{m}^2/\mathrm{W})$
$1/K_{w,i}$	[0.05, 0.07] $(\mathrm{K}\,\mathrm{m}^2/\mathrm{W})$
$1/K_{ij}$ (only for adjacent zones)	[0.16, 0.20] $(\mathrm{K}\,\mathrm{m}^2/\mathrm{W})$
C_i	[1900, 2100] (kJ/K)
$C_{r,i}$	[3000, 4000] (kJ/K)
Desired zone temperature range	[22, 26] (°C)

5.4.3. Uncoordinated Intermittent Operation

As the baseline control of the radiant system, we considered the uncoordinated intermittent operation of the pumps. For each zone i, its pump is switched on and off independently of the other zones and with hysteresis so that its zone temperature is maintained in the desired range $[22, 26]^\circ\mathrm{C}$. Whenever the zone temperature T_i is above $26\,^\circ\mathrm{C}$, its pump is switched on so that supply water flows through its piping circuit to cool the slab, hence the zone's air. Whenever the zone temperature T_i is below $22\,^\circ\mathrm{C}$, its pump is switched off. This simple control strategy was simulated in MATLAB™ for 24 hours (1 day). The initial temperatures for all zones were set to $27\,^\circ\mathrm{C}$. The resulted air temperature T_1 and core temperature $T_{c,1}$ of zone 1 are plotted in Figure 5.6 on page 153.

As obviously seen in Figure 5.6, an issue with the simulation result is that the zone temperature T_1 dropped below the lower threshold $22\,^\circ\mathrm{C}$, to as low as about $21\,^\circ\mathrm{C}$, twice during the day. For example, between point A and point B in the figure, T_1 was below $22\,^\circ\mathrm{C}$ for more than 5 hours. This phenomenon can be explained by the high thermal inertia of the radiant



(a) Bounds of ambient air temperature T_a

(b) Bounds of internal heat gain $q_{g,i}$

(c) Bounds of solar radiation heat gains $q_{s,i}$ are different between the north zones and the south zones

Figure 5.5: Time-varying constraints of the disturbances affecting each zone in the case study in Section 5.4.

system. When the circulation pump was switched off (point A), the slab stopped being cooled; however the core temperature was still significantly below the zone temperature, causing the air to continue being cooled. Therefore the zone temperature dropped below the lower threshold and only started rising up after several hours, when the zone and core temperatures were sufficiently close.

To avoid the above problem, we increased the lower threshold for control, at which the pump will be switched off, by 1 °C to 23 °C. The new simulation results are reported in Figure 5.7 on the following page. Clearly the zone temperature T_1 was driven to and maintained within the desired range (the gray-filled area in the figure), hence it was safe.

We now report the peak power demand and the total energy consumption of this control strategy. Because we did not have the power rating information of the pumps used in (Gwerder et al., 2008), we assumed a normalized power demand of 1 power unit for each

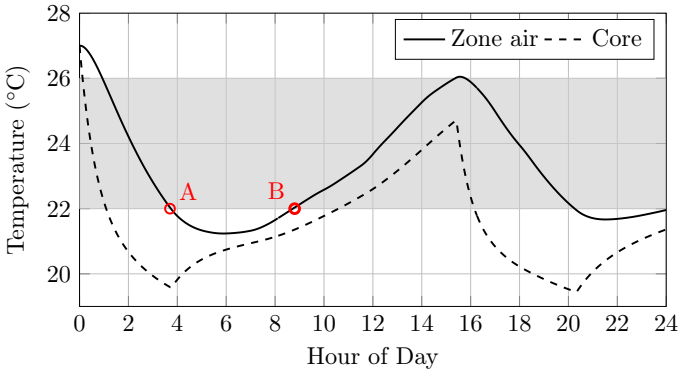
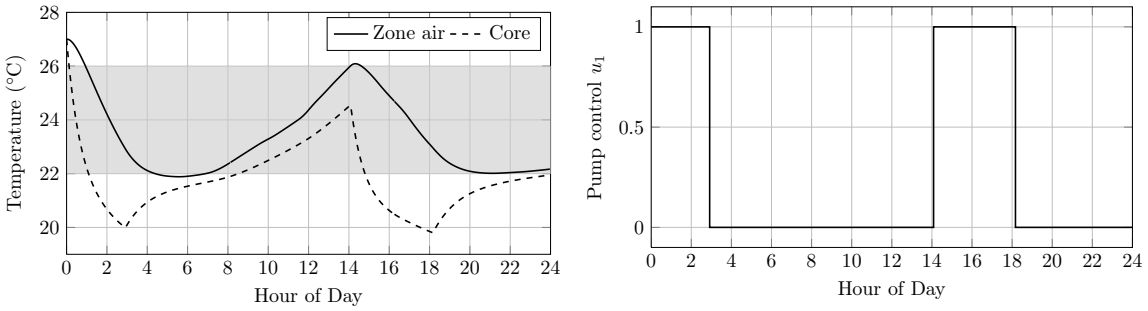


Figure 5.6: Air and core temperatures of zone 1 for the simulation of the uncoordinated intermittent operation in the case study in Section 5.4. The safety requirement was not satisfied as the zone temperature T_1 dropped below 22 °C between points A and B.



(a) Zone and core temperatures of zone 1

(b) Pump operation of zone 1

Figure 5.7: Simulation results for the safe uncoordinated intermittent operation in the case study in Section 5.4. The zone temperature T_1 was driven to and maintained within the desired range (gray-filled area), hence it was safe.

pump. The pumps were also assumed to be identical. The actual power results can be obtained by scaling the normalized results by the actual power rate of the pumps. Figure 5.8 on the next page shows the normalized total power demand of all the pumps. Evidently, the demand fluctuated significantly during the day. It attained a very high peak of 9 during the on-peak hours between around 3:30 PM and around 5:45 PM. Thus, under a demand-based electricity pricing policy (cf. Section 1.1), it would incur a high demand cost. The total energy consumption of the pumps was 62.5 (power unit × hour).

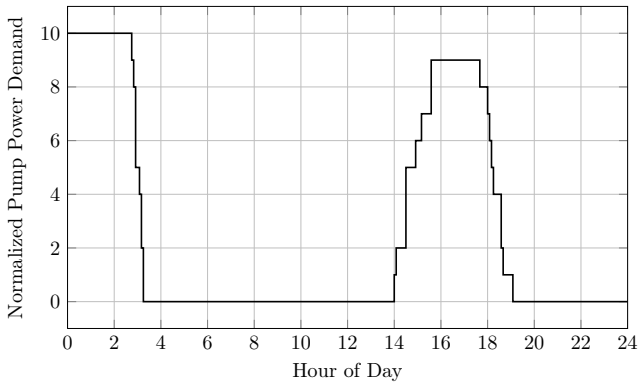


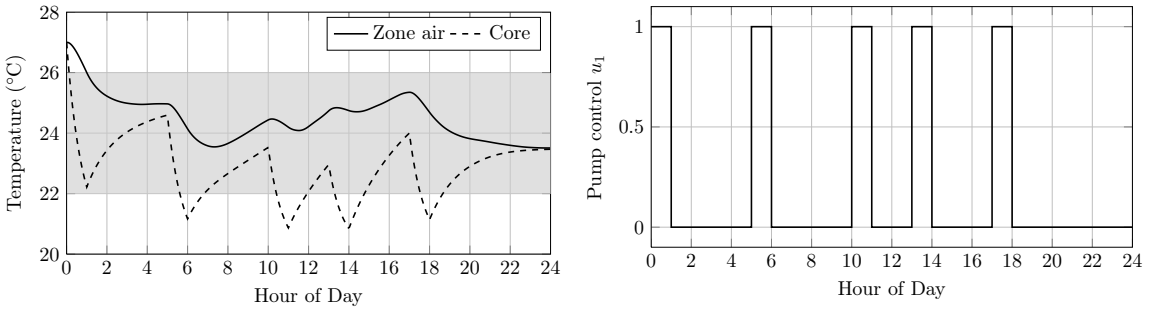
Figure 5.8: Total power demand of the pumps for the simulation of the uncoordinated intermittent operation in the case study in Section 5.4. The demand fluctuated significantly and attained a high peak of 9 during the on-peak hours.

5.4.4. Green Scheduling Implementation

To flatten out the high peak demand incurred by the uncoordinated intermittent operation of pumps, we applied the Green Scheduling approach developed in the previous chapters. A state-space model of the radiant system and the zones can be obtained as presented in Section 5.1.1. The model has 20 state variables, 10 binary control inputs (the pumps' operation status), and 21 disturbance variables (internal and solar heat gains, and ambient air temperature). In addition, the state matrix of the model depends affinely on the pumps' control inputs u. Therefore the model is a switched affine system (cf. Equation (1.10)).

Because the disturbances were significant and their constraints were time-varying, we did not use the periodic scheduling approach but the feedback scheduling approach based on attracting sets (Section 4.3). Furthermore, we assumed that at any time, one-hour predictions of the disturbances' constraints were available to the scheduler. Therefore, we applied the improved feedback scheduling strategy with disturbance prediction, presented in Section 4.3.6.

Following the schedulability analysis in Section 2.5, the peak constraint on the control inputs u was calculated to be $k = 2$ at all time. The self-triggered scheduling algorithm (Algorithm 4.4 on page 117) was then implemented for the radiant system. A MATLAB™ simulation was performed for 24 hours with the same disturbance profiles and the same initial temperatures



(a) Zone and core temperatures of zone 1

(b) Pump operation of zone 1

Figure 5.9: Simulation results for the self-triggered Green Scheduling algorithm with one-hour disturbance predictions in the case study in Section 5.4. The zone temperature T_1 was driven to and maintained within the desired range (gray-filled area), hence it was safe.

as in Section 5.4.3. We note that in each iteration of the self-triggered scheduling algorithm, the delay time was always computed to be over 1 hour but was shortened to 1 hour due to the one-hour-horizon of the disturbance predictions.

Figure 5.9 on the current page plots the simulation results for Green Scheduling. It is clear that the temperature of zone 1 (Figure 5.9a) was safe as it was driven to and maintained inside the desired range (the gray-filled area in the figure). Compared to the uncoordinated scheduling, the pump of zone 1 was switched on and off twice as often, however its switching frequency was still slow (less than once every 3 hours).

The total power demand of the pumps is plotted in Figure 5.10 on the following page in comparison with that for the uncoordinated intermittent operation in Section 5.4.3. Evidently, the peak demand incurred by Green Scheduling was flattened out and was significantly smaller than that of uncoordinated scheduling. In fact, the demand of Green Scheduling was constant at 2 for most of the day, until 9 PM after which it was reduced to 1 and then 0. The peak demand and total energy consumption of both scheduling strategies are compared in Table 5.3 on the next page. By applying Green Scheduling to the radiant system, we saved 80% in peak demand and 31.2% in total energy consumption. Under a demand-based electricity tariff, this would amount to a large saving in electricity cost.

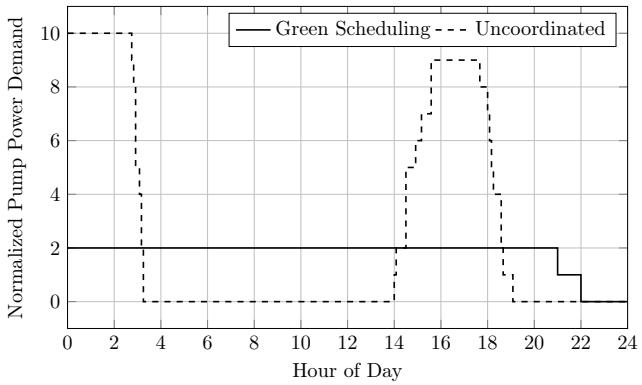


Figure 5.10: Total power demand of the pumps for the simulation of the Green Scheduling strategy in the case study in Section 5.4. The Green Scheduling's demand curve was flattened out and was constant for most of the day. The peak demand was reduced significantly compared to the uncoordinated scheduling strategy, from 10 down to 2.

Table 5.3: Comparison of the peak demand and total energy consumption of the Green Scheduling strategy and the uncoordinated scheduling strategy. Green Scheduling helped reduce the peak demand by 80% and the total energy consumption by 31.2%.

	Normalized peak demand	Normalized energy consumption
Uncoordinated control	10	62.5
Green Scheduling	**2 (-80%)**	**43 (-31.2%)**

5.5. Case Study 2: Electric Radiant Heating System in EnergyPlus

Much of the content of this section is adapted from our previous work (Nghiem et al., 2012b).

5.5.1. Description of the Building and Radiant System

In this case study, we considered a single floor, L-shaped building divided into 3 interior conditioned zones as shown in Figure 5.11 on the following page. There is a single window in the West zone South wall. An electric low temperature radiant system is used for heating the floor of each zone, with power ratings of 12kW, 8kW and 8kW for the North, West and East zones respectively. The floor of each zone consists of a slab of high thermal capacity below which the radiant heat source is embedded. The actuation is on-off, i.e., the radiant system in each zone can be either switched on, when it provides its maximal heating power rate, or

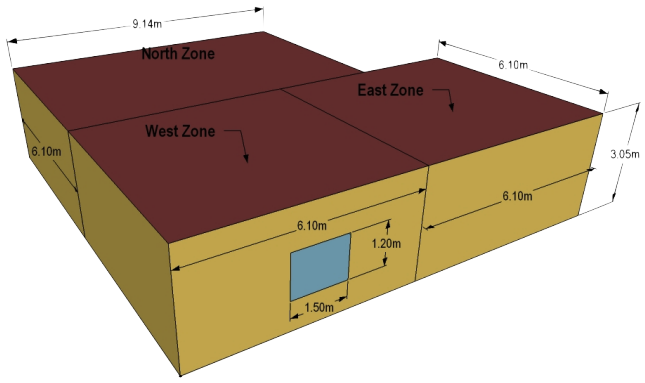


Figure 5.11: 3-D building model for the case study in Section 5.5.

switched off, when it provides no heating power. The temperature of each zone was desired to be between 22 °C and 24 °C. The ambient air temperature profile was of Chicago, IL, USA. The disturbances due to internal heat gain and solar radiation heat gain were different for every zone and time-varying.

An EnergyPlus model of the building was modified from an example distributed with EnergyPlus version 7.0. EnergyPlus is a standard energy analysis and thermal load simulation program, developed by the U.S. Department of Energy. It is designed for modelling building heating, cooling, lighting, ventilating, and other energy flows. EnergyPlus uses complex and detailed models for realistic building simulations. Therefore it can achieve high-fidelity simulations of building energy systems.

In this case study, we used the EnergyPlus model as the *ground truth* for the building, i.e., it was considered as the "real" building. System identification of the building model and implementation of controllers for the building's radiant heating system were carried out in MATLAB™, while thermal simulation of the building was performed in EnergyPlus. We used an in-house developed tool called MLE+ (Nghiem, 2011) to interface MATLAB™ with EnergyPlus for co-simulation.

5.5.2. Model Identification

Since the internal thermal model of the EnergyPlus model is not accessible from outside EnergyPlus, our first step was to identify a linear model for the building. The disturbances were considered to be the ambient air temperature, the internal heat gains for each zone, and the solar radiation for the West zone (the only zone with a window). Instead of identifying the physical parameters of the state-space model described in Section 5.1.1, we used the black-box system identification approach. The model obtained by this approach does not base on first principles or physical laws of the thermal process, therefore its parameters might not have a physical interpretation.

In particular, randomly generated binary control signals were used to excite the EnergyPlus model of the building via MLE+ and the resulted zone temperatures were recorded and imported into MATLAB™ as time series. Several such controlled experiments on the EnergyPlus building model were carried out for 5 days in January. The *System Identification Toolbox* (The MathWorks, 2009) of MATLAB™ was used to estimate a building model from the experiment data.

The identified model was then validated on January 14, which was not one of the experiment days. Figure 5.12 on the next page plots the measured (in EnergyPlus) and simulated (in MATLAB™) mean air temperatures of the zones on the validation day. The simulated outputs fit the measured outputs with accuracies of 84.24%, 76.89% and 84.26% for the West, East, and North zones respectively.

