Optimization and Model-Based Control for Max-Plus Linear and Continuous Piecewise Affine Systems

Jia Xu

Optimization and Model-Based Control for Max-Plus Linear and Continuous Piecewise Affine Systems

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft
op gezag van de Rector Magnificus prof.dr.ir. T.H.J.J. van der Hagen
voorzitter van het College voor Promoties
in het openbaar te verdedigen op
donderdag 6 juni 2019 om 10.00 uur

door

Jia XU

Bachelor of Science in Statistics, Shandong University at Weihai, China
geboren te Dezhou, Shandong, China

This dissertation has been approved by the promotors:

Promotor: Prof.dr.ir. B. De Schutter
Copromotor: Dr.ir. A.J.J. van den Boom

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof.dr.ir. B. De Schutter	Technische Universiteit Delft, promotor
Dr.ir. A.J.J. van den Boom	Technische Universiteit Delft, copromotor

Independent members:

Dr. J. Komenda	Czech Academy of Sciences
Prof.dr. I.L. Buşoniu	Universitatea Tehnica din Cluj-Napoca
Prof.dr. B.F. Heidergott	Vrije Universiteit Amsterdam
Prof.dr.ir. C. Vuik	Technische Universiteit Delft
Prof.dr.ir. A. van Keulen	Technische Universiteit Delft

The support of the China Scholarship Council (CSC) is gratefully acknowledged, as well as the support of the Delft Center for Systems and Control (DCSC), and the European Union Seventh Framework Network of Excellence “Highly-complex and networked control systems (HYCON2)”. This dissertation has been completed in partial fulfillment of the requirements of the Dutch Institute of Systems and Control (DISC) for graduate studies.

Published and distributed by: Jia Xu
E-mail: xujiajane@gmail.com

ISBN 978-94-6186-953-1

Keywords: discrete-event systems, max-plus linear systems, hybrid systems, continuous piecewise affine systems, max-min-plus-scaling systems, model predictive control, stochastic max-plus linear systems, optimistic optimization, optimistic planning, receding-horizon control

Printed in the Netherlands

Preface

This thesis is a result of my work at the Delft Center for Systems and Control (DCSC) of Delft University of Technology. At the end of my PhD journey, I want to express my sincere thanks to the people who helped and accompanied me.

First and foremost, I would like to thank my promotor and daily supervisor Bart De Schutter without whom this thesis would not have been possible. I have been really grateful for his guidance, patience, and encouragement. He can always give a quick and accurate response to any of my questions. His kindness and understanding are valuable to me in the tough times of my PhD study. It is my great honor to have Bart as my supervisor.

I would also like to thank my second promotor Ton van den Boom for his support and guidance. I greatly appreciate his suggestions and insights on my research. I also want to thank him for his comments that improve the quality of my papers and presentations.

Next, I would like to thank Lucian Buşoniu at the Technical University of Cluj-Napoca. I am grateful to him for all the discussions on research, for the detailed explanation of his work, and for the good memories at conferences. It was a pleasure for me to work with him. I also want to thank Zsófia Lendek for the nice dinner and great weekend trips in Romania.

Furthermore, I would like to thank the other members of the PhD committee, Dr. Jan Komenda, Prof.dr. Bernd Heidergott, Prof.dr.ir. Kees Vuik, and Prof.dr.ir. Fred van Keulen, for their valuable time and useful comments on the manuscript of this thesis.

I am also grateful to all my friends and colleagues at DCSC. I would like to express my gratitude to Anahita, Anna, Anqi, Bart Karsbergen, Chengpu, Dieky, Edwin, Elisabeth, Fan, Farid, Hai, Hildo, HuiZhen, Juan, Jun, Kim, Laura, Le, Maolong, Marco, Max, Mohammad, Noortje, Patricio, Pieter, Reinier, Renshi, Sachin, Sadegh, Samira, Shu, Shukai, Shuai Liu, Shuai Yuan, Subramanya, Vahab, Yashar, Yihui, Yiming, Yu Hu, Zhao, Zhe, Zhou. Many thanks to Will van Geest for helping me with developing the C codes. I would like to thank Hans Hellendoorn, Robert Babuška, Kitty, Esther, Heleen, Marieke, Kiran, and Saskia for being very kind and helpful.

I would like to thank my friends Qingqing, Zi, Xiangrong, Bo, Xiaoyan, Huarong, Zhiyong, Jianbin, Ran, Jingyi, Shijie, Jialun, Likun, Tao, Yu Sun, Changgong, and Zongwei for making my life in Delft more colorful.

I would like to extend my gratitude to Prof. Qidi Wu, Prof. Li Li, Prof. Fei Qiao, Prof. Weisheng Xu, and Prof. Jitao Sun in Tongji University, for their help and support.

Finally, my deepest gratitude goes to my family. I would like to thank my parents for their unconditional love and support through my whole life. I thank my husband Yan for always being there for me. Thank you for your listening, understanding, and encouragement. Last but not least, my sweet daughter, thanks for your presence in our life.

Jia Xu, Shanghai, May 2019

Glossary

The symbols and acronyms that occur frequently in this thesis are listed as follows.

List of Symbols

Sets

$\mathbb{R}$	set of real numbers
$\mathbb{R}_{\geq 0}$	set of nonnegative real numbers
$\mathbb{Z}$	set of integers
$\mathbb{Z}_{\geq 0}$	set of nonnegative integers
$[a, b]$	closed interval in $\mathbb{R}$: $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$
(a, b)	open interval in $\mathbb{R}$: $(a, b) = \{x \in \mathbb{R} \mid a < x < b\}$

Functions

$f : D \to T$	function with domain of definition D and target T
$O(f)$	any real function g such that $\limsup_{x\to\infty} \frac{\lvert g(x)\rvert}{f(x)}$ is finite

Matrices, Vectors, and Norms

$\mathbb{R}^{m\times n}$	set of the m by n matrices with real entries
$\mathbb{R}^n$	set of the real column vectors with n components: $\mathbb{R}^n = \mathbb{R}^{n\times 1}$
I_n	n by n identity matrix
A^T	transpose of the matrix A
A_{ij}, $[A]_{ij}$	entry of the matrix A on the i-th row and the j-th column
$A_{i,\cdot}$	i-th row of the matrix A
$A_{\cdot,j}$	j-th column of the matrix A
$\lVert A\rVert_p$	p-norm of the matrix A ($p \geq 1$)
$\lVert A\rVert_\infty$	∞-norm of the matrix A
x_i	i-th component of the vector x
$\lVert x\rVert_p$	p-norm of the vector x ($p \geq 1$)
$\lVert x\rVert_\infty$	∞-norm of the vector x

Model Predictive Control

N_p	prediction horizon length
N_c	control horizon length

Max-Plus Algebra

$\oplus$	max-algebraic addition
$\otimes$	max-algebraic multiplication
ε	$-\infty$
$\mathbb{R}_\varepsilon$	$\mathbb{R} \cup \{-\infty\}$

We use $\square$ to indicate the end of a proof or a remark.