5.5.3. Green Scheduling Implementation

We applied the Green Scheduling approach developed in previous chapters to the case study on the validation day. The heating schedule of the EnergyPlus model specifies that the heating system is switched off during the night from 6 PM to 6 AM, then is switched on to pre-heat the building from 6 AM to 8 AM, and is in normal operation mode from 8 AM to 6 PM (the working hours of the building). Therefore, we used green scheduling for controlling

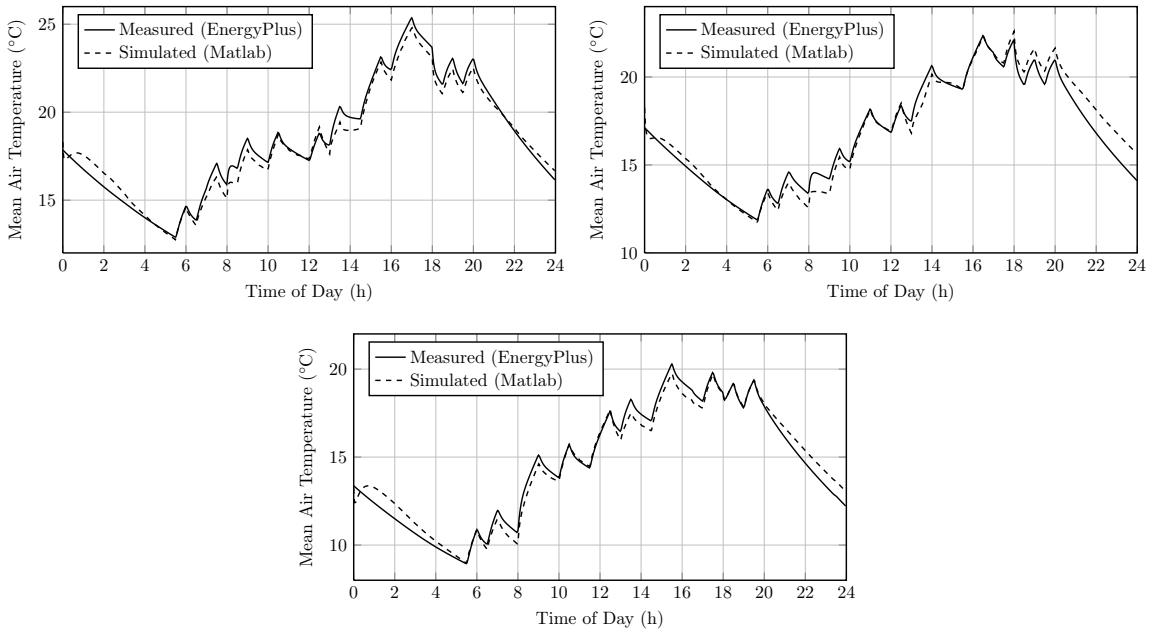


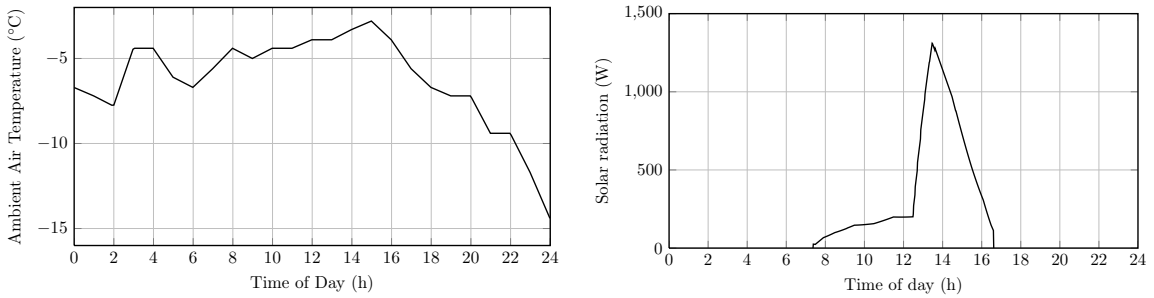
Figure 5.12: Validation of the identified model for the case study in Section 5.5. Plotted are the measured (in EnergyPlus) and simulated (in MATLAB™) mean air temperatures of the West zone (top left), the East zone (top right) and the North zone (bottom).

the radiant heating systems of the building from 8 AM to 6 PM.

Similarly to the previous case study, we first tried the feedback scheduling approach based on attracting sets. However, we failed to derive a safe scheduling strategy for the system using this approach. This failure could be attributed to the identified black-box model, on which the scheduling algorithm heavily relies. Firstly, the accuracy of the identified model was inadequate for the scheduling algorithm. Secondly, the scheduling algorithm uses state feedback but, in our experiments, the state variables of the black-box model were not accurately estimated from the mean air temperatures of the zones by a Kalman filter[2]. For this reason, we used instead the periodic Green Scheduling approach for this case study.

As the periodic Green Scheduling approach developed in Chapter 3 does not directly handle time-varying disturbances, we used nominal values of the disturbances to derive periodic schedules. In this case study, disturbance prediction was used to derive these nominal values.

[2]Based on the identified model, we designed a Kalman filter in MATLAB™ to estimate the state of the system from its outputs.



(a) Ambient air temperature profile

(b) Predicted solar radiation gain to the West zone

Figure 5.13: Disturbance profiles for the case study in Section 5.5.

According to the weather profile (Figure 5.13a on this page), the ambient air temperature varied around −5 °C between 8 AM and 6 PM, thus we used −5 °C as the nominal value for T_a. Based on the occupancy and equipment schedules, the internal heat gain of each zone can be predicted, and we chose its nominal value to be 600 W, 700 W, and 800 W for the West, East, and North zones respectively. For the predicted solar radiation gain to the West zone (the only zone with a window), we noticed a significant increase at around 1 PM due to the window's direction, from under 200 W to over 1000 W (Figure 5.13b on the current page). Therefore, we chose two different nominal solar radiation gains: 100 W before 1 PM and 600 W after 1 PM (both were averaged values for the respective intervals).

On inspecting the predicted disturbances, we decided to synthesize two periodic schedules: one to be used before 1 PM and one after 1 PM. Their parameters are reported in Table 5.4 on the following page. Notice that the time period δ were both 60 minutes, which were reasonably large. Total computation time was less than 1 s for each case. Because the computation is fast, instead of using disturbance prediction, we could monitor the environment (e.g., ambient air temperature and occupancy) and regenerate the schedule on the fly whenever there is a significant change in the disturbances.

5.5.4. Simulation Results

The periodic schedules were implemented in MATLAB™ and interfaced with the building energy simulation in EnergyPlus via MLE+. For comparison, we also implemented the

Table 5.4: Two periodic schedules for the case study in Section 5.5.

Schedule	k	η	δ (min)	Nominal disturbances
Before 1 PM	2	$[0.35, 0.42, 0.45]$	60	$\overline{T}_a = -5\,°\text{C}$, $\bar{q}_{g,\text{west}} = 600\,\text{W}$, $\bar{q}_{g,\text{east}} = 700\,\text{W}$, $\bar{q}_{g,\text{north}} = 800\,\text{W}$, $\bar{q}_{s,\text{west}} = 100\,\text{W}$
After 1 PM	1	$[0.05, 0.19, 0.31]$	60	$\overline{T}_a = -5\,°\text{C}$, $\bar{q}_{g,\text{west}} = 600\,\text{W}$, $\bar{q}_{g,\text{east}} = 700\,\text{W}$, $\bar{q}_{g,\text{north}} = 800\,\text{W}$, $\bar{q}_{s,\text{west}} = 600\,\text{W}$

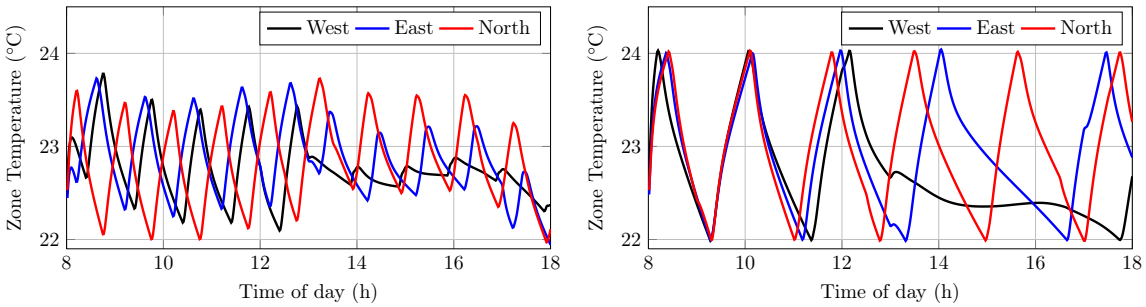


Figure 5.14: Simulation results of the case study in Section 5.5: zone temperatures for periodic green scheduling (left) and for uncoordinated on-off scheduling (right).

uncoordinated on-off scheduling strategy, where the radiant system in each zone was controlled by an individual two-position thermostat. The thermostats worked independently of each other. Zone temperatures from the simulations are plotted in Figure 5.14 on the current page. In both cases, zone temperatures were kept in the desired range between 22 °C and 24 °C. We observed that the curve of electricity demand for the uncoordinated scheduling strategy had several high spikes while that for the green scheduling strategy was flatten out (Figure 5.15 on the following page). For green scheduling, the effect of switching from the first schedule (peak constraint $k = 2$) to the second schedule (peak constraint $k = 1$) at 1 PM can be seen clearly in Figure 5.14 and Figure 5.15. In total, green scheduling helped save 8% in electricity consumption and reduce peak demand by 42.9% (Table 5.5 on the next page). There was a decrease in the total energy consumption since the periodic schedule tended to operate at a lower mean temperature than uncoordinated scheduling, as can be observed in Figure 5.14.

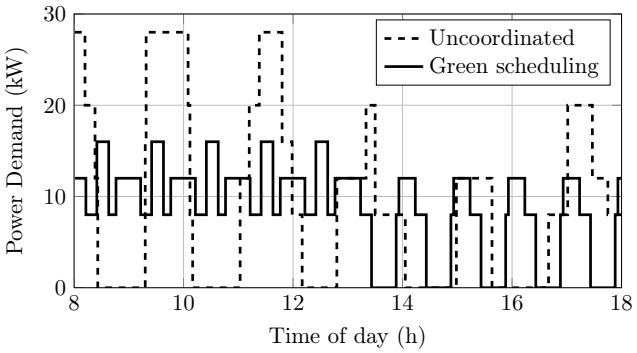


Figure 5.15: Electricity demands of the periodic green scheduling and the uncoordinated on-off scheduling for the case study in Section 5.5.

Table 5.5: Peak demand and total energy consumption of the periodic green scheduling and the uncoordinated on-off scheduling for the case study in Section 5.5.

	Uncoordinated	Green scheduling (% saved)
Consumption (kW h)	93.2	85.7 (8.0%)
Peak demand (kW)	28.0	16.0 (42.9%)

5.6. Conclusions

In this chapter, we applied the theory of Green Scheduling developed in the earlier chapters to the intermittent operation of radiant heating and cooling systems in buildings. As the pumps (for hydronic radiant systems) or the electric resistance wires (for electric radiant systems) are only switched on and off, a building of multiple such systems can incur high peaks in its energy demand if the systems are operated independently of each other. The Green Scheduling approach were shown to be effective to reduce the total peak demand of the radiant systems while ensuring thermal comfort in the zones is always maintained.

We considered two case studies. In the first case study of a hydronic radiant cooling system modeled in MATLAB™, the self-triggered feedback scheduling approach based on attracting sets (Section 4.3) was used to derive a safe scheduling strategy for the system. The peak demand and the total energy consumption were reduced significantly. However, this feedback scheduling algorithm relies on an accurate model of the system, both for computing the control actuation and for estimating the system's state (if the state cannot be measured

directly). Failure to satisfy this requirement causes failure to apply the feedback scheduling algorithm, as we showed in the second case study. In contrast, while the periodic Green Scheduling approach is not as flexible as the feedback scheduling approach, it does not require an accurate model of the system. Periodic scheduling was implemented successfully for the second case study to reduce peaks in the electricity demand of the system. In addition, the light computation of the periodic scheduling algorithm allows it to be adjusted quickly on the fly when the environment changes significantly.

Chapter 6

Conclusions

6.1. Summary of Contributions

We proposed an approach to the peak demand reduction problem by scheduling multiple interacting control systems within a constrained peak demand envelope while ensuring that safety and operational conditions are facilitated. The peak demand envelope was formulated as a constraint on the number of binary control inputs that can be activated simultaneously. We called this approach *Green Scheduling*.

Using two different approaches, namely periodic scheduling and feedback scheduling based on attracting sets, we established a spectrum of sufficient and necessary schedulability conditions for various classes of affine dynamical systems in Chapter 2. Depending on whether the systems are decoupled or coupled and whether disturbances are present, an appropriate method can be selected. We showed by numerical examples that these schedulability analysis methods were scalable for large-scale systems with 1000 control inputs or even more.

Once the schedulability of the Green Scheduling system has been established, it is desirable to synthesize scheduling algorithms that can safely control the system within the given peak demand envelope. To this end, we first developed in Chapter 3 a periodic scheduling synthesis method for Green Scheduling systems without disturbances. Through numerical simulations, the method was shown to be effective and scalable for large-scale systems. However, it does not directly take into account the influence of disturbances on the system. We then improved the periodic scheduling algorithm to make it robust to disturbances while preserving the simplicity and scalability of periodic scheduling. This improvement was discussed in Section 4.2, together with the development of event-triggered and self-triggered algorithms for reducing the frequent switching of control inputs. Taking another approach

based on attracting sets and robust control Lyapunov functions, we developed in Section 4.3 event-triggered and self-triggered scheduling algorithms that could handle large disturbances affecting the Green Scheduling system. Through simulations, these new algorithms were demonstrated to be scalable and flexible as they could exploit prediction of the disturbances to improve their performance. Finally, backward reachability analysis was used in Section 4.4 to develop a scheduling method for discrete-time Green Scheduling systems. This discrete-time scheduling algorithm can handle more sophisticated system's dynamics and safety specifications. However, it also has very high computational requirements and is thus only applicable to small-scale systems.

In the last chapter, Chapter 5, we applied the results developed in previous chapters to scheduling of radiant heating and cooling systems for peak demand reduction. Through two case studies, one in MATLAB™ and one in co-simulation with EnergyPlus, we demonstrated the effectiveness of the Green Scheduling approach in reducing the peak demand and the total energy consumption of the radiant systems while maintaining thermal comfort for occupants. These case studies also showed the advantages and limitations of the established scheduling methods. While the state feedback scheduling algorithms are more robust and more flexible, they require an accurate model of the system. On the other hand, the periodic scheduling algorithm is applicable even when we do not have a very good system's model, but it is less robust to disturbances and less flexible.

6.2. Future Work

In this work, we have established the foundational theory for Green Scheduling: the schedulability analysis and several centralized Green Scheduling algorithms. Although several developed scheduling algorithms are scalable, there are limitations to how large a system a centralized algorithm can handle. Moreover, in many practical large-scale systems, the subsystems are distributed either physically or logically, or both. Therefore, an important future extension of Green Scheduling is the development of distributed and hierarchical scheduling algorithms. Extension of the theoretical results for Green Scheduling to other

classes of systems, e.g., multi-mode systems and nonlinear systems, will be addressed.

While we developed the theoretical results in this dissertation, the main focus was on the peak constraint and the safety specifications. We have largely ignored the performance aspect of the system, for example the optimality with respect to some cost function. This cost function can represent the actual cost of operating the system. For future work, we will investigate the performance assessment of Green Scheduling systems and Green Scheduling algorithms. We will also explore other approaches to the Green Scheduling problem, for instance using Game Theory and Mechanism Design, or a multi-agent-based approach.

Appendix A

Proofs

A.1. Proofs of Chapter 2

A.1.1. Proof of Theorem 2.1

PROOF The proof is by contradiction. Assume that the system is schedulable. Let $x(0) \in \mathcal{X}_0$ be any initial state. Then, by Definition 2.1, there exists a schedule $u(\cdot)$ or a scheduling strategy $\kappa(x)$ and a finite time $\tau \geq 0$ such that $x(t) \in \texttt{Safe}\ \forall t \geq \tau$. By condition (2.3a) we have that

$$g(x(t)) \geq 0 \quad \forall t \geq \tau. \tag{A.1}$$

By condition (2.3b), at any time $t \geq \tau$, there exists an admissible disturbance input $d(t) \in \mathcal{D}$ such that $\frac{\mathrm{d}}{\mathrm{d}t} g(x(t)) = \nabla g(x(t)) \cdot f(x(t), u(t), d(t)) \leq -\epsilon$. Therefore, there exists an admissible disturbance signal $d(\cdot)$ such that along the flow of the system[1], $g(x(t))$ always decays at a rate at least $-\epsilon$ after time instant τ. Since g is differentiable, it is continuous. Furthermore, `Safe` is compact, thus g is bounded on `Safe`, that is there exists a finite number M such that $|g(x)| \leq M\ \forall x \in \texttt{Safe}$. It follows that for all $t > \frac{M}{\epsilon} + \tau$, $g(x(t)) \leq g(x(\tau)) - \epsilon\,(t - \tau) < 0$, which contradicts inequality A.1. Therefore the system must be non-schedulable. ∎

A.1.2. Proof of Lemma 2.1

PROOF For the sake of clarity, we drop the subscript i in this proof; all variables and parameters herein implicitly refer to those with the subscript i. First, note that $0 < \underline{\eta} < \bar{\eta} < 1$ due to Equation (2.6) on page 25, thus η satisfying $\underline{\eta} < \eta < \bar{\eta}$ always exists. Define $y(j) := x(j\delta)$ and $z(j) := x((j+\eta)\delta)$ for $j \in \mathbb{N}$, corresponding to the state values at the time

[1] For a more detailed proof, see (Girard and Pappas, 2007a, Theorem 2) or the proof of Theorem 2.9 in Appendix A.1.9 on page 182.

instants when u switches either from 0 to 1 or from 1 to 0. It follows from the monotonicity of the dynamics in Assumption 2.1 that $x(t)$ is bounded between the sequences $\{y(j)\}_{j\in\mathbb{N}}$ and $\{z(j)\}_{j\in\mathbb{N}}$. We will show that these sequences are convergent and their limits depend on the values of η and δ.