Acronyms

MPC	Model Predictive Control
DES	Discrete-Event System
MPL	Max-Plus Linear
SMPL	Stochastic Max-Plus Linear
PWA	Piecewise Affine
DOO	Deterministic Optimistic Optimization
OPD	Optimistic Planning of Deterministic

Contents

Chapter 1

Introduction

1.1 Motivation of the research

Techniques to model, analyze and control man-made complex systems such as flexible manufacturing systems, timetable dependent transportation networks, array processors, communications networks, queuing systems, have received more and more attention from both industry and academia. These systems are typical examples of discrete-event systems (DES). The dynamics of DES are event-driven as opposed to time-driven, i.e., the behavior of a DES evolves in time by the occurrence of events at possibly irregular (deterministic or stochastic) time intervals. An event corresponds to the start or the end of an activity. If we consider a production system, then possible events are: the arrival of raw materials, the completion of a product on a machine, a machine breakdown, and so on.

There are many modeling and analysis techniques for DES [11, 28, 75], such as queuing theory, Petri nets [102, 154], max-plus algebra [7, 74, 89], state machines, formal languages, automata, temporal logic, perturbation analysis [76], generalized semi-Markov processes, and so on. All these modeling and analysis techniques have particular advantages and disadvantages. The selection of the most appropriate method from the above methodologies depends on the system we want to model and on the goals we want to achieve. In particular, the max-plus-algebraic approach allows us to determine and to analyze many properties of the system, but this approach can only be applied to a subclass of DES with synchronization but no choice. Synchronization requires the availability of several resources at the same time, whereas choice appears when at a certain time a user has to choose among several resources. Consider a production system consisting of a finite number of machines that can manufacture several product types. Before we can assemble a product on a machine, the raw materials (intermediate goods) have to be available and the machine has to be idle. This reflects the synchronization feature. The starting time of a machine is related to the maximum of the arrival times of the raw materials (intermediate goods) and the time of completion of the previous product. The completion time of a product on a machine is the sum of the starting time of the machine and the processing time of the product. Hence, maximization and addition are the two basic operations of max-plus-algebraic models. In addition, a product may be manufactured on one of several machines that can process that product and that are idle at that time, so the product must choose among those machines. However, there is no choice if each product type has been assigned a fixed processing route.

Although in general DES lead to a nonlinear description in conventional algebra, there exists a subclass of DES, namely DES with synchronization but no choice, for which we can get a "linear" model in the max-plus algebra. Such systems are called max-plus linear (MPL) systems. There exists a remarkable analogy between the basic operations of the max-plus algebra (maximization and addition) and the basic operations of conventional algebra (addition and multiplication). As a consequence, many concepts and properties of conventional algebra also have a max-plus-algebraic analogue. Furthermore, this analogy also allows us to translate many concepts, properties, and techniques from conventional linear system theory to system theory for MPL systems. However, there are also some major differences that prevent a straightforward translation of properties, concepts, and algorithms from conventional linear algebra and linear system theory to the max-plus algebra and max-plus-algebraic system theory for DES. MPL models were first introduced in the 1960s by Cuninghame-Green [42, 43] and Giffler [59–61]. Later the theory of max-plus algebra was further developed by Cuninghame-Green [44, 45] and this topic attracted more attention, e.g., [35–37, 65, 66, 105–107]. The book [7] provides an extensive overview of related work until the early 1990s. A more recent textbook on max-plus algebra and its applications is [74]. Many results have been obtained for modeling and control of MPL systems, see [1, 2, 27, 49, 58, 63, 69–71, 77, 79, 84, 95, 98, 124–126, 130, 136, 143, 144] and the references therein.

In recent decades technological innovations have caused a considerable interest in the study of dynamical processes of a mixed continuous and discrete nature. Such processes are called hybrid systems [73, 93, 139] and they are characterized by the interaction of continuous-variable models (governed by differential or difference equations) on the one hand, and logic rules and DES (described by, e.g., automata, finite state machines, ets.) on the other hand. Hybrid systems arise in many fields and some specific examples of hybrid systems are temperature control systems, electrical networks with diodes and switches, traffic networks, power networks, manufacturing systems, robots, fermentation processes, etc. One particular feature of hybrid systems is that there exist many different modeling frameworks [3, 15, 93, 139] (such as hybrid automata, timed Petri nets, piecewise affine systems, ...), that offer a trade-off between modeling power and decision power, i.e., the more accurate a model is, the more difficult it is to make analytic statements about the model (often resulting in intractable, NP-hard or undecidable problems).

Piecewise affine (PWA) systems are defined using a number of non-overlapping polyhedral regions in the input-state space, where in each region the system has affine dynamics. Typical examples of systems that can be modeled using PWA systems are electrical networks, mechanical systems subject to constraints, and systems subject to saturation. In fact, PWA systems can be considered as one of the simplest extensions of the well-known class of linear systems, that on the one hand can describe nonlinear phenomena and also approximate nonlinear systems to any desired accuracy, and for which on the other hand tractable analysis and control methods have been developed. The earliest result on PWA systems is [127]. PWA systems have been studied by many researchers [8, 10, 12, 32, 33, 50, 80, 81, 86, 87, 90, 112, 115, 121, 129, 131, 140, 145, 146, 153].

Model predictive control (MPC) [94, 114] has been developed for application in the process industry, where it has become a very popular advanced control strategy. A key advantage of MPC is that it is able to deal with multi-input multi-output systems and that it can include constraints on input, outputs, and states. Furthermore, MPC can handle structural changes, such as sensor or actuator failures and changes in system parameters or system structure, by adapting the model. In essence, MPC uses a prediction model in combination with (on-line) optimization to determine a sequence of control inputs that optimizes a given performance criterion over a given prediction window subject to various operation constraints. The computed control inputs are applied to the system in a moving horizon fashion (i.e., the first control input sample is applied to the system, after which the new state of the system is measured or estimated and the whole optimization procedure is repeated), which introduces feedback into the control loop.

In general, most MPC approaches for MPL systems and PWA systems subject to linear constraints and/or general linear or piecewise objective functions result in mixed integer linear programming (MILP) problems. Although there exist efficient solvers for MILP problems, MILP is in essence an NP-hard problem, which implies that the computation time required to solve the problem increases significantly if the size of the MPC problem increases (e.g., when higher-order systems or longer control horizons are considered). Hence, there is a need for efficient MPC approaches for MPL systems and PWA systems.

1.2 Research goals and approach

The main aim of this thesis is to develop efficient model-based optimal control approaches for (stochastic) MPL systems and continuous PWA systems. In this thesis we will in particular focus on the following topics:

1. improving the efficiency of current MPC approaches for MPL systems,
2. improving the performance of current MPC approaches for continuous PWA systems (with linear constraints on the inputs and the outputs),
3. extension of MPC approaches to stochastic MPL systems.