From the dynamics in Equation (2.4), the sequence $\{y(j)\}_{j\in\mathbb{N}}$ is given by

$$y(0) = x(0) \text{ and } y(j+1) = A(\delta)y(j) + B_y(\delta), \quad \forall j \in \mathbb{N} \tag{A.2}$$

where

$$A(\delta) = \mathrm{e}^{-\delta(a_{\text{off}}(1-\eta)+a_{\text{on}}\eta)}$$
$$B_y(\delta) = \mathrm{e}^{-a_{\text{off}}(1-\eta)\delta}\left(\frac{b_{\text{on}}}{a_{\text{on}}}\left(1-\mathrm{e}^{-a_{\text{on}}\eta\delta}\right) - \frac{b_{\text{off}}}{a_{\text{off}}}\right) + \frac{b_{\text{off}}}{a_{\text{off}}}$$

in which A and B_y depend only on δ since η is given. Observe that Equation (A.2) characterizes a discrete-time linear system. Because $\delta > 0$, $a_{\text{off}} > 0$, $a_{\text{on}} > 0$ and $0 < \eta < 1$, we have $0 < A(\delta) < 1$. From linear system theory (Rugh, 1996), the sequence $\{y(j)\}_{j\in\mathbb{N}}$ is monotonic and asymptotically converges to $\alpha(\delta) = B_y(\delta)/(1 - A(\delta))$. Similarly, the sequence $\{z(j)\}_{j\in\mathbb{N}}$ is monotonic and asymptotically converges to $\beta(\delta) = B_z(\delta)/(1 - A(\delta))$ where

$$B_z(\delta) = \mathrm{e}^{-a_{\text{on}}\eta\delta}\left(\frac{b_{\text{off}}}{a_{\text{off}}}\left(1-\mathrm{e}^{-a_{\text{off}}(1-\eta)\delta}\right) - \frac{b_{\text{on}}}{a_{\text{on}}}\right) + \frac{b_{\text{on}}}{a_{\text{on}}}.$$

By simple calculations, we can verify that

$$B_z(\delta) - B_y(\delta) = \left(\frac{b_{\text{on}}}{a_{\text{on}}} - \frac{b_{\text{off}}}{a_{\text{off}}}\right)\left(1-\mathrm{e}^{-a_{\text{off}}(1-\eta)\delta}\right)\left(1-\mathrm{e}^{-a_{\text{on}}\eta\delta}\right) > 0$$

in which the inequality (2.6) is used. Hence, $\beta(\delta) > \alpha(\delta)$. Furthermore, in the limit, $x(t)$ is bounded in the interval $[\alpha(\delta), \beta(\delta)]$ (cf. Figure 2.3 on page 27). Therefore, we only need to find δ so that $[\alpha(\delta), \beta(\delta)] \subset [l, h]$, i.e., $\alpha(\delta) > l$ and $\beta(\delta) < h$.

The limit $\alpha(\delta)$: Function $\alpha(\delta)$ on the domain $\{\delta > 0\}$ is continuous and, by L'Hôpital's rule, has

$$\lim_{\delta\to 0^+} \alpha(\delta) = \lim_{\delta\to 0^+} \frac{B_y(\delta)}{1 - A(\delta)} = \lim_{\delta\to 0^+} \frac{B_y(\delta)}{(1 - A(\delta))'} = \frac{b_{\text{on}}\eta + b_{\text{off}}(1-\eta)}{a_{\text{on}}\eta + a_{\text{off}}(1-\eta)}$$

From $\eta > \underline{\eta}$ and the definition of $\underline{\eta}$ (Equation (2.8) on page 28), we have

$$\begin{aligned} \frac{a_{\text{off}}l - b_{\text{off}}}{(a_{\text{off}}l - b_{\text{off}}) - (a_{\text{on}}l - b_{\text{on}})} = \underline{\eta} < \eta \quad &\Leftrightarrow \quad a_{\text{off}}l - b_{\text{off}} < ((a_{\text{off}}l - b_{\text{off}}) - (a_{\text{on}}l - b_{\text{on}}))\,\eta \\ &\Leftrightarrow \quad (a_{\text{on}}\eta + a_{\text{off}}(1-\eta))\,l < b_{\text{on}}\eta + b_{\text{off}}(1-\eta) \\ &\Leftrightarrow \quad l < \frac{b_{\text{on}}\eta + b_{\text{off}}(1-\eta)}{a_{\text{on}}\eta + a_{\text{off}}(1-\eta)}. \end{aligned}$$

Thus $\lim_{\delta\to 0^+} \alpha(\delta) > l$. Therefore, there exists $\delta_y > 0$ such that $0 < \delta < \delta_y$ implies $\alpha(\delta) > l$.

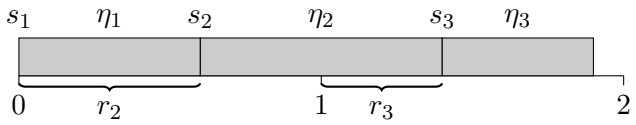
The limit $\beta(\delta)$: Similarly, there exists $\delta_z > 0$ such that $0 < \delta < \delta_z$ implies $\beta(\delta) < h$.

Let $\delta^\star = \min\{\delta_y, \delta_z\} > 0$. We will prove that with any $0 < \delta < \delta^\star$, there exists a finite $\tau \geq 0$ such that $x(t) \in [l, h]\ \forall t \geq \tau$. Evidently, $l < \alpha(\delta) < \beta(\delta) < h$. Select any $\epsilon > 0$ satisfying $\epsilon \leq \min\{\alpha(\delta) - l, h - \beta(\delta)\}$. Then the convergence of $\{y(j)\}_{j\in\mathbb{N}}$ and $\{z(j)\}_{j\in\mathbb{N}}$ guarantees that there exists a finite time $\tau > 0$ such that for all $j \geq \frac{\tau}{\delta}$, $y(j)$ and $z(j)$ are in the interval $[\alpha(\delta) - \epsilon, \beta(\delta) + \epsilon] \subseteq [l, h]$. This implies that $x(t) \in [l, h]\ \forall t \geq \tau$. ∎

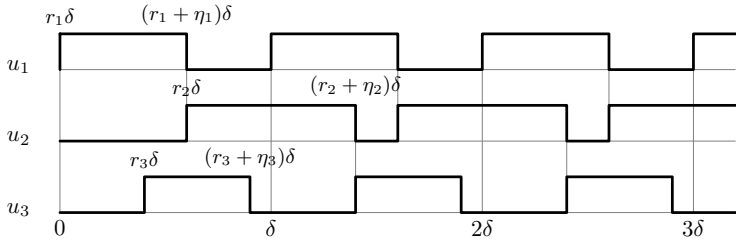
A.1.3. Proof of Theorem 2.2

Proof We prove this Theorem by constructing a safe δ-periodic schedule $u(\cdot)$ that satisfies the peak constraint. By Lemma 2.1, for each i, there exists $\delta_i^\star > 0$ such that for any $0 < \delta < \delta_i^\star$, the δ-periodic schedule $u_i(\cdot)$ specified in Equation (2.7) on page 27 will realize the state variable x_i safe. The time period δ is chosen so that $0 < \delta < \min\{\delta_1^\star, \ldots, \delta_n^\star\}$.

Because $\sum_{i=1}^n \underline{\eta}_i < k$ we can always select utilization values η_i so that $\underline{\eta}_i < \eta_i < \bar{\eta}_i$ for each i and that $\sum_{i=1}^n \eta_i \leq k$. We then distribute n non-overlapping right-open intervals, each of length η_i respectively, into the interval $[0, k]$ on the real line (Figure A.1a on the current



(a) Distribution of n non-overlapping intervals into $[0, k]$.



(b) Timing diagram of the constructed schedules $u_i(\cdot)$ from the interval distribution: $r_1 = s_1$, $r_2 = s_2$, $r_3 = s_3 - 1$. At any time t, $\|u(t)\|_1 \leq k = 2$.

Figure A.1: Illustration of the construction of safe periodic schedules for $n = 3$ and $k = 2$.

page). Let interval i be $[s_i, s_i + \eta_i) \subseteq [0, k]$. Since $\sum_{i=1}^{n} \eta_i \leq k$, such a distribution is always possible. Each control input $u_i(\cdot)$ is then constructed as

$$u_i(t) = \begin{cases} 1 & \text{if } (r_i + j)\,\delta \leq t < (r_i + j + \eta_i)\,\delta,\ j \in \mathbb{N} \\ 0 & \text{otherwise} \end{cases}$$

in which $r_i := s_i - \lfloor s_i \rfloor \in [0, 1)$. That is, $u_i(\cdot)$ is δ-periodic of the form in Equation (2.7) after some initial delay $r_i\delta$ which does not exceed one time period δ. Because the delay is finite, the safety of the resulted trajectory is not affected. Figure A.1b on this page illustrates this construction for $n = 3$ and $k = 2$.

We will show that the constructed schedules satisfy the peak constraint $\|u(t)\|_1 \leq k$ for all $t \geq 0$. Consider any time $t = (j + \tau)\delta$ where $j \in \mathbb{N}$ and $0 \leq \tau < 1$. By construction, $u_i(t) = 1$ if and only if the interval $[s_i, s_i + \eta_i)$ contains one of the points $\{\tau, 1 + \tau, \ldots, k - 1 + \tau\}$, and only one point since $\eta_i < 1$. Because these intervals are non-overlapping in $[0, k]$, the number of control inputs $u_i(t)$ that are 1, i.e., $\|u(t)\|_1$, is equal to the number of points $\{\tau, 1 + \tau, \ldots, k - 1 + \tau\}$ that lie in those intervals. Since there are at most k of these points, $\|u(t)\|_1 \leq k$. ∎

A.1.4. Proof of Theorem 2.3

PROOF Define a differentiable function $g : \mathcal{X} \to \mathbb{R}$ of state as

$$g(x) = \sum_{i=1}^{n} \frac{\underline{\eta}_i}{a_{\text{off},i} l_i - b_{\text{off},i}} (x_i - l_i)$$

Because $\underline{\eta}_i > 0$ and $a_{\text{off},i} l_i - b_{\text{off},i} > 0$ for each i (cf. Equation (2.6) on page 25), it is straightforward to see that $g(x) \geq 0$ for all $x \in \texttt{Safe} = [l_1, h_1] \times \cdots \times [l_n, h_n]$.

Consider the time derivative of g along the flow of the dynamics (cf. Equation (2.3b) on page 24)

$$\nabla g(x) \cdot f(x, u) = \sum_{i=1}^{n} \frac{\underline{\eta}_i}{a_{\text{off},i} l_i - b_{\text{off},i}} f_i(x_i, u_i)$$

for $x \in \texttt{Safe}$ and $u \in \mathcal{U}$, in which $f_i(x_i, u_i)$ is the right-hand side of the differential equation (2.4) on page 25. When $u_i = 0$, we have

$$\frac{\underline{\eta}_i}{a_{\text{off},i} l_i - b_{\text{off},i}} f_i(x_i, u_i) = \underline{\eta}_i \frac{-a_{\text{off},i} x_i + b_{\text{off},i}}{a_{\text{off},i} l_i - b_{\text{off},i}} \leq \underline{\eta}_i \frac{-a_{\text{off},i} l_i + b_{\text{off},i}}{a_{\text{off},i} l_i - b_{\text{off},i}} = -\underline{\eta}_i$$

while when $u_i = 1$ we have

$$\frac{\underline{\eta}_i}{a_{\text{off},i} l_i - b_{\text{off},i}} f_i(x_i, u_i) = \underline{\eta}_i \frac{-a_{\text{on},i} x_i + b_{\text{on},i}}{a_{\text{off},i} l_i - b_{\text{off},i}} \leq \underline{\eta}_i \frac{-a_{\text{on},i} l_i + b_{\text{on},i}}{a_{\text{off},i} l_i - b_{\text{off},i}} = 1 - \underline{\eta}_i$$

In the above calculations, we used the definition of $\underline{\eta}_i$ in Equation (2.8) on page 28. It follows that for all $u \in \mathcal{U}$ (i.e., $\|u\|_1 \leq k$) $\nabla g(x) \cdot f(x, u) \leq k - \sum_{i=1}^{n} \underline{\eta}_i$. If $\sum_{i=1}^{n} \underline{\eta}_i < k$ then $\nabla g(x) \cdot f(x, u) \leq -\epsilon$ where $\epsilon = \sum_{i=1}^{n} \underline{\eta}_i - k > 0$. By Theorem 2.1 on page 24, the system is not k-schedulable.

In the special case when $\sum_{i=1}^{n} \underline{\eta}_i = k$, the time derivative of $g(x(t))$ is zero if and only if all of the above inequalities are equalities, that is

- $\|u\|_1 = k$, i.e., exactly k of u_i are 1; and

- $x_i = l_i$ for all i.

Because each x_i always either increases or decreases at l_i (Equation (2.5)), the above equalities do not hold most of the time. Therefore, $g(x(t))$ will almost always decrease as long as $x(t) \in \texttt{Safe}$, regardless of $u(t)$. By the proof of Theorem 2.1, we can conclude that if $\sum_{i=1}^{n} \underline{\eta}_i = k$ then the system is not k-schedulable. ■

A.1.5. Proof of Lemma 2.3

PROOF Because state matrix A is Hurwitz, there exist positive constants α, β such that $\left\|\mathrm{e}^{At}\right\| \leq \beta \mathrm{e}^{-\alpha t}$ for all $t \geq 0$. It follows from the state error solution in Equation (2.14) on page 38 that for any $t \geq 0$,

$$\begin{aligned} \|\xi(t)\| &= \left\| \left(\sum_{i=0}^{\sigma-1} \mathrm{e}^{A(t-(i+1)\delta)} \right) \xi_\delta + \xi(t-\sigma\delta) \right\| \\ &\leq \left(\sum_{i=0}^{\sigma-1} \left\| \mathrm{e}^{A(t-(i+1)\delta)} \right\| \right) \|\xi_\delta\| + \|\xi(t-\sigma\delta)\| . \end{aligned} \tag{A.3}$$

The matrix norm inside the sum is bounded by $\left\|\mathrm{e}^{A(t-(i+1)\delta)}\right\| \leq \beta \mathrm{e}^{-\alpha(t-(i+1)\delta)}$. Therefore,

$$\sum_{i=0}^{\sigma-1} \left\| \mathrm{e}^{A(t-(i+1)\delta)} \right\| \leq \beta \sum_{i=0}^{\sigma-1} \mathrm{e}^{-\alpha(t-(i+1)\delta)} = \beta \mathrm{e}^{-\alpha(t-\sigma\delta)} \sum_{i=0}^{\sigma-1} \mathrm{e}^{-i\alpha\delta}$$

where $\mathrm{e}^{-\alpha(t-\sigma\delta)} \leq 1$ because $\sigma = \lfloor t/\delta \rfloor \leq t/\delta$, hence

$$\sum_{i=0}^{\sigma-1} \left\| \mathrm{e}^{A(t-(i+1)\delta)} \right\| \leq \beta \frac{1-\mathrm{e}^{-\sigma\alpha\delta}}{1-\mathrm{e}^{-\alpha\delta}} \leq \beta \frac{1}{1-\mathrm{e}^{-\alpha\delta}}. \tag{A.4}$$

From Equation (2.12) on page 37 we have

$$\xi(t-\sigma\delta) = \int_0^{t-\sigma\delta} \mathrm{e}^{A((t-\sigma\delta)-s)} B\left(u(s)-\eta\right) \, ds.$$

Consider the term $B\left(u(s)-\eta\right)$. Because $u(s)$ is a binary vector of length m, there are

only a finite number of possible values of $u(s)$. It follows that $\|B\left(u(s)-\eta\right)\|$ is bounded above by some finite constant γ, that is $\|B\left(u(s)-\eta\right)\| \leq \gamma$ for all $s \geq 0$. The last term in Equation (A.3) can then be bounded by:

$$\begin{aligned}
\|\xi(t-\sigma\delta)\| &\leq \int_0^{t-\sigma\delta} \left\| \mathrm{e}^{A((t-\sigma\delta)-s)} \right\| \|B\left(u(s)-\eta\right)\| \, ds \\
&\leq \gamma\beta \int_0^{t-\sigma\delta} \mathrm{e}^{-\alpha((t-\sigma\delta)-s)} \, ds \\
&= \frac{\gamma\beta}{\alpha}\left(1-\mathrm{e}^{-\alpha(t-\sigma\delta)}\right) \\
&\leq \frac{\gamma\beta}{\alpha}\alpha\left(t-\sigma\delta\right) \\
&\leq \gamma\beta\delta
\end{aligned} \tag{A.5}$$

in which we use the inequality $1-\mathrm{e}^{-x} \leq x$ for all $x \geq 0$, and the fact that $0 \leq t-\sigma\delta < \delta$.