To achieve these research goals, in this thesis we will consider the following approaches to reduce the computational burden of the MPC optimization problem:

- **Optimistic optimization algorithms**

  Optimistic optimization [18, 68, 100, 101, 132] is a class of algorithms that start from a hierarchical partition of the feasible set and gradually focus on the most promising area until they eventually perform a local search around the global optimum of the function. A sequence of feasible solutions is generated during the process of iterations and the best solution is returned at the end of the algorithm. The gap between the best value returned by the algorithm and the real global optimum can be expressed as a function of the number of iterations, which can be specified in advance.

- **Optimistic planning algorithms**

  Optimistic planning [20, 22, 78, 96, 101] is a class of planning algorithms originating in artificial intelligence applying the ideas of optimistic optimization. This class of algorithms works for discrete-time systems with general nonlinear (deterministic or stochastic) dynamics and discrete control actions. Based on the current system state, a control sequence is obtained by optimizing an infinite-horizon sum of discounted bounded stage costs (or the expectation of these costs for the stochastic case). Optimistic planning uses a receding-horizon scheme and provides a characterization of the relationship between the computational budget and near-optimality.

- **Stochastic model predictive control**

  Due to model mismatch or disturbances, uncertainties are often considered in the prediction model of MPC. For the situation that the uncertainties are characterized as random variables, stochastic MPC [54, 99] has emerged as a useful control design method where usually the expected value of an objective function is optimized subject to input, state, or output constraints. Due to the probabilistic nature of the uncertainties, those constraints are usually formulated as chance constraints, i.e., the probability of constraint violation is limited to a predefined probability level. Stochastic MPC takes advantage of the knowledge of the probability distributions of the uncertainties and is based on stochastic programming and chance-constrained programming [25, 30, 53, 142].

1.3 Contributions of the thesis

The main contributions achieved in this thesis are listed as follows:

- We adapt optimistic optimization for solving the MPC optimization problem for MPL systems. We consider MPC for MPL systems with simple bound constraints on the increments of the control inputs. The objective function is a trade-off between the output cost (i.e., weighted tardiness-earliness penalty with respect to a due-date signal) and the input cost (i.e., feed as late as possible). A dedicated semi-metric is developed satisfying the necessary requirements for optimistic optimization. Based on the theoretical analysis, we prove that the complexity of optimistic optimization is exponential in the control horizon instead of the prediction horizon. Hence, using optimistic optimization is computationally more efficient when the control horizon is small and the prediction horizon is large.

- The infinite-horizon optimal control problem for MPL systems is addressed. The considered objective function is a sum of discounted stage costs over an infinite horizon. We consider the increments of the control inputs as control variables and the control space is discretized as a finite set. The resulting optimal control problem is equivalently transformed into an online planning problem that involves maximizing a reward function. We adapt an optimistic planning algorithm to solve this problem. Given a finite computational budget, a control sequence is returned and the first control action or a subsequence of the returned control sequence is applied to the system and then a receding-horizon scheme is adopted. The proposed optimistic planning approach yields a characterization of the near-optimality of the resulting solution. The simulation results show that when a subsequence of the returned control sequence is applied, this approach results in a lower tracking error compared with a fintie-horizon approach.

- We further adapt optimistic optimization for solving the MPC optimization problem for continuous PWA systems. The considered 1-norm and ∞-norm objective functions are continuous PWA functions. The linear constraints on the states and the inputs are treated as soft constraints and replaced by adding a penalty function to the objective function. The proposed optimistic optimization approach is based on recursive partitioning of the resulting hyperbox feasible set. We derive expressions for the core parameters of optimistic optimization and discuss the near-optimality of the resulting solution by applying optimistic optimization. The performance of the proposed approach is illustrated with a case study on adaptive cruise control.

- We extend optimistic optimization from a hyperbox feasible set to a polytopic feasible set. More specifically, we propose a partitioning framework of the polytopic feasible set satisfying the requirements of optimistic optimization by employing Delaunay triangulation and edgewise subdivision. For this partitioning approach, we derive analytic expressions for the core ingredients that are used for characterizing the near-optimality of the solution obtained by optimistic optimization. When applied for optimizing PWA functions, the proposed optimistic optimization approach is computationally more efficient than MILP if the number of polyhedral subregions in the domain is much larger than the number of variables of the PWA function.

- MPC for stochastic MPL systems with linear constraints on the inputs and the outputs is considered. Due to the uncertainties, these linear constraints are formulated as probabilistic or chance constraints. The proposed probabilistic constraints can be equivalently rewritten into a max-affine form (i.e., the maximum of affine terms) if the linear constraints are monotonically nondecreasing as a function of the outputs. Based on the resulting max-affine form, two methods are developed for solving the chance-constrained MPC problem for stochastic MPL systems: Method 1 uses Boole's inequality to convert the multivariate chance constraints into univariate chance constraints for which the probability can be computed more efficiently. Furthermore, Method 2 employs the multidimensional Chebyshev inequality and transforms the multivariate chance constraints into constraints that are linear in the inputs. The simulation results show that the two proposed methods are faster than the Monte Carlo simulation method and yield lower closed-loop costs than the nominal MPC method.

1.4 Outline of the thesis

The structure of this thesis is illustrated in Figure 1.1. Chapter 2 presents the background knowledge required to understand the main contributions of this thesis. Chapter 3 addresses model-based control of MPL systems by using optimistic optimization and optimistic planning respectively. In Chapters 4 and 5, optimistic optimization is applied to solve the MPC optimization problem of continuous PWA systems and further the more general optimization problem of continuous nonconvex PWA functions with a given polytopic feasible set. In Chapter 6, we investigate efficient MPC approaches for stochastic MPL systems with chance constraints.

More specifically, the thesis is organized as follows:

Chapter 2 *Background*

First, the basics of max-plus algebra, max-plus linear (MPL) discrete-event systems and piecewise affine (PWA) systems are presented. Next, we provide a short introduction to model predictive control (MPC) for general nonlinear systems. Moreover, the formulations of MPC approach for MPL systems and PWA systems are presented. Afterwards, we describe optimistic optimization algorithms as well as one particular algorithm, the deterministic optimistic optimization (DOO) algorithm including its hierarchical partitioning framework, necessary assumptions, and performance analysis. Optimistic optimization algorithms have been applied to planning problems resulting in optimistic planning algorithms. In this chapter we discuss one variant called optimistic planning for deterministic systems (OPD).

Chapter 3 *Optimistic optimization and planning for model-based control of MPL systems*

This chapter considers model-based control of MPL systems with continuous and discrete-valued control variables respectively. Here control variables refer to the increments of the control inputs. On the one hand, we apply DOO to solve the MPC optimization problem of MPL systems with continuous-valued control variables, which usually leads to a nonsmooth nonconvex optimization problem. Dedicated semi-metrics are developed for different types of objective functions such that the required assumptions of DOO are satisfied. On the other hand, we address the infinite-horizon optimal control problem of MPL systems with discrete-valued control variables. OPD is used to solve such problem where a sum of discounted state costs over an infinite horizon is considered as the objective function.

This chapter is based on the papers [147–149].

Chapter 4 *Optimistic optimization for MPC of continuous PWA systems*

In general MPC for continuous PWA systems leads to a nonlinear, nonconvex optimization problem. In this chapter we consider 1-norm and ∞-norm objective functions subject to linear constraints on the states and the inputs. The feasible set is transformed into a hyperbox by considering the linear constraints as soft constraints and adding a penalty function to the objective function. Based on recursive partitioning of the hyperbox, analytic expressions for the core parameters required by DOO are derived. Then the guarantee on the performance of the solution returned by the algorithm is discussed in terms of these parameters.

This chapter is based on the paper [150].

Chapter 5 *Optimistic optimization of continuous nonconvex PWA functions*

From the previous chapter, it is observed that the optimization of continuous nonconvex PWA functions arises in the context of control of continuous PWA systems. In order to get a hyperbox feasible set, the linear constraints on the states and the inputs are treated as soft constraints and replaced by a penalty function. To prevent this compromise, in this chapter we consider the optimization of continuous nonconvex PWA functions over a given polytope with arbitrary shape. As a consequence, we need to design an alternative partitioning approach instead of the standard partitioning. We introduce a partitioning approach by employing Delaunay triangulation and edgewise subdivision based on which DOO is applied to solve such optimization problem. This leads to a better performance than the MILP method when the number of polyhedral subregions in the domain of the PWA function is large.

This chapter is based on the paper [151].

Chapter 6 *MPC for stochastic MPL systems with chance constraints*

The behavior of an MPL system evolves in time by the occurrence of events at possibly irregular time intervals. In practice, these time intervals may be not deterministic due to stochastic durations of the activities. In this chapter we consider MPC for stochastic MPL systems where the distribution of the stochastic uncertainties is supposed to be known. Due to the uncertainties, the linear constraints on the inputs and the outputs are formulated as probabilistic or chance constraints, i.e., the constraints are required to be satisfied with a predefined probability level. Under the assumption that the linear constraints are monotonically nondecreasing as a function of the outputs, the proposed chance constraints are equivalently rewritten into a max-affine form (i.e., the maximum of affine terms). Subsequently, two approaches based on Boole's inequality and Chebyshev's inequality respectively are developed to solve the chance-constrained MPC problem for stochastic MPL systems.

This chapter is based on the paper [152].

Chapter 7 *Conclusions and recommendations*

The thesis is concluded with the main contributions and some recommendations for future research.

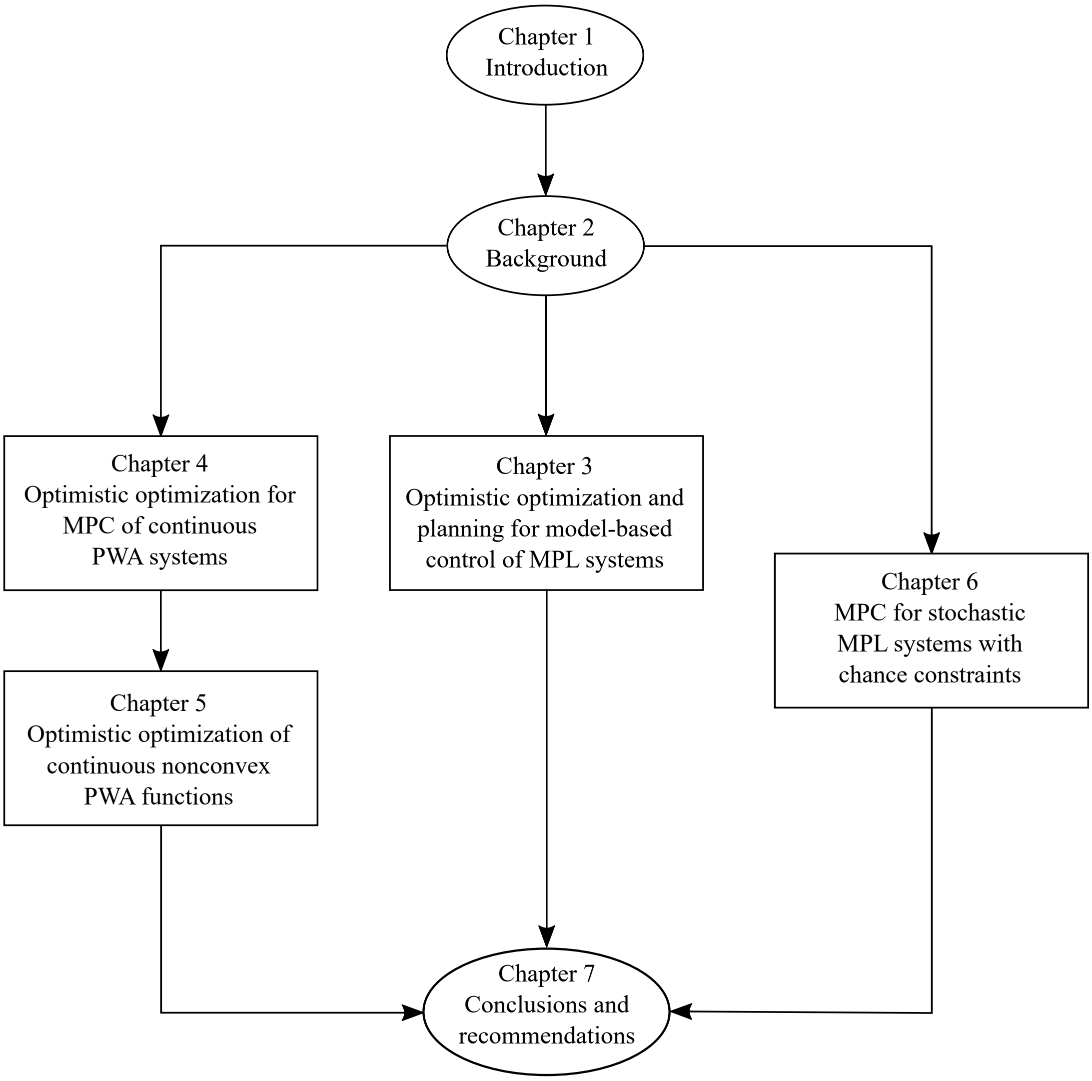


Figure 1.1: Structure of the thesis

Chapter 2

Background

In this chapter we first give a brief overview of max-plus linear systems and piecewise affine systems. Next, we introduce model predictive control for general nonlinear systems and present the formulation of model predictive control for max-plus linear systems and piecewise affine systems. Subsequently, we provide a description of optimistic optimization algorithms and optimistic planning algorithms.