The same bound could have been used for $\|\xi_\delta\|$, however it would make the bound in Equation (A.3) not go to 0 as $\delta \to 0$, which is undesirable. A better bound for $\|\xi_\delta\|$ can be achieved by rewriting ξ_δ as:

$$\begin{aligned}
\xi_\delta &= \int_0^\delta \mathrm{e}^{A(\delta-s)} B\left(u(s)-\eta\right) \, ds \\
&= \int_0^\delta \left(\mathrm{e}^{A(\delta-s)} - \mathrm{I}\right) B\left(u(s)-\eta\right) \, ds + B\int_0^\delta \left(u(s)-\eta\right) \, ds
\end{aligned}$$

and noting that $\int_0^\delta \left(u(s)-\eta\right) \, ds = 0$ by definition of the utilization vector, and that $\mathrm{e}^{A(\delta-s)} - \mathrm{I} = \left(\int_0^{\delta-s} \mathrm{e}^{Av} \, dv\right) A$, hence

$$= \int_0^\delta \left(\int_0^{\delta-s} \mathrm{e}^{Av} \, dv\right) AB\left(u(s)-\eta\right) \, ds.$$

The inner integral is bounded by

$$\left\| \int_0^{\delta-s} \mathrm{e}^{Av} \, dv \right\| \leq \int_0^{\delta-s} \left\| \mathrm{e}^{Av} \right\| \, dv \leq \beta \int_0^{\delta-s} \mathrm{e}^{-\alpha v} \, dv = \frac{\beta}{\alpha}\left(1-\mathrm{e}^{-\alpha(\delta-s)}\right) \leq \beta(\delta-s).$$

Therefore,

$$\begin{aligned}\|\xi_\delta\| &\leq \int_0^\delta \left\| \int_0^{\delta-s} \mathrm{e}^{Av}\, dv \right\| \|A\| \|B\left(u(s)-\eta\right)\|\, ds \\ &\leq \|A\|\,\gamma\beta \int_0^\delta (\delta - s)\, ds \\ &= \frac{1}{2}\|A\|\,\gamma\beta\delta^2. \end{aligned} \tag{A.6}$$

Combining bounds (A.3) to (A.6) gives us the upper-bound in Lemma 2.3. ∎

A.1.6. Proof of Theorem 2.6

PROOF Condition 2 implies that there exists $\epsilon > 0$ such that $\mathfrak{B}(\bar{x}^\star, \epsilon) \subseteq \texttt{Safe}$ where $\bar{x}^\star = -A^{-1}(B_0 + B\eta)$ is the equilibrium of the average system (2.10). Because the average system is uniformly exponentially stable, for any initial state $\bar{x}_0 = x_0$, there exists a finite $T_{\epsilon,x_0} \geq 0$ such that $\|\bar{x}(t) - \bar{x}^\star\| < \frac{\epsilon}{2}$ for all $t \geq T_{\epsilon,x_0}$. Also, by Lemma 2.4, there exists $\delta_\epsilon > 0$ such that for any δ-periodic control signal $u(\cdot)$ with $0 < \delta \leq \delta_\epsilon$ and with utilization η, $\|x(t) - \bar{x}(t)\| < \frac{\epsilon}{2}$ for all $t \geq 0$. Therefore $\|x(t) - \bar{x}^\star\| < \epsilon$, hence $x(t) \in \texttt{Safe}$, for all $t \geq T_{\epsilon,x_0}$. Condition 1 implies that there exist δ-periodic control signals $u(\cdot)$ satisfying $\|u(t)\|_1 \leq k$ for all $t \geq 0$ (cf. Appendix A.1.3). This concludes the proof. ∎

A.1.7. Proof of Theorem 2.7

PROOF Define $\mathcal{P} \subset \mathcal{X}$ to be the set of all equilibrium points of the average system for all values of η

$$\mathcal{P} := \{-A^{-1}(B_0 + B\eta) \,:\, \eta \in [0,1]^m, \|\eta\|_1 \leq k\}.$$

It is straightforward to verify that the set $\{\eta \,:\, \eta \in [0,1]^m, \|\eta\|_1 \leq k\}$ is convex and compact, hence $\mathcal{P}$ as an affine image of this set is also convex and compact (Boyd and Vandenberghe, 2006; Kreyszig, 1989). The safe set `Safe` is convex and compact by assumption (cf. Section 2.4.1).

Suppose there is no utilization vector η satisfying $-A^{-1}(B_0 + B\eta) \in \texttt{Safe}$. Then $\mathcal{P}$ and $\texttt{Safe}$ are disjoint. The separating hyperplane theorem (see Boyd and Vandenberghe, 2006, sec. 2.5) states that for any two disjoint non-empty convex sets, one of which is compact and the other is closed, there exists a hyperplane that strictly separates them. Therefore, there exist a vector $a \neq 0$ and numbers b and $\epsilon > 0$ such that $a^T x \leq b - \epsilon \; \forall x \in \texttt{Safe}$ and $a^T x \geq b \; \forall x \in \mathcal{P}$. Define the differentiable function $g : \mathcal{X} \to \mathbb{R}$ of state as $g(x) := \alpha^T x - \beta$ where $\alpha^T = a^T A^{-1} \neq 0$ and $\beta = \inf_{x \in \texttt{Safe}} \alpha^T x$ is finite as $\texttt{Safe}$ is bounded. It is obvious that $g(x) \geq 0$ for all $x \in \texttt{Safe}$. Its time derivative along the flow of the dynamics (2.9) is

$$\begin{aligned}
\dot{g}(x(t)) &= \alpha^T \left(Ax(t) + B_0 + Bu(t)\right) \\
&= a^T \left(x(t) + A^{-1}(B_0 + Bu(t))\right) \\
&= a^T x(t) - a^T \left(-A^{-1}(B_0 + Bu(t))\right)
\end{aligned}$$

which satisfies the inequality $\dot{g}(x(t)) \leq b - \epsilon - b = -\epsilon$, $\forall x(t) \in \texttt{Safe} \wedge \forall u(t) \in \mathcal{U}$, due to the property of the strictly separating hyperplane. By Theorem 2.1 on page 24, the system is not k-schedulable. ∎

A.1.8. Proof of Lemma 2.5

To prove Lemma 2.5, we will need the following preliminary result about the integral of the product of a continuous, exponentially bounded function and a zero-mean periodic function under fast switching.

Lemma A.1 *Let $f_\delta : \mathbb{R}^+ \to \mathbb{R}$ and $g_\delta : \mathbb{R}^+ \to \mathbb{R}$ be two families of functions parameterized by $\delta > 0$. Let $\delta^\star > 0$ be such that for all $\delta \leq \delta^\star$,*

1. *f_δ is continuous.*
2. *f_δ is piecewise differentiable and its derivative is globally bounded by $\gamma > 0$, i.e., $|f'_\delta(t)| \leq \gamma$ for all $t \geq 0$ where $f'_\delta(t)$ exists.*
3. *$f_\delta(t) \leq \beta \mathrm{e}^{-\alpha t} \; \forall t \geq 0$ where α and β are two positive numbers independent of δ.*

4. *g_δ is piecewise continuous and globally bounded by $\chi > 0$, i.e., $|g_\delta(t)| \le \chi \; \forall t \ge 0$.*

5. *g_δ has zero mean on every interval $[i\delta, (i+1)\delta]$, i.e., $\int_{i\delta}^{(i+1)\delta} g_\delta(\sigma)\, d\sigma = 0$ for all integers $i \ge 0$.* □

Then for any $\epsilon > 0$, there exists $\delta_\epsilon > 0$ such that for all $0 < \delta \le \delta_\epsilon$,

$$\left| \int_0^t f_\delta(\sigma) g_\delta(\sigma)\, d\sigma \right| \le \epsilon \qquad \forall t \ge 0. \tag{A.7}$$

PROOF Given any $\epsilon > 0$. Let

$$\theta = \frac{\epsilon}{2\chi} \frac{1 - \mathrm{e}^{-\alpha\delta^\star}}{\delta^\star} > 0$$

and

$$t_\theta = \frac{1}{\alpha} \log \frac{2\beta}{\theta}.$$

Without loss of generality, assume ϵ is small enough so that $\theta < 2\beta$ and thus $t_\theta > 0$. It is obvious that t_θ satisfies $|f_\delta(t)| \le \beta \mathrm{e}^{-\alpha t} \le \theta/2$ for all $t \ge t_\theta$ and all $\delta \le \delta^\star$. Let

$$\zeta = \frac{\gamma}{2} \left(\sqrt{t_\theta^2 + \frac{\epsilon}{\gamma\chi}} - t_\theta \right) > 0$$

and

$$\delta_\epsilon = \min \left\{ \frac{\zeta}{\gamma}, \frac{\epsilon}{4\beta\chi}, \delta^\star \right\} > 0.$$

We will show that for all $\delta \le \delta_\epsilon$, inequality (A.7) is satisfied.

Consider any $\delta \le \delta_\epsilon$ and any $t \ge 0$. Because $\delta \le \delta_\epsilon \le \delta^\star$, all the hypotheses of the lemma hold. Let $s = \lfloor t/\delta \rfloor \ge 0$ that verifies $t - s\delta < \delta \le \delta_\epsilon \le \frac{\epsilon}{4\beta\chi}$. The integral in (A.7) can be written as

$$\int_0^t f_\delta(\sigma) g_\delta(\sigma)\, d\sigma = \sum_{i=0}^{s-1} \int_{i\delta}^{(i+1)\delta} f_\delta(\sigma) g_\delta(\sigma)\, d\sigma + \int_{s\delta}^t f_\delta(\sigma) g_\delta(\sigma)\, d\sigma$$

in which the sum of integrals vanishes if $s = 0$. Hypotheses 3 and 4 implies that the second

integral is bounded

$$\left|\int_{s\delta}^{t} f_\delta(\sigma)g_\delta(\sigma)\,d\sigma\right| \le \int_{s\delta}^{t} |f_\delta(\sigma)|\,|g_\delta(\sigma)|\,d\sigma \le \beta\chi(t-s\delta) \le \beta\chi\frac{\epsilon}{4\beta\chi} = \frac{\epsilon}{4}. \tag{A.8}$$

If $s = 0$ then (A.8) implies (A.7).

When $s > 0$, within the compact interval $[i\delta, (i+1)\delta]$, where $0 \le i < s$, we have

$$\begin{aligned}\int_{i\delta}^{(i+1)\delta} f_\delta(\sigma)g_\delta(\sigma)\,d\sigma &= \int_{i\delta}^{(i+1)\delta} (f_\delta(\sigma) - f_\delta(i\delta))\,g_\delta(\sigma)\,d\sigma + \int_{i\delta}^{(i+1)\delta} f_\delta(i\delta)g_\delta(\sigma)\,d\sigma \\ &= \int_{i\delta}^{(i+1)\delta} (f_\delta(\sigma) - f_\delta(i\delta))\,g_\delta(\sigma)\,d\sigma\end{aligned}$$

in which the second integral vanishes due to the zero-mean property of g_δ (hypothesis 5). From hypotheses 1, 2 and that $\delta \le \zeta/\gamma$, we have $|f_\delta(t) - f_\delta(i\delta)| \le \delta\gamma \le \zeta$ for all $t \in [i\delta, (i+1)\delta]$. Furthermore, since the bound on $|f_\delta(t)|$ is decreasing with t (hypothesis 3), we also have $|f_\delta(t) - f_\delta(i\delta)| \le 2\beta\mathrm{e}^{-\alpha i\delta}$. It follows that

$$\left|\int_{i\delta}^{(i+1)\delta} f_\delta(\sigma)g_\delta(\sigma)\,d\sigma\right| \le \int_{i\delta}^{(i+1)\delta} |f_\delta(\sigma) - f_\delta(i\delta)|\,|g_\delta(\sigma)|\,d\sigma \le \delta\chi\min\{\zeta, 2\beta\mathrm{e}^{-\alpha i\delta}\} \tag{A.9}$$

for every $i < s$. Therefore,

$$\begin{aligned}\left|\int_0^t f_\delta(\sigma)g_\delta(\sigma)\,d\sigma\right| &\le \sum_{i=0}^{s-1}\left|\int_{i\delta}^{(i+1)\delta} f_\delta(\sigma)g_\delta(\sigma)\,d\sigma\right| + \left|\int_{s\delta}^{t} f_\delta(\sigma)g_\delta(\sigma)\,d\sigma\right| \\ &\le \delta\chi\sum_{i=0}^{s-1}\min\{\zeta, 2\beta\mathrm{e}^{-\alpha i\delta}\} + \frac{\epsilon}{4}.\end{aligned}$$

Because the right-hand side is increasing with s, the left-hand side is bounded by the limit of the right-hand side as $s \to \infty$. In particular,

$$\left|\int_0^t f_\delta(\sigma)g_\delta(\sigma)\,d\sigma\right| \le \frac{\epsilon}{4} + \delta\chi\sum_{i=0}^{\infty}\min\{\zeta, 2\beta\mathrm{e}^{-\alpha i\delta}\}$$

for any $t \ge 0$. Let $\bar{s} = \lceil t_\theta/\delta \rceil \ge 1$ which satisfies $t_\theta \le \bar{s}\delta < t_\theta + \delta \le t_\theta + \delta_\epsilon$. For all $i \ge \bar{s}$ we

have

$$\beta \mathrm{e}^{-\alpha i \delta} = \beta \mathrm{e}^{-\alpha \bar{s} \delta} \mathrm{e}^{-\alpha (i - \bar{s}) \delta} \leq \beta \mathrm{e}^{-\alpha t_\theta} \mathrm{e}^{-\alpha (i - \bar{s}) \delta} = \frac{\theta}{2} \mathrm{e}^{-\alpha (i - \bar{s}) \delta}$$

where the last equality comes from the definition of t_θ. Consequently,

$$\begin{aligned}
\left| \int_0^t f_\delta(\sigma) g_\delta(\sigma) \, d\sigma \right| &\leq \frac{\epsilon}{4} + \delta \chi \sum_{i=0}^{\bar{s}-1} \zeta + \delta \chi \sum_{i=\bar{s}}^{\infty} 2\beta \mathrm{e}^{-\alpha i \delta} \\
&\leq \frac{\epsilon}{4} + \bar{s} \delta \chi \zeta + \delta \chi \theta \sum_{i=0}^{\infty} \mathrm{e}^{-\alpha i \delta} \\
&\leq \frac{\epsilon}{4} + (t_\theta + \delta_\epsilon) \chi \zeta + \chi \theta \frac{\delta}{1 - \mathrm{e}^{-\alpha \delta}} \\
&\leq \frac{\epsilon}{4} + (t_\theta + \delta_\epsilon) \chi \zeta + \chi \theta \frac{\delta^\star}{1 - \mathrm{e}^{-\alpha \delta^\star}}
\end{aligned}$$

in which the last inequality follows from the monotonicity of $\frac{\delta}{1 - \mathrm{e}^{-\alpha \delta}}$ in δ. Recall that $\delta_\epsilon \leq \frac{\zeta}{\gamma}$ and plug in the definitions of ζ and θ, we have for any $t \geq 0$

$$\begin{aligned}
\left| \int_0^t f_\delta(\sigma) g_\delta(\sigma) \, d\sigma \right| &\leq \frac{\epsilon}{4} + \frac{\gamma \chi}{4} \left(\sqrt{t_\theta^2 + \frac{\epsilon}{\gamma \chi}} + t_\theta \right) \left(\sqrt{t_\theta^2 + \frac{\epsilon}{\gamma \chi}} - t_\theta \right) + \frac{\epsilon}{2} \\
&= \frac{3\epsilon}{4} + \frac{\gamma \chi}{4} \left(t_\theta^2 + \frac{\epsilon}{\gamma \chi} - t_\theta^2 \right) \\
&= \epsilon
\end{aligned}$$

Therefore, inequality (A.7) holds for all $\delta \leq \delta_\epsilon$. ∎

Because u belongs to a finite set $\mathcal{U}$ of binary vectors, the periodic control signal $u(\cdot)$ is piecewise constant of the form

$$u(t) = \begin{cases} u^1 & \text{if } t \bmod \delta \in [0, w_1 \delta) \\ u^2 & \text{if } t \bmod \delta \in [w_1 \delta, (w_1 + w_2) \delta) \\ \vdots & \\ u^q & \text{if } t \bmod \delta \in [(w_1 + w_2 + \ldots w_{q-1}) \delta, \delta) \end{cases} \tag{A.10}$$

where $u^j \in \mathcal{U}$ for all $1 \leq j \leq q$, $u^j \neq u^{j+1}$ for all $1 \leq j \leq q-1$, and w_j are positive real numbers such that $\sum_{j=1}^{q} w_j = 1$. It is straightforward to verify that $\eta_i = \sum_{j=1}^{q} w_j u_i^j$ for every $1 \leq i \leq m$ and therefore $\overline{A}_\eta = \sum_{j=1}^{q} w_j \left(A_0 + \sum_{i=1}^{m} A_i u_i^j \right)$. Define the δ-periodic time-varying matrix $A(t) := (A_0 + \sum_{i=1}^{m} A_i u_i(t))$. We have that $\overline{A}_\eta = \frac{1}{\delta} \int_t^{t+\delta} A(t)\, d\sigma$ for all $t \geq 0$.