2.1 Max-plus linear (MPL) discrete-event systems

Complex discrete-event systems (DES) include man-made systems, such as production systems, railway networks, logistic systems, that consist of a finite number of resources (e.g., machines, railway tracks) shared by several users (e.g., workpieces, trains) all of which pursue some common goal (e.g., the assembly of products, transportation of people or goods) [7]. The state transitions of such systems are driven by the occurrence of asynchronous events. Events correspond to starting or ending of some time-consuming activities (e.g., the start or completion of a processing step, the arrival or departure of a train in a station). In general, DES lead to nonlinear descriptions in conventional algebra [7, 28]. However, there exists a subclass of DES for which we can get a “linear” model in the max-plus algebra [7, 74], which has maximization and addition as its basic operations. These systems are called max-plus linear (MPL) systems. In the next subsections, we introduce some basic concepts of the max-plus algebra and of MPL systems.

2.1.1 Max-plus algebra

Define $\varepsilon = -\infty$ and $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$. For any $x, y \in \mathbb{R}_\varepsilon$, define the operations $\oplus$ and $\otimes$ by

$$x \oplus y = \max(x, y), \quad x \otimes y = x + y.$$

The structure $(\mathbb{R}_\varepsilon, \oplus, \otimes)$ is called the max-plus algebra [7, 45, 74]. The operations $\oplus$ and $\otimes$ are called the max-plus-algebraic addition and max-plus-algebraic multiplication, respectively. Many concepts and properties from linear algebra can be translated to the max-plus algebra by replacing $+$ by $\oplus$ and $\cdot$ by $\otimes$. The elements ε and 0 are called the zero element and identity element, respectively, i.e., for any $x \in \mathbb{R}_\varepsilon$, we have

$$x \oplus \varepsilon = \varepsilon \oplus x = x, \quad x \otimes \varepsilon = \varepsilon \otimes x = \varepsilon, \quad x \otimes 0 = 0 \otimes x = x.$$

For matrices $A, B \in \mathbb{R}_\varepsilon^{m\times n}$ and $C \in \mathbb{R}_\varepsilon^{n\times p}$, the max-plus-algebraic operations can be extended as follows:

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij}), \quad i = 1, \ldots, m,\ j = 1, \ldots, n,$$
$$[A \otimes C]_{il} = \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kl} = \max_{k=1,\ldots,n} (a_{ik} + c_{kl}), \quad i = 1, \ldots, m,\ l = 1, \ldots, p.$$

The $m \times n$ zero matrix $\mathcal{E}_{m\times n}$ in the max-plus algebra has all its entries equal to ε. The $n \times n$ identity matrix E_n in the max-plus algebra has the diagonal entries equal to 0 and the other entries equal to ε. The max-plus algebraic matrix power of $A \in \mathbb{R}_\varepsilon^{n\times n}$ is defined as follows:

$$A^{\otimes^0} = E_n \ ,$$
$$A^{\otimes^k} = A \otimes A^{\otimes^{k-1}}, \text{ for } k = 1, 2, \ldots.$$

Note that we use both linear algebra and the max-plus algebra in this thesis. Therefore, we will always write $\oplus$ and $\otimes$ explicitly in all equations. The conventional multiplication $(\cdot)$ is usually omitted.

2.1.2 MPL systems

MPL systems are characterized by synchronization (expressed by maximization, i.e., a new operation starts as soon as all preceding operations have been finished), passing of time (expressed by addition, the finishing time of an operation equals the starting time plus the duration of activities), and the absence of choice. Synchronization requires the availability of several resources at the same time (e.g., if we consider a production system, a processing step can only start as soon as raw materials or intermediate products are available and the previous cycle has been completed), whereas choice appears when some user must choose among several resources (e.g., the absence of choice implies that a production system has been assigned a fixed route schedule for each workpiece) [7]. MPL systems can be described by equations of the following form:

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k), \tag{2.1}$$
$$y(k) = C \otimes x(k), \tag{2.2}$$

where the index k is the event counter, $x(k) \in \mathbb{R}_\varepsilon^{n_x}$ is the state, $u(k) \in \mathbb{R}_\varepsilon^{n_u}$ is the input, $y(k) \in \mathbb{R}_\varepsilon^{n_y}$ is the output, and where $A \in \mathbb{R}_\varepsilon^{n_x\times n_x}$, $B \in \mathbb{R}_\varepsilon^{n_x\times n_u}$, and $C \in \mathbb{R}_\varepsilon^{n_y\times n_x}$ are the system matrices.

The elements of $u(k)$, $x(k)$, and $y(k)$ are typically time instants at which input events, internal processes, and output events occur for the k-th time. For example, if we consider the MPL system (2.1)-(2.2) as a model of a manufacturing system, then $u(k)$ represents the k-th feeding times of raw materials, $x(k)$ contains the k-th starting times of the production processes, and $y(k)$ gives the k-th completion times for the end products. Note that in practice the event times can easily be measured; so we consider the case of full state information in this thesis.

Since the inputs represent event times, a typical constraint of MPL systems is that the input sequence should be nondecreasing, i.e.,

$$u(k+1) - u(k) \geq 0, \quad k = 0, 1, 2, \ldots. \tag{2.3}$$

2.2 Piecewise affine (PWA) systems

Hybrid systems contain both continuous and discrete dynamics that interact with each other. Typical examples of hybrid systems include temperature control systems, automotive engines, chemical processes, robotic manufacturing systems, and air-traffic management systems [3]. PWA systems [81, 127] are often used to model hybrid systems and have received increasing attention since they are capable of describing hybrid phenomena and since they are considered as the "simplest" extension of linear systems that can approximate nonlinear and nonsmooth systems with arbitrary accuracy. Briefly speaking, PWA systems are defined using a polyhedral partition of the state and input space where each polyhedron is associated with an affine dynamical description. Next, we present some definitions related to PWA systems and some descriptions equivalent to PWA systems.

2.2.1 Definitions

This section is based on [15, 113].

Definition 2.1 (Polyhedron) *A polyhedron $\mathscr{P}$ is a convex set given as the intersection of a finite number of closed half-spaces, i.e.,*

$$\mathscr{P} = \{x \in \mathbb{R}^n | Ax \le b\},$$

for some matrix A and some vector b.

Definition 2.2 (Polytope) *A bounded polyhedron $\mathscr{P}$ is called a polytope. A polytope $\mathscr{P}$ can also be defined as the convex hull of a finite number of points and can be written as*

$$\mathscr{P} = \Big\{ \sum_{i=1}^{V_{\mathscr{P}}} \lambda_i v_i \Big| \lambda_i \ge 0, i = 1, \ldots, V_{\mathscr{P}}, \sum_{i=1}^{V_{\mathscr{P}}} \lambda_i = 1 \Big\},$$

where v_i denotes the i-th vertex of $\mathscr{P}$ and $V_{\mathscr{P}}$ is the total number of vertices of $\mathscr{P}$.

Definition 2.3 (Polyhedral partition) *Given a polyhedron $\mathscr{P} \subseteq \mathbb{R}^n$, then a polyhedral partition of $\mathscr{P}$ is a finite collection $\{\mathscr{P}_i\}_{i=1}^N$ of nonempty polyhedra satisfying*
(i) $\bigcup_{i=1}^N \mathscr{P}_i = \mathscr{P}$;
(ii) $(\mathscr{P}_i \setminus \partial\mathscr{P}_i) \bigcap (\mathscr{P}_j \setminus \partial\mathscr{P}_j) = \emptyset$ for all $i \neq j$ where ∂ denotes the boundary.

Definition 2.4 (PWA function) *A scalar-valued function $f : \mathscr{P} \to \mathbb{R}$, where $\mathscr{P} \subseteq \mathbb{R}^n$ is a polyhedron, is PWA if there exists a polyhedral partition $\{\mathscr{P}_i\}_{i=1}^N$ of $\mathscr{P}$ such that f is affine on each $\mathscr{P}_i$, i.e.*

$$f(x) = \alpha_{(i)}^T x + \beta_{(i)},$$

for all $x \in \mathscr{P}_i$, with $\alpha_{(i)} \in \mathbb{R}^n$, $\beta_{(i)} \in \mathbb{R}$, for $i = 1, \ldots, N$.

If a PWA function f is continuous on the boundary of any two neighboring regions, then f is said to be continuous PWA.

A vector-valued function is continuous PWA if each of its components is continuous PWA.

Proposition 2.5 *[67, 108] If $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous PWA function, then f can be represented in the max-min canonical form*

$$f(w) = \max_{i \in \mathscr{I}} \min_{j \in \mathscr{J}_i} \{\alpha_{ij}^T w + \beta_{ij}\}, \tag{2.4}$$

or in the min-max canonical form

$$f(w) = \min_{i \in \tilde{\mathscr{I}}} \max_{j \in \tilde{\mathscr{J}}_i} \{\tilde{\alpha}_{ij}^T w + \tilde{\beta}_{ij}\}, \tag{2.5}$$

where $\mathscr{I}, \mathscr{J}_i, \tilde{\mathscr{I}}, \tilde{\mathscr{J}}_i$ are finite index sets and $\alpha_{ij}, \tilde{\alpha}_{ij} \in \mathbb{R}^n$, $\beta_{ij}, \tilde{\beta}_{ij} \in \mathbb{R}$ for all i, j. For vector-valued functions, the above forms exist component-wise.

2.2.2 PWA systems

A PWA system is a dynamical system of the form

$$\begin{aligned} x(k+1) &= f_X(x(k), u(k)), \\ y(k) &= f_Y(x(k), u(k)), \end{aligned}$$

where f_X, f_Y are vector-valued PWA functions. Moreover, if f_X, f_Y are continuous, then the system is continuous PWA.

Consider the following explicit description of a discrete-time PWA system:

$$x(k+1) = A_i x(k) + B_i u(k) + g_i, \quad \text{for} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \mathscr{P}_i, \tag{2.6}$$

where the index k is the time counter, $x(k) \in \mathbb{R}^{n_x}$ is the state, $u(k) \in \mathbb{R}^{n_u}$ is the input, A_i, B_i, and g_i are the system matrices and vectors for the i-th region with $i \in \{1, \ldots, N\}$ where N is the number of regions. Each region $\mathscr{P}_i$ is a polyhedron given as $\mathscr{P}_i = \{F_i x(k) + G_i u(k) \leq h_i\}$ where F_i, G_i, and h_i are suitable matrices and vectors and $\{\mathscr{P}_i\}_{i=1}^N$ is a polyhedral partition of the state and input space.

As shown in [8], the system (2.6) can equivalently be represented as

$$\begin{aligned} & x(k+1) = \sum_{i=1}^N z_i(k), \\ & z_i(k) \triangleq [A_i x(k) + B_i u(k) + g_i] \sigma_i(k), \\ & \sum_{i=1}^N \sigma_i(k) = 1, \\ & E_{1k} u(k) + E_{2k} \sigma(k) + E_{3k} z(k) \leq E_{4k} x(k) + E_{5k}, \end{aligned} \tag{2.7}$$

where $\sigma_i(k) \in \{0, 1\}$, $\sigma(k) = \begin{bmatrix} \sigma_1(k) & \cdots & \sigma_N(k) \end{bmatrix}^T$, $z(k) = \begin{bmatrix} z_1(k) & \cdots & z_N(k) \end{bmatrix}^T$, and $E_{1k}, \ldots, E_{5k}$ are appropriately defined linear constraint matrices at time step k. Systems in the form of (2.7) are a specific type of mixed logical dynamical systems.

Definition 2.6 (Max-min-plus-scaling (MMPS) function) *A scalar-valued MMPS function $f : \mathbb{R}^n \to \mathbb{R}$ is defined by the recursive grammar*

$$f(x) = x_i | \alpha | \max(f_k(x), f_l(x)) | \min(f_k(x), f_l(x)) | f_k(x) + f_l(x) | \beta f_k(x),$$

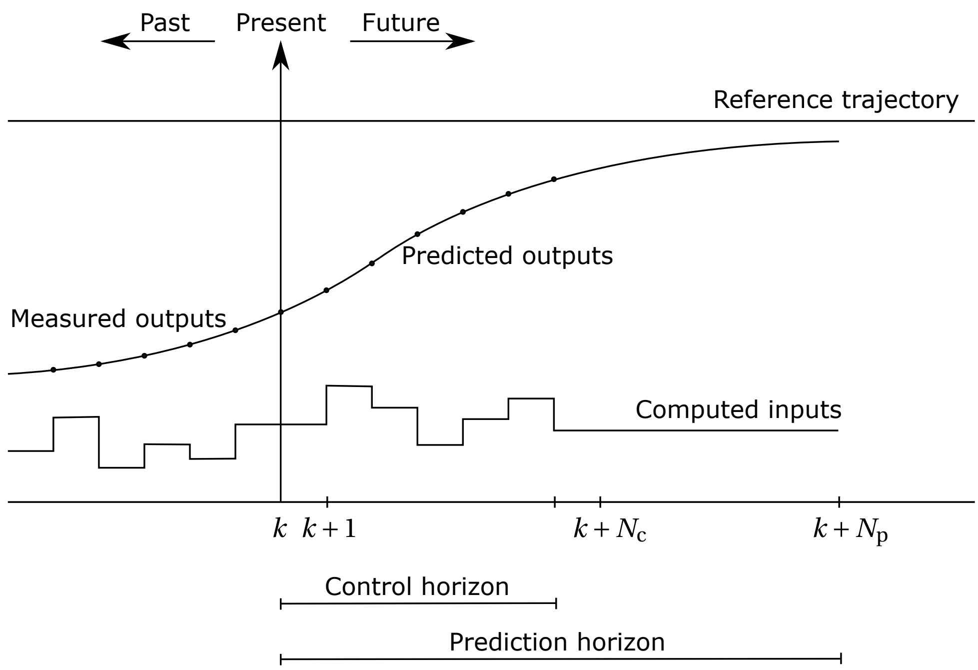


Figure 2.1: MPC scheme [26]

with $i \in \{1,\dots,n_x\}$, $\alpha,\beta \in \mathbb{R}$, *and where* $f_k : \mathbb{R}^n \to \mathbb{R}$, $f_l : \mathbb{R}^n \to \mathbb{R}$ *are again MMPS functions; the symbol* | *stands for "or", and max and min are performed entrywise.*

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ *is a (general) MMPS function if all entries are scalar-valued MMPS functions.*

Systems that can be described as

$$x(k+1) = \mathscr{M}(x(k), u(k)), \tag{2.8}$$

with input u and state x and where $\mathscr{M}$ is an MMPS function, are called MMPS systems.