From linear system theory (Rugh, 1996), the solution of the system (2.16) with switching state matrix is given by

$$x(t) = \Phi(t,0)x_0 + \int_0^t \Phi(t,\sigma)\left(B_0 + Bu(\sigma)\right)\, d\sigma$$

where $\Phi(t,\sigma)$ is the state transition matrix of $A(t)$. Note that $\Phi(\cdot,\cdot)$ is not periodic. The following Lemma (adapted from Lemmas 3.22 and 2.11 in Sun and Ge, 2005) provides an estimation for the state transition matrix under fast switching.

Lemma A.2 *If $\overline{A}_\eta$ is Hurwitz then there exist positive numbers $\delta^\star$, α and β such that the transition matrix $\Phi(t,\sigma)$ with period $\delta \leq \delta^\star$ is exponentially convergent by*

$$\|\Phi(t,\sigma)\|_2 \leq \beta \mathrm{e}^{-\alpha(t-\sigma)} \qquad \forall t \geq \sigma. \qquad \square$$

PROOF Since $\overline{A}_\eta$ is Hurwitz, there exist positive numbers κ and γ such that for all $t \geq \sigma$,

$$\|\Psi(t,\sigma)\|_2 \leq \kappa \mathrm{e}^{-\gamma(t-\sigma)}$$

where $\Psi(t,\sigma)$ is the state transition matrix of the average system. Let α be any number such that $0 < \alpha < \gamma$. Then the result follows from Lemma 2.11 in (Sun and Ge, 2005) by letting $\epsilon = \gamma - \alpha$. ■

A consequence of Lemma A.2 is that each element $\phi_{ij}(t,\sigma)$ of the state transition matrix, for $1 \leq i,j \leq n$, is also exponentially convergent by $|\phi_{ij}(t,\sigma)| \leq \beta \mathrm{e}^{-\alpha(t-\sigma)}$, $\forall t \geq \sigma$.

We now prove Lemma 2.5.

PROOF (LEMMA 2.5) Define the error between the trajectories of the switched system and its average system as $\xi(t) = x(t) - \bar{x}(t)\ \forall t \geq 0$ and differentiate it, we have

$$\begin{aligned}\dot{\xi}(t) &= \left(A_0 + \sum_{i=1}^{m} A_i u_i(t)\right) x(t) + B_0 + Bu(t) - \overline{A}_\eta \bar{x}(t) - \overline{B}_\eta \\ &= A(t)\xi(t) + \Delta A(t)\bar{x}(t) + \Delta B(t)\end{aligned} \tag{A.11}$$

where

$$\Delta A(t) := A(t) - \overline{A}_\eta, \qquad \Delta B(t) := Bu(t) - \overline{B}_\eta$$

and with initial error $\xi(0) = x(0) - \bar{x}(0) = 0$. System (A.11) is a switched affine system similar to (2.16), however it depends on $\bar{x}(t)$. Matrix $\Delta A(t)$ and vector $\Delta B(t)$ are both δ-periodic, while the state transition matrix $\Phi(t, \sigma)$ of $A(t)$ is not. The solution to (A.11) is

$$\xi(t) = \int_0^t \Phi(t, \sigma) \left(\Delta A(u(\sigma))\bar{x}(\sigma) + \Delta B(u(\sigma))\right)\, d\sigma, \qquad \forall t \geq 0. \tag{A.12}$$

We note that except for $\bar{x}$ and the independent variables, all elements in (A.12) are dependent on the period δ although this is not explicitly indicated for brevity.

Let ϕ_{ij}, Δa_{ij}, Δb_j, ξ_i, and $\bar{x}_j$ represent the elements of Φ, ΔA, ΔB, ξ, and $\bar{x}$. Then from Equation (A.12) we can write each $\xi_i(t)$ as

$$\xi_i(t) = \sum_{j=1}^{n} \sum_{k=1}^{n} \int_0^t \phi_{ij}(t, \sigma) \Delta a_{jk}(\sigma) \bar{x}_k(\sigma)\, d\sigma + \sum_{j=1}^{n} \int_0^t \phi_{ij}(t, \sigma) \Delta b_j(\sigma)\, d\sigma. \tag{A.13}$$

Because $\overline{A}_\eta$ is Hurwitz, by Lemma A.2, there exist $\delta^\star > 0$, $\alpha > 0$, and $\beta > 0$ such that with any time period $\delta \leq \delta^\star$, $|\phi_{ij}(t, \sigma)| \leq \beta \mathrm{e}^{-\alpha(t-\sigma)}\ \forall t \geq \sigma$. We wish to bound each integral in (A.13) by $\tilde{\epsilon} = \frac{\epsilon}{n^2+n} > 0$ by applying Lemma A.1, so that $|\xi_i(t)| \leq \epsilon$ for all $t \geq 0$.

Consider the first integral in (A.13), which can be written as $\int_0^t \phi_{ij}(t, \sigma) \Delta a_{jk}(\sigma) \bar{x}_k(\sigma)\, \mathrm{d}\sigma = \int_0^t f_\delta(\sigma) g_\delta(\sigma)\, \mathrm{d}\sigma$ where $f_\delta(\sigma) = \phi_{ij}(t, t-\sigma)\bar{x}_k(t-\sigma)$ and $g_\delta(\sigma) = \Delta a_{jk}(t-\sigma)$. Note that the dependence of these functions on δ is explicitly indicated. We will verify that all hypotheses

of Lemma A.1 are satisfied for all $0 < \delta \leq \delta^\star$:

1. Since $\phi_{ij}(t,\cdot)$ and $\bar{x}_k(\cdot)$ are continuous, $f_\delta(\cdot)$ is continuous.

2. Because $\phi_{ij}(t,\cdot)$ is piecewise differentiable and $\bar{x}_k(\cdot)$ is differentiable, $f_\delta(\cdot)$ is piecewise differentiable. Moreover, its derivative is given by

$$\begin{aligned} f'_\delta(\sigma) &= -\phi'_{ij}(t,t-\sigma)\bar{x}_k(t-\sigma) + \phi_{ij}(t,t-\sigma)\bar{x}'_k(t-\sigma) \\ &= -\bar{x}_k(t-\sigma)\sum_{l=1}^{n}\phi_{il}(t,t-\sigma)A_{lj}(t-\sigma) \\ &\quad + \phi_{ij}(t,t-\sigma)\sum_{l=1}^{n}\left([\overline{A}_\eta]_{kl}\bar{x}_l(t-\sigma) + [\overline{B}_\eta]_k\right) \end{aligned}$$

   in which we use Equation (2.17) and the equality $\frac{\partial\Phi(t,\sigma)}{\partial\sigma} = -\Phi(t,\sigma)A(\sigma)$ (Rugh, 1996). Since $\overline{A}_\eta$ is Hurwitz, $|\bar{x}_k(t-\sigma)|$ is globally bounded by $\|\bar{x}(\cdot)\|_\infty$ which is finite. Also, $|\phi_{il}(t,t-\sigma)|$ is bounded by β and $|A_{lj}(t-\sigma)|$ is finitely bounded by $\max_{u\in\mathcal{U}}\|A_0 + \sum_{i=1}^m A_i u_i\|_{\max}$. Thus, $|f'_\delta(\sigma)| \leq \gamma\ \forall\sigma \geq 0$, for some finite constant γ.

3. For all $\sigma \geq 0$, $|f_\delta(\sigma)| \leq \|\bar{x}(\cdot)\|_\infty \beta \mathrm{e}^{-\alpha\sigma}$.

4. $g_\delta(\sigma) = \Delta a_{jk}(t-\sigma)$ is piecewise constant (thus piecewise continuous) and globally bounded by $\chi = \max_{u\in\mathcal{U}}\left\|A_0 + \sum_{i=1}^m A_i u_i - \overline{A}_\eta\right\|_{\max}$.

5. On every interval $[i\delta,(i+1)\delta]$, $i \geq 0$, we have

$$\int_{i\delta}^{(i+1)\delta} g_\delta(\sigma)\,\mathrm{d}\sigma = \int_{i\delta}^{(i+1)\delta}\Delta a_{jk}(t-\sigma)\,\mathrm{d}\sigma = \int_{i\delta}^{(i+1)\delta} A_{jk}(t-\sigma)\,\mathrm{d}\sigma - \delta[\overline{A}_\eta]_{jk} = 0.$$

By Lemma A.1, there exists $\delta_{\tilde{\epsilon}} > 0$ such that for all $0 < \delta \leq \delta_{\tilde{\epsilon}}$,

$$\left|\int_0^t \phi_{ij}(t,\sigma)\Delta a_{jk}(\sigma)\bar{x}_k(\sigma)\,d\sigma\right| \leq \tilde{\epsilon}.$$

Similarly, Lemma A.1 can be applied to all integrals in Equation (A.13) for all elements in

Equation (A.12). By selecting the minimum of these $\delta_{\tilde{\epsilon}}$, we obtain $\delta_\epsilon > 0$ such that for all $0 < \delta \leq \delta_\epsilon$,

$$|\xi_i(t)| \leq n^2\tilde{\epsilon} + n\tilde{\epsilon} = \epsilon$$

and thus $\|\xi(t)\|_\infty = \|x(t) - \bar{x}(t)\|_\infty \leq \epsilon$ for all $t \geq 0$. ■

A.1.9. Proof of Theorem 2.9

The proof of Theorem 2.9 is adapted from that of Theorem 2 in (Girard and Pappas, 2007a) and requires several preliminary results. In the following, we will consider the control system (2.20) and its solution. We also assume all the hypotheses of Theorem 2.9.

Lemma A.3 *Let $x(0) \in \mathbb{R}^n$ be any initial state. Then for all disturbance signals $d(\cdot)$ and all control inputs $u(\cdot)$, the trajectory $x(\cdot)$ satisfies for all $0 \leq t \leq t'$*

$$\left\|x(t') - x(t)\right\| \leq \sup_{(u,d)\in\mathcal{U}\times\mathcal{D}} \|f(x(t),u,d)\| \, \frac{\mathrm{e}^{\lambda(t'-t)} - 1}{\lambda},$$

where λ is the Lipschitz constant of f. □

The proof of this Lemma can be found in (Girard and Pappas, 2007a, Lemma 2). Note that because $\mathcal{U}$ and $\mathcal{D}$ are compact, $\sup_{(u,d)\in\mathcal{U}\times\mathcal{D}} \|f(x(t),u,d)\|$ is finite.

Lemma A.4 *Let $x(0) \in \mathbb{R}^n$ be any initial state and $T > 0$. Then for any $\epsilon > 0$, there exists $h > 0$ such that for all disturbance signals $d(\cdot)$ and all control inputs $u(\cdot)$, the trajectory $x(\cdot)$ satisfies for all $d \in \mathcal{D}$, all $u \in \mathcal{U}$, and all $t, t' \in [0,T]$ with $0 \leq t \leq t' \leq t + h$,*

$$\left|\nabla V(x(t')) \cdot f(x(t'),u,d) - \nabla V(x(t)) \cdot f(x(t),u,d)\right| \leq \epsilon.$$ □

Proof (cf. Lemma 3 in (Girard and Pappas, 2007a)) Consider any $\epsilon > 0$. By Lemma A.3, we have for all $t \in [0,T]$, $x(t) \in \mathcal{C} := \mathfrak{B}(x(0), r)$ where $r = \sup_{(u,d)\in\mathcal{U}\times\mathcal{D}} \|f(x(0),u,d)\| \, \frac{\mathrm{e}^{\lambda T}-1}{\lambda}$. Since r is finite, $\mathcal{C}$ is compact. Because $\nabla V(x)\cdot f(x,u,d)$ is continuous, by the HeineâĂŞCantor theorem, it is uniformly continuous on the compact set $\mathcal{C}\times\mathcal{U}\times\mathcal{D}$. Thus, there exists $\zeta > 0$

such that for all $d \in \mathcal{D}$, all $u \in \mathcal{U}$ and all $x, x' \in \mathcal{C}$ with $\|x' - x\| \leq \zeta$, the following inequality holds

$$\left|\nabla V(x') \cdot f(x', u, d) - \nabla V(x) \cdot f(x, u, d)\right| \leq \epsilon. \tag{A.14}$$

Again, from Lemma A.3, there exists $h > 0$ such that for all $t, t' \in [0, T]$ with $t \leq t' \leq t + h$, $\|x(t') - x(t)\| \leq \zeta$. Clearly, $x(t), x(t') \in \mathcal{C}$. Therefore, the conclusion follows from Equation (A.14) applied to $x(t)$ and $x(t')$. ■

Lemma A.5 *Let $x(0) \in \mathbb{R}^n$ satisfying $V(x(0)) \leq \alpha^2$ and $T > 0$. Then for all disturbance signals $d(\cdot)$, there exists a control input $u(\cdot)$ such that the trajectory $x(\cdot)$ satisfies for all $t \in [0, T]$,*

$$V(x(t)) \leq \alpha^2.$$

□

PROOF Choose any $0 < \epsilon \leq \gamma$. Let $h > 0$ be given as in Lemma A.4. Without loss of generality, assume that $\frac{T}{h} = N \in \mathbb{N}$. Consider any $i \in \{0, 1, \ldots, N-1\}$ and suppose that $V(x(ih)) \leq \alpha^2$. Let $v(\cdot)$ be any control input for the sub-interval $[ih, (i+1)h]$ which results in a trajectory $z(\cdot)$. If for all $t \in [ih, (i+1)h]$, $V(z(t)) \leq \alpha^2$ then we can simply choose $u(t) = v(t)$ for all $t \in [ih, (i+1)h]$. Otherwise, let $t^\star \in [ih, (i+1)h)$ be the first time instant when $V(z(t^\star)) = \alpha^2$ and let $x^\star = z(t^\star)$. From Equation (2.21) on page 49, we can choose control input $u(\cdot)$ for the sub-interval $[ih, (i+1)h]$ such that $u(t) = v(t)\ \forall t \in [ih, t^\star)$ and for all $t \in [t^\star, (i+1)h]$, $\nabla V(x^\star) \cdot f(x^\star, u(t), d(t)) \leq -\gamma$. We have for all $t \in [t^\star, (i+1)h]$,

$$\begin{aligned} V(x(t)) - V(x^\star) &= \int_{t^\star}^{t} \nabla V(x(s)) \cdot f(x(s), u(s), d(s))\, ds \\ &\leq \int_{t^\star}^{t} \left(\nabla V(x^\star) \cdot f(x^\star, u(s), d(s)) + \epsilon\right)\, ds \\ &\leq (\epsilon - \gamma)\,(t - t^\star) \leq 0 \end{aligned}$$

in which the first inequality follows from Lemma A.4 and the last inequality comes from $\epsilon \leq \gamma$. Hence, for all $t \in [ih, (i+1)h]$, $V(x(t)) \leq \alpha^2$; in particular $V(x((i+1)h)) \leq \alpha^2$. Because $x(0)$ satisfies $V(x(0)) \leq \alpha$, we can conclude the Lemma. ■

Lemma A.6 *Let $x(0) \in \mathbb{R}^n$ satisfying $V(x(0)) > \alpha^2$ and $T > 0$. Then for all $\epsilon > 0$, $\epsilon \leq \gamma$,*

and for all disturbance signals $d(\cdot)$, there exists a control input $u(\cdot)$ such that the trajectory $x(\cdot)$ satisfies for all $t \in [0, T]$

$$V(x(t)) \leq \max\big(V(x(0)) + (\epsilon - \gamma)t, \alpha^2\big).$$

□

PROOF (cf. Lemma 4 in (Girard and Pappas, 2007a)) Consider any $\epsilon > 0$. Let $h > 0$ be given as in Lemma A.4. We then discretize the interval $[0, T]$ by time step h and assume without loss of generality that $T = Nh$ for some $N \in \mathbb{N}$. We will inductively construct control inputs $u(\cdot)$ for each sub-interval $[ih, (i+1)h]$ and concatenate them.