A scalar-valued MMPS function $\mathscr{M}$ can be rewritten into the max-min canonical form (2.4) or into the min-max canonical form (2.5) with $w = [x^T \ u^T]^T$. For vector-valued MMPS functions this statement holds componentwise. By introducing additional auxiliary variables or extra constraints, the equivalence between (2.6) and (2.8) can be established (see [72] for details). If the system (2.6) is continuous (i.e. the right-hand side of (2.6) is continuous on the boundary of any two neighboring regions), then a direct connection between (2.6) and (2.8) can be derived following Proposition 2.5 (see [48] for details).

2.3 Model predictive control (MPC)

2.3.1 MPC for general nonlinear systems

Model predictive control (MPC) [26, 57, 94, 114, 116] is an advanced control strategy for control of multivariate systems in the presence of input and state/output constraints. Figure 2.1 is a representation of the MPC strategy. In MPC, a prediction model is used to predict the future outputs from time step $k+1$ up to $k+N_p$ where N_p is called the prediction horizon. The prediction of outputs depends on the known inputs, states, and outputs up to the current time step k and on the future input sequence

$u(k), \ldots, u(k+N_\mathrm{p}-1)$ which are to be calculated. At every step, the future input sequence is calculated by optimizing a given objective function subject to constraints on states, inputs, and outputs. In addition, a control horizon $N_\mathrm{c} \leq N_\mathrm{p}$ is usually used in MPC to reduce the number of variables of the MPC optimization problem by assuming

$$u(k+j) = u(k+N_\mathrm{c}-1),$$

for $j = N_\mathrm{c}, \ldots, N_\mathrm{p}-1$, resulting in a decrease of the computational burden.

Consider a general nonlinear discrete-time system of the form

$$x(k+1) = f(x(k), u(k)), \tag{2.9}$$

$$y(k) = h(x(k), u(k)), \tag{2.10}$$

where f and h are the state and output functions, the vector x represents the state, u is the input, and y is the output. Define the sequence vectors

$$\begin{aligned}
\tilde{x}(k) &= \begin{bmatrix} x^T(k+1) & \cdots & x^T(k+N_\mathrm{p}) \end{bmatrix}^T, \\
\tilde{y}(k) &= \begin{bmatrix} y^T(k+1) & \cdots & y^T(k+N_\mathrm{p}) \end{bmatrix}^T, \\
\tilde{u}(k) &= \begin{bmatrix} u^T(k) & \cdots & u^T(k+N_\mathrm{p}-1) \end{bmatrix}^T.
\end{aligned}$$

At time step k, the MPC optimization problem is then described as follows:

$$\min_{\tilde{u}(k), \tilde{x}(k), \tilde{y}(k)} J(k) \tag{2.11}$$

subject to

$$\text{the prediction model (2.9)-(2.10)}, \tag{2.12}$$

$$u(k+j) = u(k+N_\mathrm{c}-1) \text{ for } j = N_\mathrm{c}, \ldots, N_\mathrm{p}-1, \tag{2.13}$$

$$\big(\tilde{u}(k), \tilde{x}(k), \tilde{y}(k)\big) \in \mathbb{C}, \tag{2.14}$$

where $\mathbb{C}$ represents the set of feasible states, feasible outputs, and feasible inputs and where J is a given objective function, usually a function of the input energy and the differences between the predicted outputs and the reference signal. The optimal future input sequence is determined by solving the problem (2.11)-(2.14). Moreover, MPC uses a receding-horizon principle. At time step k, only the first element $u(k)$ of the optimal input sequence is applied to the system. At the next time step, the known information is updated by new measurements and the prediction horizon is shifted. The problem (2.11)-(2.14) is solved again at time step $k+1$ based on the new information. The feedback from the measurements makes MPC a closed-loop controller. The whole process is represented in Figure 2.2.

2.3.2 MPC for MPL systems

The MPC framework has been extended to MPL systems in [47]. In this section, we briefly introduce the formulation of MPC problem for MPL systems. We consider the following MPL

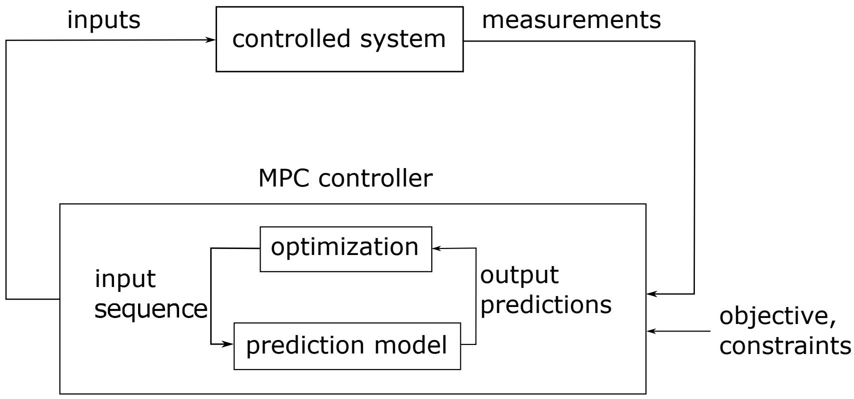


Figure 2.2: MPC loop

system:

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k), \tag{2.15}$$

$$y(k) = C \otimes x(k), \tag{2.16}$$

where $A \in \mathbb{R}_\varepsilon^{n_x \times n_x}$, $B \in \mathbb{R}_\varepsilon^{n_x \times n_u}$, and $C \in \mathbb{R}_\varepsilon^{n_y \times n_x}$. As indicated in Section 2.1.2, we assume that at event step k, the state $x(k)$ can be measured or estimated using previous measurements. We can then use (2.15)-(2.16) to predict the future outputs of the system from event step $k+1$ up to $k+N_\mathrm{p}$. Define the sequence vectors

$$\tilde{y}(k) = \begin{bmatrix} y^T(k+1) & \cdots & y^T(k+N_\mathrm{p}) \end{bmatrix}^T,$$
$$\tilde{u}(k) = \begin{bmatrix} u^T(k) & \cdots & u^T(k+N_\mathrm{p}-1) \end{bmatrix}^T.$$