Consider the first sub-interval $[0, h]$. By Equation (2.21) on page 49, there exists a control input $u(t)$ for $t \in [0, h]$ such that for all $t \in [0, h]$, $\nabla V(x(0)) \cdot f(x(0), u(t), d(t)) \leq -\gamma$. We have for all $t \in [0, h]$,

$$\begin{aligned} V(x(t)) - V(x(0)) &= \int_0^t \nabla V(x(s)) \cdot f(x(s), u(s), d(s))\, ds \\ &\leq \int_0^t \left(\nabla V(x(0)) \cdot f(x(0), u(s), d(s)) + \epsilon\right)\, ds \\ &\leq (\epsilon - \gamma)t \end{aligned}$$

in which the first inequality comes from Lemma A.4. Hence, for all $t \in [0, h]$,

$$V(x(t)) - V(x(0)) \leq \max\big(V(x(0)) + (\epsilon - \gamma)t, \alpha^2\big).$$

Suppose that for some $i \in \{1, 2, \ldots, N-1\}$ we have a control input $u(\cdot)$ for $t \in [0, ih]$ such that for all $t \in [0, ih]$

$$V(x(t)) \leq \max\big(V(x(0)) + (\epsilon - \gamma)t, \alpha^2\big). \tag{A.15}$$

This holds for $i = 1$ as we have shown above. We will construct a control input for the sub-interval $[ih, (i+1)h]$. There are two cases:

- If $V(x(ih)) > \alpha^2$ then similar to the derivation for the first sub-interval, we can choose

a control input $u(\cdot)$ for sub-interval $[ih, (i+1)h]$ such that for all $t \in [ih, (i+1)h]$,

$$V(x(t)) - V(x(ih)) \leq (\epsilon - \gamma)(t - ih).$$

Note that since $V(x(ih)) > \alpha^2$, $V(x(0)) + (\epsilon - \gamma)ih > \alpha^2$ and thus $V(x(ih)) \leq V(x(0)) + (\epsilon - \gamma)ih$. Therefore, together with Equation (A.15), we have for all $t \in [ih, (i+1)h]$

$$V(x(t)) \leq V(x(0)) + (\epsilon - \gamma)t \leq \max\big(V(x(0)) + (\epsilon - \gamma)t, \alpha^2\big).$$

- If $V(x(ih)) \leq \alpha^2$ then from Lemma A.5, there exists a control input $u(\cdot)$ for the sub-interval $[ih, (i+1)h]$ such that for all $t \in [ih, (i+1)h]$,

$$V(x(t)) \leq \alpha^2 \leq \max\big(V(x(0)) + (\epsilon - \gamma)t, \alpha^2\big).$$

By the principle of induction, Equation (A.15) holds for $i = N$. Therefore, there exists a control input $u(\cdot)$ such that for all $t \in [0, T]$,

$$V(x(t)) \leq \max\big(V(x(0)) + (\epsilon - \gamma)t, \alpha^2\big).$$

■

We can now prove Theorem 2.9. Let $x(0) \in \mathcal{B} := \mathbb{R}^n$ be any initial state. If $V(x(0)) \leq \alpha^2$ then it is straightforward from Lemma A.5 that there exists a control input $u(\cdot)$ such that for all $t \geq 0$, $V(x(t)) \leq \alpha^2$, i.e., $x(t) \in \mathcal{A}$.

Consider the case when $V(x(0)) > \alpha^2$. Choose any $0 < \epsilon < \gamma$ and let $T = \frac{V(x(0)) - \alpha^2}{\gamma - \epsilon} > 0$ which is finite. From Lemma A.6, there exists a control input $u(\cdot)$ such that for all $t \in [0, T]$,

$$V(x(t)) \leq \max\big(V(x(0)) + (\epsilon - \gamma)t, \alpha^2\big).$$

At $t = T$ we have $V(x(T)) \leq \alpha^2$ because $V(x(0)) + (\epsilon - \gamma)T = \alpha^2$. For $t \geq T$, we can again apply Lemma A.5 to obtain a control input that keeps $x(t) \in \mathcal{A}$. Therefore $x(t) \in \mathcal{A}$ for all $t \geq T$. This concludes the proof.

A.1.10. Proof of Lemma 2.6

PROOF We have that for all $x \in \mathbb{R}^n$ and for all $d \in \mathcal{D}$, there exists $u \in \text{co}(\mathcal{U})$ such that $Ax_c + B_0 + Bu + Wd = 0$, hence $(x - x_c)^T M (Ax_c + B_0 + Bu + Wd) = 0$. It follows that $\min_{u \in \text{co}(\mathcal{U})} (x - x_c)^T M (Ax_c + B_0 + Bu + Wd) \leq 0$, $\forall x \in \mathbb{R}^n$ and $\forall d \in \mathcal{D}$, which implies $\sup_{d \in \mathcal{D}} \min_{u \in \text{co}(\mathcal{U})} (x - x_c)^T M (Ax_c + B_0 + Bu + Wd) \leq 0$ $\forall x \in \mathbb{R}^n$. Thus, for all $x \in \mathbb{R}^n$ such that $V(x) = (x - c_c)^T M (x - c_c) \geq \alpha^2$,

$$-2\lambda (x - c_c)^T M (x - c_c) + 2 \sup_{d \in \mathcal{D}} \min_{u \in \text{co}(\mathcal{U})} (x - c_c)^T M (Ax_c + B_0 + Bu + Wd) \leq -2\lambda\alpha^2.$$

Therefore inequality (2.22) holds with $\gamma = 2\lambda\alpha^2 > 0$. Then Theorem 2.9 allows us to conclude. ∎

A.1.11. Proof of Proposition 2.1

PROOF Equation (2.25b) is equivalent to $A_\lambda^T M + M A_\lambda \preceq 0$ where $A_\lambda = A + \lambda \text{I}$. The eigenvalues of A_λ are $(\lambda_i + \lambda)$ where λ_i are the corresponding eigenvalues of A. Because A is Hurwitz, all its eigenvalues have strictly negative real parts, hence there always exists $\lambda > 0$ small enough such that A_λ is also Hurwitz. It then follows from linear system theory (Rugh, 1996) that there exists $M \succeq 0$ satisfying the Lyapunov inequality $A_\lambda^T M + M A_\lambda \preceq 0$. ∎

A.1.12. Proof of Theorem 2.10

PROOF We will show that the condition in Proposition 2.2 is equivalent to the condition in this Theorem. Indeed,

$$\begin{aligned} &\exists x_c \in \text{int}(\texttt{Safe}) \,:\, \forall d \in \mathcal{D}, \exists u \in \text{co}(\mathcal{U}) \,:\, Ax_c + B_0 + Bu + Wd = 0 \\ \Leftrightarrow\; &\exists x_c \in \text{int}(\texttt{Safe}) \,:\, \forall d \in \mathcal{D}, \exists u \in \text{co}(\mathcal{U}) \,:\, x_c + A^{-1}(B_0 + Wd) = -A^{-1}Bu \end{aligned}$$

$$\Leftrightarrow \exists x_c \in \text{int}(\texttt{Safe}) \,:\, \forall d \in \mathcal{D}, x_c + A^{-1}(B_0 + Wd) \in -A^{-1}B\,\text{co}(\mathcal{U})$$

then using the definition of Pontryagin difference

$$\Leftrightarrow \exists x_c \in \text{int}(\texttt{Safe}) \,:\, x_c \in -A^{-1}B\,\text{co}(\mathcal{U}) \ominus A^{-1}(B_0 + W\mathcal{D}) = Q$$

$$\Leftrightarrow \text{int}(\texttt{Safe}) \cap Q \neq \emptyset.$$

The result then follows directly from Proposition 2.2. ∎

A.1.13. Proof of Theorem 2.11

Proof **First part:** Let the disturbances be constant: $d(t) = d^\star$ for all $t \geq 0$. Define $\mathcal{P} := B_0 + B\,\text{co}(\mathcal{U}) + Wd^\star$. It is straightforward to verify that $\mathcal{P}$ and $-A\texttt{Safe}$ are convex and compact. Because they are disjoint ($\mathcal{P} \cap (-A\texttt{Safe}) = \emptyset$), by the separating hyperplane theorem (see Boyd and Vandenberghe, 2006, sec. 2.5), there exists a hyperplane that strictly separates them. Hence, there exist a vector $a \neq 0$ and numbers b and $\epsilon > 0$ such that $a^T x \geq b\ \forall x \in -A\texttt{Safe}$ and $a^T x \leq b - \epsilon\ \forall x \in \mathcal{P}$. Define the differentiable function $g : \mathcal{X} \to \mathbb{R}$ of state as $g(x) := a^T x - \beta$ where $\beta = \inf_{x \in \texttt{Safe}} a^T x$ is finite as $\texttt{Safe}$ is bounded. It is obvious that $g(x) \geq 0$ for all $x \in \texttt{Safe}$. We have that, $\forall x \in \texttt{Safe}, \forall u \in \mathcal{U} \subset \text{co}(\mathcal{U})$,

$$\begin{aligned}\nabla g(x) \cdot f(x,u,d) &= a^T\left(Ax + B_0 + Bu + Wd^\star\right)\\ &= a^T Ax + a^T\left(B_0 + Bu + Wd^\star\right)\\ &\leq -b + b - \epsilon = -\epsilon.\end{aligned}$$

Therefore, by Theorem 2.1 on page 24, the system is not schedulable.

Second part: There exists $d^\star \in \mathcal{D}$ such that $-Wd^\star \notin (A\texttt{Safe} \oplus B\,\text{co}(\mathcal{U})) + B_0$, hence $0 \notin (A\texttt{Safe} \oplus B\,\text{co}(\mathcal{U})) + B_0 + Wd^\star$. Therefore, for all $x \in \texttt{Safe}$ and all $u \in \text{co}(\mathcal{U})$, $B_0 + Bu + Wd^\star \neq -Ax$. It follows that $B_0 + B\,\text{co}(\mathcal{U}) + Wd^\star$ and $-A\texttt{Safe}$ are disjoint. The non-schedulability comes from the first part. ∎

A.1.14. Proof of Theorem 2.13

PROOF Define $\mathcal{P} := -(B_0 + B\,\mathrm{co}(\mathcal{U}) + Wd^\star)$ and $\mathcal{Q} := \{Ax \in \mathbb{R}^p : x \in \mathbb{R}^n, Cx \in \texttt{Safe}\}$. Obviously, because $\mathrm{co}(\mathcal{U})$ is convex and compact, $\mathcal{P}$ is convex and compact. We will show that $\mathcal{Q}$ is convex and closed.

- $\mathcal{Q}$ is convex: for any $z_1, z_2 \in \mathcal{Q}$ and any $\alpha \in [0,1]$ we have

$$\alpha z_1 + (1-\alpha)z_2 = \alpha Ax_1 + (1-\alpha)Ax_2 = A(\alpha x_1 + (1-\alpha)x_2)$$

  for some $x_1, x_2 \in \mathbb{R}^n$ such that $Cx_1 \in \texttt{Safe}$ and $Cx_2 \in \texttt{Safe}$. Since $\texttt{Safe}$ is convex, $\alpha Cx_1 + (1-\alpha)Cx_2 = C(\alpha x_1 + (1-\alpha)x_2) \in \texttt{Safe}$, thus $A(\alpha x_1 + (1-\alpha)x_2) \in \mathcal{Q}$. Therefore $\mathcal{Q}$ is convex.

- $\mathcal{Q}$ is closed: let $\mathcal{R} := \{x \in \mathbb{R}^n : Cx \in \texttt{Safe}\}$ be the pre-image of $\texttt{Safe}$ under linear map $x \mapsto Cx$. We can decompose $\mathcal{R}$ as $\mathcal{R} = \mathcal{R}_0 + \ker(C)$ where $\mathcal{R}_0 := \{x \in \ker(C)^\perp : Cx \in \texttt{Safe}\}$. The notation $\ker(A)$ denotes the kernel (or nullspace) of a matrix A and $S^\perp$ is the orthogonal complement of a subspace S of an inner product space (for definitions of these notions, see e.g., (Strang, 2006)). The set $\mathcal{R}_0$ is compact because $\texttt{Safe}$ is compact. Hence $\mathcal{Q} = A\mathcal{R}_0 + A\ker(C)$, where $A\mathcal{R}_0$ is compact and $A\ker(C)$ is a linear subspace. It follows that $\mathcal{Q}$ is the pre-image of a closed set under projection. Therefore, $\mathcal{Q}$ is closed.

Because $\mathcal{P}$ and $\mathcal{Q}$ are disjoint non-empty convex sets, $\mathcal{P}$ is compact while $\mathcal{Q}$ is closed, by the separating hyperplane theorem (see Boyd and Vandenberghe, 2006, sec. 2.5), there exists a hyperplane that strictly separates them. The rest of this proof is similar to that of Theorem 2.11 (Appendix A.1.13). ∎

A.1.15. Proof of Theorem 2.15

PROOF There exists $d^\star \in \mathcal{D}$ such that $-Wd^\star \notin \mathrm{co}(\{A(u)\texttt{Safe} + Bu + B_0 : u \in \mathcal{U})$, hence $0 \notin \mathrm{co}(\{A(u)\texttt{Safe} + B_0 + Bu + Wd^\star : u \in \mathcal{U})$. Because the convex hull is non-empty and

closed, by the separating hyperplane theorem, there exists a hyperplane that strictly separates this set and the origin 0. Hence, there exist a vector $a \neq 0$ and $\epsilon > 0$ such that $a^T x \leq -\epsilon$ for all $x \in \text{co}(\{A(u)\texttt{Safe} + B_0 + Bu + Wd^\star : u \in \mathcal{U})$. Consider the function $g(x) := a^T x - \beta$ where $\beta = \inf_{x \in \texttt{Safe}} a^T x$ is finite as Safe is bounded. Obviously, $g(x) \geq 0$ for all $x \in \texttt{Safe}$. We have that, for all $x \in \texttt{Safe}$ and all $u \in \mathcal{U}$, $A(u)x + B_0 + Bu + Wd^\star \in \text{co}(\{A(u)\texttt{Safe} + B_0 + Bu + Wd^\star : u \in \mathcal{U})$. Therefore $\nabla g(x) \cdot f(x, u, d) = a^T \left(A(u)x + B_0 + Bu + Wd^\star\right) \leq -\epsilon$. By Theorem 2.1 on page 24, the system is not schedulable. ■

A.2. Proofs of Chapter 4

A.2.1. Proof of Theorem 4.1

PROOF Define function $V(x) := x^T M x = \|x\|_M^2$. The time derivative of $V(x)$ along the flow $x(t)$ of the system is

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} V(x(t)) &= \nabla V(x) \cdot \dot{x}(t) \\
&= 2x(t)^T M \left(Ax(t) + B_0 + Wd(t)\right) \\
&= 2x(t)^T M A x(t) + 2x(t)^T M \left(B_0 + Wd(t)\right) \\
&\leq -2\lambda x(t)^T M x(t) + 2x(t)^T M \left(B_0 + Wd(t)\right)
\end{aligned}$$

in which we use the inequality (4.2b). Equation (4.3) implies that for all $x \in \mathbb{R}^n$ and all $d \in \mathcal{D}$

$$x^T M \left(B_0 + Wd\right) \leq \sup_{d \in \mathcal{D}} x^T M \left(B_0 + Wd\right) = \alpha\lambda\sqrt{x^T M x}.$$

It follows that

$$\dot{V}(t) \leq -2\lambda x(t)^T M x(t) + 2\alpha\lambda\sqrt{x(t)^T M x(t)} = 2\lambda\sqrt{V(t)}\left(\alpha - \sqrt{V(t)}\right)$$

in which we write $V(x(t))$ as a function of time t.

Let $p : \mathbb{R}^+ \to \mathbb{R}^+$ be a differentiable function of time t that satisfies the differential equation $\dot{p}(t) = 2\lambda\sqrt{p(t)}(\alpha - \sqrt{p(t)})$ with initial condition $p(0) = V(0) = x(0)^T M x(0)$. Observe that if at any time $t \geq 0$, $p(t) = V(t)$ then $\dot{p}(t) = 2\lambda\sqrt{V(t)}\left(\alpha - \sqrt{V(t)}\right) \geq \dot{V}(t)$. From the Müller's existence theorem, i.e., the comparison theorem for differential inequalities (Müller, 1926; Walter, 1997; Kieffer et al., 2006), $p(t)$ is an upper bound of $V(t)$, that is $V(t) \leq p(t)$ for all $t \geq 0$. It is straightforward to verify that $p(t) = \alpha^2 e^{-2\lambda t}\left(K + e^{\lambda t}\right)^2$, where K is calculated from the initial condition as $K = \frac{\sqrt{p(0)}}{\alpha} - 1 = \frac{\|x(0)\|_M}{\alpha} - 1$. Therefore, for all $t \geq 0$

$$\|x(t)\|_M = \sqrt{V(t)} \leq \sqrt{p(t)} = \alpha \mathrm{e}^{-\lambda t}\left(K + e^{\lambda t}\right) = (\|x(0)\|_M - \alpha)\mathrm{e}^{-\lambda t} + \alpha. \qquad \blacksquare$$

A.2.2. Comparison of state trajectory bounds in Equations (4.5) and (4.8)

We will show that the bound in Equation (4.5) is as tight as, and usually tighter than, the bound in Equation (4.8).

If $\lambda > 0$ (hence $\alpha \geq 0$), we have that

$$\alpha(1 - \mathrm{e}^{-\lambda t}) \leq \alpha\lambda t, \qquad \forall t \geq 0$$

and

$$\frac{\mu}{\|A\|_M}\left(\mathrm{e}^{\|A\|_M t} - 1\right) \geq \frac{\mu}{\|A\|_M}\|A\|_M t = \mu t = \alpha\lambda t, \qquad \forall t \geq 0$$

where we used the inequalities $1 - \mathrm{e}^{-x} \leq x$ and $\mathrm{e}^x - 1 \geq x$ for all $x \geq 0$. Therefore $\alpha(1 - \mathrm{e}^{-\lambda t}) \leq \frac{\mu}{\|A\|_M}\left(\mathrm{e}^{\|A\|_M t} - 1\right)$ for all $t \geq 0$. Furthermore, the equality occurs if and only if $t = 0$; thus for $t > 0$, the inequality is strict.

Suppose $\lambda < 0$, hence $\alpha \leq 0$. It follows from the proof of Proposition 2.1 (see Appendix A.1.11 on page 186) that any $\lambda < -\max_i \mathrm{Re}\,\lambda_i$ will suffice the conditions of Theorem 4.1, where λ_i are the eigenvalues of A and $\mathrm{Re}\,c$ denotes the real part of a complex number c. On the other hand, Proposition 3.10 in (Dullerud and Paganini, 2000) verifies that $\rho(A) \leq \|A\|_M$ where

$\rho(A) = \max_i |\lambda_i|$ is the spectral radius of A. Thus, for almost all matrices A, we can choose λ such that $-\lambda \leq \|A\|_M$. Consider

$$\frac{\mu}{\|A\|_M}\left(\mathrm{e}^{\|A\|_M t} - 1\right) - \alpha(1 - \mathrm{e}^{-\lambda t}) = \alpha\left(\frac{\lambda}{\|A\|_M}\left(\mathrm{e}^{\|A\|_M t} - 1\right) - \left(1 - \mathrm{e}^{-\lambda t}\right)\right).$$

Its derivative with respect to t is

$$\alpha\lambda\left(\mathrm{e}^{\|A\|_M t} - \mathrm{e}^{-\lambda t}\right) \geq 0, \qquad \forall t \geq 0.$$

Therefore,

$$\frac{\mu}{\|A\|_M}\left(\mathrm{e}^{\|A\|_M t} - 1\right) - \alpha(1 - \mathrm{e}^{-\lambda t}) \geq 0, \qquad \forall t \geq 0.$$

A.2.3. Proof of Theorem 4.2

PROOF Since $\lambda > 0$, Equation (4.15c) implies that

$$\min_{u\in\mathcal{U}} \max_{d\in\mathcal{D}} z^T M\left(Ax_c + B_0 + Bu + Wd\right) \leq \lambda(\alpha - \epsilon)\sqrt{z^T M z}, \qquad \forall z \in \mathbb{R}^n$$

for some $\epsilon > 0$. Together with Equations (4.15a) and (4.15b), we have that for all $x \in \mathbb{R}^n$ such that $V(x) = (x - x_c)^T M (x - x_c) \geq \alpha^2$,

$$\begin{aligned}
&\inf_{u\in\mathcal{U}} \sup_{d\in\mathcal{D}} \nabla V(x) \cdot f(x, u, d) \\
&= 2(x - x_c)^T MA(x - x_c) + 2\min_{u\in\mathcal{U}} \max_{d\in\mathcal{D}} (x - x_c)^T M\left(Ax_c + B_0 + Bu + Wd\right) \\
&\leq -2\lambda(x - x_c)^T M(x - x_c) + 2\min_{u\in\mathcal{U}} \max_{d\in\mathcal{D}} (x - x_c)^T M\left(Ax_c + B_0 + Bu + Wd\right) \\
&\leq -2\lambda(x - x_c)^T M(x - x_c) + 2\lambda(\alpha - \epsilon)\sqrt{(x - x_c)^T M(x - x_c)} \\
&= 2\lambda\sqrt{(x - x_c)^T M(x - x_c)}\left(\alpha - \epsilon - \sqrt{(x - x_c)^T M(x - x_c)}\right) \\
&\leq -2\lambda\alpha\epsilon.
\end{aligned}$$

Hence Equation (2.21) holds with $\gamma = 2\lambda\alpha\epsilon$. It follows from Theorem 2.9 that $\mathcal{A}$ is an attracting set of control system (4.14). Furthermore, a state feedback control law $u = \kappa(x)$ associated with $\mathcal{A}$ must satisfy

$$\max_{d\in\mathcal{D}} \nabla V(x) \cdot f(x, \kappa(x), d) \leq -\gamma$$

for all $x \in \mathcal{X}$ such that $V(x) \geq \alpha^2$. Obviously the control law

$$\begin{aligned}\kappa(x) &= \arg\min_{u\in\mathcal{U}} \max_{d\in\mathcal{D}} (x - x_c)^T M \left(Ax_c + B_0 + Bu + Wd\right) \\ &= \arg\min_{u\in\mathcal{U}} (x - x_c)^T MBu\end{aligned}$$

for all $x \in \mathcal{X}$ satisfies that condition. ∎

A.2.4. Proof of Proposition 4.1

PROOF The existence of M and $\lambda > 0$ is verified by Proposition 2.1 on page 53. Now we only need to show the existence of finite $\alpha > 0$ that satisfies Equation (4.15c). The right hand side of this inequality reads

$$\begin{aligned}&\max_{z^TMz=1} \left(\min_{u\in\mathcal{U}} \max_{d\in\mathcal{D}} z^T M \left(Ax_c + B_0 + Bu + Wd\right) \right) \\ \leq &\max_{z^TMz=1} z^T M \left(Ax_c + B_0\right) + \max_{u\in\mathcal{U}} \max_{d\in\mathcal{D}} \max_{z^TMz=1} z^T M \left(Bu + Wd\right) \\ \leq &\sqrt{\left(Ax_c + B_0\right)^T M \left(Ax_c + B_0\right)} + \max_{u\in\mathcal{U}} \max_{d\in\mathcal{D}} \sqrt{\left(Bu + Wd\right)^T M \left(Bu + Wd\right)}.\end{aligned}$$

Because $\mathcal{U}$ and $\mathcal{D}$ are compact sets, the upper-bound above is finite. Therefore, there exists finite $\alpha > 0$ such that Equation (4.15c) holds. ∎

NOMENCLATURE

Systems

$\mathcal{R}_t^{-1}(X)$	Robust backward reachability operator at time step t from set X
$\mathcal{R}_{[t,t']}^{-1}(X)$	Robust backward reachability operator for interval $[t, t']$ of time steps from set X, $t' \geq t$
δ	The time period of periodic schedules, page 27
$\mathcal{X}$	The state space, i.e., the set of all valid values of state x, page 12
η	The utilization vector of periodic control signal $u(\cdot)$, page 27
η_i	The utilization of periodic control input i, page 27

Functions and Signals

$\|s(\cdot)\|_\infty$	$\mathcal{L}_\infty$-norm of the signal $s(\cdot)$; $\|s(\cdot)\|_\infty = \sup_t \|s(t)\|$
$\mathfrak{F}(X, Y)$	The set of all measurable functions from set X to set Y, page 13
∇f	Gradient of function $f : \mathbb{R}^n \to \mathbb{R}$
Df	Derivative (Jacobian) matrix of function $f : \mathbb{R}^n \to \mathbb{R}^m$

Vectors and Matrices

e_i	The i^{th} standard basis vector, with 1 in the i^{th} position and 0 everywhere else
I	Identity matrix
$\mathbf{1}_n$	Vector in $\mathbb{R}^n$ with all components one (dimension omitted if clear from context)

$\mathbf{0}_n$	Vector in $\mathbb{R}^n$ with all components zero (dimension omitted if clear from context)
$\|\cdot\|$	A vector norm or an induced matrix norm
$\|x\|_1$	ℓ_1-norm of vector x
$\|x\|_2$	Euclidean (or ℓ_2-) norm of vector x
$\|x\|_\infty$	ℓ_∞-norm (or maximum norm) of vector x
$\|x\|_M$	M-norm of vector x, $\|x\|_M := \sqrt{x^T M x}$, where $M \succeq 0$
$\|A\|_2$	Induced ℓ_2-norm (or spectral norm) of matrix A
$\|A\|_{\max}$	Maximum norm of matrix A, i.e., $\max_{i,j} \|A_{ij}\|$
$\|A\|_M$	Induced M-norm of matrix A, i.e., $\sup_x \frac{\|Ax\|_M}{\|x\|_M}$, where $M \succeq 0$
$\mathrm{diag}(x)$	Diagonal matrix with diagonal entries in vector x
$\mathrm{diag}(x_1, x_2, \ldots)$	Diagonal matrix with diagonal entries $x_1, x_2, \ldots$
$X \succ Y$	Strict matrix inequality between symmetric matrices X and Y
$X \succeq Y$	Matrix inequality between symmetric matrices X and Y
$x \succ y$	Strict component-wise inequality between vectors x and y
$x \succeq y$	Component-wise inequality between vectors x and y

Sets

$\|X\|$	Cardinality of set X
$\mathfrak{B}(c, r)$	The ball with center c and radius r, page 37
$\mathfrak{B}_d(c, r)$	The ball in metric d with center c and radius r

$\mathrm{co}(X)$	The convex hull of set X
$\mathrm{dist}(x, S)$	The distance between a point x and a set S
$\mathrm{int}(X)$	The interior of set X
$\ker(A)$	The kernel (or nullspace) of a matrix A, page 188
$S^{\perp}$	The orthogonal complement of a subspace S of an inner product space, page 188
$X \ominus Y$	Pontryagin difference of two sets X, Y
$X \oplus Y$	Minkowski sum of two sets X, Y
$\mathbb{N}$	The set of natural numbers, i.e., $\mathbb{N} = \{0, 1, 2, \dots\}$
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^+$	The set of non-negative real numbers, i.e., $\mathbb{R}^+ = \{x \in \mathbb{R} : x \geq 0\}$

Other Symbols

$\lceil c \rceil$	The smallest integer not less than c
$\lfloor c \rfloor$	The largest integer not exceeding c
$\mathrm{Re}\, c$	The real part of a complex number c, page 190
$f \circ g$	Composition of two functions/operators: $(f \circ g)(x) = f(g(x))$

BIBLIOGRAPHY

M. H. Albadi and E. F. El-Saadany. Demand response in electricity markets: An overview. In *Proc. IEEE Power Engineering Society General Meeting*, pages 1–5, 2007.

R. Alur. Formal verification of hybrid systems. In *11th International Conference on Embedded Software*, 2011.

R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

R. Alur, T. A. Henzinger, and H. Wong-Toi. Symbolic analysis of hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 702–707, 1997.

R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

R. Alur, A. Trivedi, and D. Wojtczak. Optimal scheduling for constant-rate multi-mode systems. In *Proceedings of the ACM international conference on Hybrid Systems: Computation and Control*, pages 75–84. ACM, 2012.

A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, 2010.

Z. Artstein. Stabilization with relaxed controls. *Nonlinear Analysis: Theory, Methods & Applications*, 7(11):1163–1173, 1983.

E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE*, 88(7):1011–1025, 2000.

E. Asarin, T. Dang, and A. Girard. Reachability analysis of nonlinear systems using conservative approximation. In *Hybrid Systems: Computation and Control*, volume 2623, pages 20–35. Springer Berlin/Heidelberg, 2003.

ASHRAE. *ANSI/ASHRAE Standard 55-2004, Thermal Comfort Conditions for Human Occupancy*. American Society of Heating, Air-Conditioning, and Refrigeration Engineers, 2004.

ASHRAE. *HVAC Systems and Equipment*. ASHRAE Handbook. ASHRAE, Atlanta, GA, 2008.

ASHRAE. *Fundamentals*. ASHRAE Handbook. ASHRAE, Atlanta, GA, 2009.

K. Åström and B. Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. In *Proceedings of the 14th IFAC World congress*, volume 11, pages 301–306, 1999.

A. Beghi, L. Cecchinato, and M. Rampazzo. Thermal and comfort control for radiant heating/cooling systems. In *Proceedings of IEEE International Conference on Control Applications*, pages 258–263, 2011.

D. Bertsekas. Infinite time reachability of state-space regions by using feedback control. *IEEE Transactions on Automatic Control*, 17(5):604–613, 1972.

K. Bhattacharyya and M. L. Crow. A fuzzy logic based approach to direct load control. *IEEE Transactions on Power Systems*, 11(2):708–714, 1996.

F. Blanchin and W. Ukovich. Linear programming approach to the control of discrete-time periodic systems with uncertain inputs. *Journal of Optimization Theory and Applications*, 78:523–539, 1993.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2006.

S. Braithwait and K. Eakin. The role of demand response in electric power market design. *Edison Electric Institute*, 2002.

BUGSENG. The Parma Polyhedra Library (PPL), 2012. URL `http://bugseng.com/products/ppl`.

G. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Real-Time Systems Series. Springer, 2011.

A. Camacho, P. Martí, M. Velasco, C. Lozoya, R. Villa, J. Fuertes, and E. Griful. Self-triggered networked control systems: An experimental case study. In *IEEE International Conference on Industrial Technology*, pages 123–128. IEEE, 2010.

J. Chen, F. N. Lee, A. M. Breipohl, and R. Adapa. Scheduling direct load control to minimize system operation cost. *IEEE Transactions on Power Systems*, 10(4):1994–2001, 1995.

T. Y. Chen. Application of adaptive predictive control to a floor heating system with a large thermal lag. *Energy and Buildings*, 34(1):45–51, 2002.

S. Cho and M. Zaheer-Uddin. An experimental study of multiple parameter switching control for radiant floor heating systems. *Energy*, 24(5):433–444, 1999.

C.-M. Chu and T.-L. Jong. A novel direct air-conditioning load control method. *IEEE Transactions on Power Systems*, 23(3):1356–1363, 2008.

C.-M. Chu, T.-L. Jong, and Y.-W. Huang. A direct load control of air-conditioning loads with thermal comfort control. In *Proceedings of the IEEE Power Engineering Society General Meeting*, pages 664–669, 2005.

A. Chutinan and B. H. Krogh. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In *Hybrid Systems: Computation and Control*, pages 76–90. Springer, 1999.

A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control*, 48(1):64–75, 1 2003.

A. I. Cohen and C. C. Wang. An optimization method for load management scheduling. *IEEE Transactions on Power Systems*, 3(2):612–618, 1988.

R. I. Davis and A. Burn. A survey of hard real-time scheduling algorithms and schedulability analysis techniques for multiprocessor systems. Technical report, University of York, 2009.

E. De Santis, M. D. Di Benedetto, and L. Berardi. Computation of maximal safe sets for switching systems. *IEEE Transactions on Automatic Control*, 49(2):184–195, 2004.

D. Dimarogonas, E. Frazzoli, and K. Johansson. Distributed self-triggered control for multi-agent systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6716–6721. IEEE, 2010.

I. Doebber, M. Moore, and M. Deru. Radiant slab cooling for retail. *ASHRAE Journal*, 52 (6):28, 2010.

R. Dorf and R. Bishop. *Modern Control Systems*. Pearson Prentice Hall, 2008.

G. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach*. Texts in Applied Mathematics. Springer, 2000.

C. Edwards and S. Spurgeon. *Sliding Mode Control: Theory And Applications*. Taylor & Francis systems and control book series. Taylor & Francis, 1998.

T. Facchinetti and M. D. Vedova. Real-time modeling for direct load control in cyber-physical power systems. *IEEE Transactions on Industrial Informatics*, 7(4):689–698, 2011.

T. Facchinetti, E. Bibi, and M. Bertogna. Reducing the peak power through real-time scheduling techniques in cyber-physical energy systems. In *First International Workshop on Energy Aware Design and Analysis of Cyber Physical Systems*, 2010.

G. E. Fainekos, A. Girard, and G. J. Pappas. Temporal logic verification using simulation. In *Proceedings of FORMATS*, volume 4202 of *LNCS*, pages 171–186. Springer, 2006.

G. Franklin, J. Powell, and M. Workman. *Digital control of dynamic systems*. Addison-Wesley, 1998.

R. A. Freeman and P. V. Kokotovic. Inverse optimality in robust stabilization. *SIAM Journal on Control and Optimization*, 34(4):1365–1391, 1996.

A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, volume 3414, pages 291–305, Zurich, Switzerland, 2005.

A. Girard. Controller synthesis for safety and reachability via approximate bisimulation. *Automatica*, 48(5):947–953, 2012.

A. Girard and G. Pappas. Approximate bisimulation: A bridge between computer science and control theory. *European Journal of Control*, 17(5):568, 2011.

A. Girard and G. J. Pappas. Approximate bisimulations for nonlinear dynamical systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 684–689, 2005.

A. Girard and G. J. Pappas. Verification using simulation. In *Hybrid Systems: Computation and Control*, pages 272–286. Springer, 2006.

A. Girard and G. J. Pappas. Approximate bisimulation relations for constrained linear systems. *Automatica*, 43(8):1307–1317, 2007a.

A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5):782–798, 2007b.

R. Gondhalekar. Performance analysis of asynchronous model predictive control laws. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1233–1238, 2011.

R. Gondhalekar and C. N. Jones. MPC of constrained discrete-time linear periodic systems – a framework for asynchronous control: Strong feasibility, stability and optimality via periodic invariance. *Automatica*, 47(2):326–333, 2011.

R. Gondhalekar, F. Oldewurtel, and C. N. Jones. Least-restrictive robust MPC of periodic affine systems with application to building climate control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5257–5263, 2010.

L. Grüne. *Asymptotic Behavior of Dynamical and Control Systems under Perturbation and Discretization*. Number 1783 in Lecture Notes in Mathematics. Springer, 2002.

C. L. Guernic and A. Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250–262, 2010.

M. Gwerder, B. Lehmann, J. Tödtli, V. Dorer, and F. Renggli. Control of thermally activated building systems (TABS). *Applied Energy*, 85:565–581, 2008.

M. Gwerder, J. Tödtli, B. Lehmann, V. Dorer, W. Güntensperger, and F. Renggli. Control of thermally activated building systems (TABS) in intermittent operation with pulse width modulation. *Applied Energy*, 86(9):1606–1616, 2009.

W. Heemels, J. Sandee, and P. Van Den Bosch. Analysis of event-driven controllers for linear systems. *International journal of control*, 81(4):571–590, 2008.

T. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th IEEE Symposium on Logic in Computer Science*, pages 278–292, 1996.

R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1990.

Y.-Y. Hsu and C.-C. Su. Dispatch of direct load control using dynamic programming. *IEEE Transactions on Power Systems*, 6(3):1056–1061, 1991.

J. Hubbard and B. West. *Differential Equations: A Dynamical Systems Approach. Part II: Higher Dimensional Systems.* Texts in Applied Mathematics. Springer, 1995.

W. Hurewicz. *Lectures on Ordinary Differential Equations.* Dover Publications, 2002.

S. Jha, S. Gulwani, S. Seshia, and A. Tiwari. Synthesizing switching logic for safety and dwell-time requirements. In *First International Conference on Cyber-physical Systems*, pages 22–31, 2010.

S. Jha, S. A. Seshia, and A. Tiwari. Synthesis of optimal switching logic for hybrid systems. In *EMSOFT*, pages 107–116, 2011.

A. A. Julius and G. J. Pappas. Trajectory based verification using local finite-time invariance. In *Hybrid Systems: Computation and Control*, pages 223–236. Springer, 2009.

A. A. Julius, G. E. Fainekos, M. Anand, I. Lee, and G. J. Pappas. Robust test generation and coverage for hybrid systems. In *Hybrid Systems: Computation and Control*, volume 4416 of *LNCS*, pages 329–342. Springer, 2007.

E. C. Kerrigan. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control.* PhD thesis, University of Cambridge, 2000.

H. K. Khalil. *Nonlinear systems.* Macmillan New York, 1992.

M. Kieffer, E. Walter, and I. Simeonov. Guaranteed nonlinear parameter estimation for continuous-time dynamical models. In *Proceedings of IFAC Symposium on System Identification*, pages 843–848, 2006.

S. Koch, J. Mathieu, and D. Callaway. Modeling and control of aggregated heterogeneous thermostatically controlled loads for ancillary services. In *Proceedings of the Power Systems Computation Conference*, pages 1–7, 2011.

M. Koschenz and V. Dorer. Interaction of an air system with concrete core conditioning. *Energy and Buildings*, 30(2):139–145, 1999.

E. Kreyszig. *Introductory functional analysis with applications.* Wiley, 1989.

S. Kundu, N. Sinitsyn, S. Backhaus, and I. Hiskens. Modeling and control of thermostatically controlled loads. In *Proceedings of the Power Systems Computation Conference*, 2011.

A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 52(1):26–38, 1 2007.

M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004. URL `http://control.ee.ethz.ch/~mpt/`.

J. Le Ny, E. Feron, and G. J. Pappas. Resource constrained LQR control under fast sampling. In *Hybrid Systems: Computation and Control*, pages 271–280, 2011.

J.-Y. Lee, M.-S. Yeo, and K.-W. Kim. Predictive control of the radiant floor heating system in apartment buildings. *Journal of Asian Architecture and Building Engineering*, 1(1): 105–112, 2002.

S. C. Lee, S. J. Kim, and S. H. Kim. Demand side management with air conditioner loads based on the queuing system model. *IEEE Transactions on Power Systems*, 26(2):661–668, 2011.

T.-F. Lee, H.-Y. Wu, Y.-C. Hsiao, P.-J. Chao, F.-M. Fang, and M.-Y. Cho. Relaxed dynamic programming for constrained economic direct loads control scheduling. In *Proceedings of International Conference on Intelligent Systems Applications to Power Systems*, pages 1–6, 2007.

B. Lehmann, V. Dorer, and M. Koschenz. Application range of thermally activated building systems (TABS). *Energy and Buildings*, 39(5):593–598, 2007.

B. Lehmann, V. Dorer, M. Gwerder, F. Renggli, and J. Tödtli. Thermally activated building systems (TABS): Energy efficiency as a function of control strategy, hydronic circuit topology and (cold) generation system. *Applied Energy*, 88(1):180–191, 2011.

M. Lemmon, T. Chantem, X. S. Hu, and M. Zyskowski. On self-triggered full-information H-infinity controllers. In *Hybrid Systems: Computation and Control*, pages 371–384. Springer-Verlag, 2007.

Z. Li, P.-C. Huang, A. K. Mok, T. X. Nghiem, M. Behl, G. J. Pappas, and R. Mangharam. On the feasibility of linear discrete-time systems of the green scheduling problem. In *Proceedings of the 32nd IEEE Real-Time Systems Symposium*, pages 295–304, 2011.

D. Liberzon. *Switching in Systems and Control*. Birkhäuser, 2003.

D. Liberzon, E. D. Sontag, and Y. Wang. Universal construction of feedback laws achieving ISS and integral-ISS disturbance attenuation. *Systems & Control Letters*, 46(2):111–127, 2002.

H. Lin. *Robust analysis and synthesis of uncertain linear hybrid systems with networked control applications*. PhD thesis, University of Notre Dame, 2005.

H. Lin and P. J. Antsaklis. Stability and stabilizability of switched linear systems: A survey of recent results. *IEEE Transactions on Automatic Control*, 54(2):308–322, 2009.

J. W. S. Liu. *Real-time Systems*. Prentice Hall, 2000.

A. Lodi, S. Martello, and M. Monaci. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252, 2002.

J. Löfberg. Pre- and post-processing sum-of-squares programs in practice. *IEEE Transactions on Automatic Control*, 54(5):1007–1011, 2009.

J. Löfberg. Automatic robust convex programming. *Optimization methods and software*, 27 (1):115–129, 2012.

J. Lunze and D. Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211–215, 2010.

J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35:349–370, 1999.

Y. Ma, F. Borrelli, B. Hencey, A. Packard, and S. Bortoff. Model predictive control of thermal energy storage in building cooling systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 392 –397, 12 2009.

Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves. Model predictive control for the operation of building cooling systems. *IEEE Transactions on Control Systems Technology*, 20(3):796–803, 2012.

O. Maler. Computing reachable sets: An introduction. Technical report, VERIMAG, 2008.

J. Mathieu and D. Callaway. State estimation and control of heterogeneous thermostatically controlled loads for load following. In *Hawaii International Conference on System Science (HICSS)*, pages 2002–2011, 2012.

M. Mazo, A. Anta, and P. Tabuada. On self-triggered control for linear systems: Guarantees and complexity. In *European Control Conference*, 2009.

F. C. McQuiston, J. D. Parker, and J. D. Spitler. *Heating, Ventilating and Air Conditioning Analysis and Design*. Wiley, 6 edition, 2005.

I. Mitchell and C. Tomlin. Level set methods for computation in hybrid systems. In *Hybrid Systems: Computation and Control*, pages 310–323. Springer, 2000.

I. Mitchell, A. Bayen, and C. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 7 2005.

I. M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In *Hybrid systems: computation and control*, pages 428–443. Springer, 2007.

A. Molina-Garcia, M. Kessler, J. Fuentes, and E. Gomez-Lazaro. Probabilistic characterization of thermostatically controlled loads to model the impact of demand response programs. *IEEE Transactions on Power Systems*, 26(1):241–251, 2011.

N. Motegi, M. Piette, D. Watson, S. Kiliccote, and P. Xu. Introduction to commercial

building control strategies and techniques for demand response. Technical Report 59975, Lawrence Berkeley National Laboratory, 2007.

M. Müller. Über das fundamentaltheorem in der theorie der gewöhnlichen differentialgleichungen. *Mathematische Zeitschrift*, 26:619–645, 1926.

T. X. Nghiem. MLE+: a Matlab-EnergyPlus co-simulation interface. `http://www.seas.upenn.edu/~nghiem/mleplus.html`, 2011.

T. X. Nghiem, M. Behl, R. Mangharam, and G. J. Pappas. Green scheduling of control systems for peak demand reduction. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5131–5136, 2011a.

T. X. Nghiem, M. Behl, G. J. Pappas, and R. Mangharam. Green scheduling: Scheduling of control systems for peak power reduction. In *Proceedings of International Green Computing Conference and Workshops (IGCC)*, pages 1–8, 2011b.

T. X. Nghiem, M. Behl, R. Mangharam, and G. J. Pappas. Scalable scheduling of building control systems for peak demand reduction. In *Proceedings of the American Control Conference*, 2012a.

T. X. Nghiem, M. Behl, G. J. Pappas, and R. Mangharam. Green scheduling for radiant systems in buildings. In *IEEE Conference on Decision and Control*, 2012b.

C. Nowzari and J. Cortés. Self-triggered coordination of robotic networks for optimal deployment. In *Proceedings of American Control Conference*, pages 1039–1044, 2011.

B. Olesen. Radiant floor heating in theory and practice. *ASHRAE Journal*, 44(7):19–24, 2002.

P. Otanez, J. Moyne, and D. Tilbury. Using deadbands to reduce communication in networked control systems. In *Proceedings of the American Control Conference*, volume 4, pages 3015–3020, 2002.

P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

R. Postoyan, P. Tabuada, D. Nesic, and A. Anta. Event-triggered and self-triggered stabilization of distributed networked control systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2565–2570, 2011.

S. Prajna. *Optimization-based methods for nonlinear and hybrid systems verification*. PhD thesis, California Institute of Technology, 2005.

S. Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42(1):117–126, 2006.

S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, pages 477–492. Springer, 2004.

S. Prajna, A. Jadbabaie, and G. J. Pappas. Stochastic safety verification using barrier certificates. In *Proceedings of the IEEE Conference on Decision and Control*, volume 1, pages 929–934, 2004.

S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. SOSTOOLS and its control applications. *Positive Polynomials in Control*, 1:273–292, 2005.

S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8): 1415–1428, 2007.

B. Ramanathan and V. Vittal. A framework for evaluation of advanced direct load control with minimum disruption. *IEEE Transactions on Power Systems*, 23(4):1681–1688, 2008.

D. G. Roberson and D. J. Stilwell. $\mathcal{L}_2$ gain of periodic linear switched systems: Fast switching behavior. *IEEE Transactions on Automatic Control*, 54(7):1632–1637, 2009.

W. J. Rugh. *Linear System Theory.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, second edition, 1996.

D. Saelens, W. Parys, and R. Baetens. Energy and comfort performance of thermally activated building systems including occupant behavior. *Building and Environment*, 46(4): 835–848, 2011.

F. Sakellariou. Model predictive control for thermally activated building systems. Master's thesis, Eindhoven University of Technology, 2011.

J. Seem. *Modeling of heat transfer in buildings.* University of Wisconsin–Madison, 1987.

A. Seuret and C. Prieur. Event-triggered sampling algorithms based on a lyapunov functional. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6128–6133, 2011.

J. Siegenthaler. *Modern Hydronic Heating: For Residential and Light Commercial Buildings.* Cengage Learning, 2011.

C. Stetiu. Energy and peak power savings potential of radiant cooling systems in US commercial buildings. *Energy and Buildings*, 30(2):127–138, 1999.

G. Strang. *Linear Algebra and Its Applications.* Thomson Brooks/Cole, 2006.

A. Subramanian, M. Garcia, A. Dominguez-Garcia, D. Callaway, K. Poolla, and P. P. Varaiya. Real-time scheduling of deferrable electric loads. In *Proceedings of the American Control Conference*, pages 3643–3650, 2012.

Z. Sun and S. S. Ge. *Switched Linear Systems: Control and Design.* Springer, 2005.

P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

The MathWorks. System identification toolbox, 2009.

J. Tokarzewski. Stability of periodically switched linear systems and the switching frequency. *International Journal of Systems Science*, 18(4):697–726, 1987.

TRFund Study. Understanding PECO's general service tariff, 2007.

U.S. Department of Energy. The smart grid: an introduction, 2008.

M. D. Vedova, M. Ruggeri, and T. Facchinetti. On real-time physical systems. In *Proceedings of International Conference on Real-Time and Network Systems*, pages 41–49, 11 2010.

M. Velasco, J. Fuertes, and P. Marti. The self triggered task model for real-time control systems. In *Work-in-Progress Session of the 24th IEEE Real-Time Systems Symposium*, volume 384, 2003.

A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

W. Walter. Differential inequalities and maximum principles: theory, new methods and applications. *Nonlinear Analysis: Theory, Methods & Applications*, 30(8):4695–4711, 1997.

R. Watson and K. Chapman. *Radiant Heating and Cooling Handbook*. McGraw-Hill Handbooks. McGraw-Hill, 2002.

D.-C. Wei and N. Chen. Air conditioner direct load control by multi-pass dynamic programming. *IEEE Transactions on Power Systems*, 10(1):307–313, 1995.

H.-T. Yang and K.-Y. Huang. Direct load control using fuzzy dynamic programming. *IEE Proceedings of Generation, Transmission and Distribution*, 146(3):294–300, 1999.

L. Yao, W.-C. Chang, and R.-L. Yen. An iterative deepening genetic algorithm for scheduling of direct load control. *IEEE Transactions on Power Systems*, 20(3):1414–1421, 2005.

M. M. Zavlanos, A. A. Julius, S. P. Boyd, and G. J. Pappas. Inferring stable genetic networks from steady-state data. *Automatica*, 47(6):1113–1122, 2011.

B. Zhou, W. X. Zheng, and G.-R. Duan. Stability and stabilization of discrete-time periodic linear systems with actuator saturation. *Automatica*, 47(8):1813–1820, 2011.