The evolution of the MPL system can be presented as follows [47]:

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k), \tag{2.17}$$

where

$$H = \begin{bmatrix} C \otimes B & \varepsilon & \cdots & \varepsilon \\ C \otimes A \otimes B & C \otimes B & \cdots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ C \otimes A^{\otimes^{N_\mathrm{p}-1}} \otimes B & C \otimes A^{\otimes^{N_\mathrm{p}-2}} \otimes B & \cdots & C \otimes B \end{bmatrix},$$

$$g(k) = \begin{bmatrix} C \otimes A \\ C \otimes A^{\otimes^2} \\ \vdots \\ C \otimes A^{\otimes^{N_\mathrm{p}}} \end{bmatrix} \otimes x(k) \ .$$

In [47], different choices for the objective function in MPC for MPL systems have been considered. A typical example of an objective function J at event step k is as follows:

$$J(k) = J_\mathrm{out}(k) + \lambda J_\mathrm{in}(k),$$

$$J_{\text{out}}(k) = \sum_{j=1}^{N_\text{p}} \sum_{i=1}^{n_y} \max(y_i(k+j) - r_i(k+j), 0),$$

$$J_{\text{in}}(k) = -\sum_{j=1}^{N_\text{p}} \sum_{l=1}^{n_u} u_l(k+j-1),$$

where the nonnegative scalar λ is the trade-off between the output objective function J_{out} and the input objective function J_{in}. Considering a manufacturing system, J_{out} corresponds to a penalty for every late delivery and J_{in} corresponds to feeding the raw materials as late as possible.

The MPL-MPC problem at event step k is defined as follows:

$$\min_{\tilde{u}(k),\tilde{x}(k),\tilde{y}(k)} J(k) \tag{2.18}$$

subject to

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k), \tag{2.19}$$

$$u(k+j) \geq u(k+j-1), \text{ for } j = 0, \ldots, N_\text{p} - 1, \tag{2.20}$$

$$\Delta u(k+j) = \Delta u(k+N_\text{c}-1), \text{ for } j = N_\text{c}, \ldots, N_\text{p} - 1, \tag{2.21}$$

$$\big(\tilde{u}(k), \tilde{x}(k), \tilde{y}(k)\big) \in \mathbb{C} , \tag{2.22}$$

where $\Delta u(k) = u(k) - u(k-1)$. In general, the problem (2.18)-(2.22) is a nonlinear nonconvex optimization problem due to the nonconvex constraint $\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k)$. If inputs, states, and outputs are bounded, then the problem can be transformed into a mixed-integer linear programming problem. For some special cases, namely, if the objective function is a monotonically non-decreasing piecewise affine function of the output and an affine function of the input and if the constraints are linear and monotonically non-decreasing as a function of the output, then the problem can be reduced to a linear programming problem [47].

2.3.3 MPC for PWA systems

Since PWA systems are a special class of nonlinear systems, the MPC problem for PWA systems can be defined similarly as in Section 2.3.1 with the difference that the prediction model in problem (2.11)-(2.14) is replaced by the PWA model or its equivalent forms. In MPC for PWA systems, the output objective function is usually taken as a $1/2/\infty$-norm of the differences between the output and the reference signal. More details of PWA-MPC problems will be discussed in Chapter 4.

2.4 Optimistic optimization algorithms

Optimistic optimization algorithms [101] have been introduced for solving large-scale optimization problems given a finite computational budget. These algorithms can be applied to function optimization over general feasible solution spaces, such as metric spaces, trees, graphs, and Euclidean spaces. The motivation for designing optimistic optimization algorithms comes from the experimental success of the Upper Confidence

Bound strategy applied to Trees (UCT) [88] which is very efficient in sequential decision making problems. However, the potential risk of UCT is to stop exploring the optimal branch too early because the current upper confidence bound of the optimal branch is underestimated and it may take a long time to rediscover the optimal branch. This risk can possibly result in poor performance of UCT on simple problems for a limited computation time. Thus the objective of optimistic optimization algorithms is to obtain efficient algorithms with finite-time performance guarantees. The performance of optimistic optimization algorithms depends on the local behavior of the objective function around its global optima and is expressed in terms of the quantity of near-optimal solutions measured with some metric. To illustrate the basic idea of the optimistic optimization algorithms, in the next subsections, we present an optimization problem of a function f solved by an optimistic strategy, more precisely, the deterministic optimistic optimization (DOO) algorithm. This section is based on [100, 101].

Consider a minimization of a deterministic function f over a feasible set $\mathcal{X}$. The notations f and $\mathcal{X}$ remain generic in this section. Since the implementation of the optimistic optimization algorithms is based on a hierarchical partitioning of the feasible set, we first introduce the partitioning framework of the feasible set before going to the details of DOO.

2.4.1 Partitioning of the feasible set

For any integer $h \in \{0, 1, \dots\}$, the feasible set $\mathcal{X}$ is recursively split into K^h subsets (called cells) where K is a finite positive integer denoting the maximum number of child cells of a parent cell. The partition may be represented by a tree structure, as illustrated in Figure 2.3. The whole set $\mathcal{X}$ is denoted as $X^{0,0}$ and corresponds to the root node $(0,0)$ of the tree. Each cell at any depth h is denoted as $X^{h,d}$ for $d \in \{0, \dots, K^h - 1\}$ and corresponds to a node (h, d) in the tree. A cell $X^{h,d}$ at depth h is split into K child cells $\{X^{h+1,d_i}\}_{i=1}^{K}$. Each cell $X^{h,d}$ is characterized by a representative point $x^{h,d} \in X^{h,d}$ in which f may be evaluated.

Remark 2.7 For a hypercube feasible set, one can get a partitioning satisfying the assumptions of DOO by using the standard partitioning [109] where each cell is split into regular same-sized subcells and the split occurs along one dimension. For a hyperbox feasible set, the feasible set can be divided by bisecting each dimension as shown in Figure 2.3. Moreover, a partitioning approach for a polytopic feasible set is developed in Chapter 5. □

2.4.2 Assumptions

To obtain a measure of complexity of the optimization problem, some assumptions need to be made about the function and the partitioning of the feasible set [100]. These assumptions are expressed in the form of a semi-metric, which is defined as follows. Let $\mathbb{R}_{\geq 0}$ denote the set of nonnegative real numbers.

Definition 2.8 (Semi-metric ℓ) *A semi-metric on a set $\mathcal{X}$ is a function $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ satisfying the following conditions for any $x, y \in \mathcal{X}$:*

i) $\ell(x, y) = \ell(y, x) \geq 0$;

ii) $\ell(x, y) = 0$ if and only if $x = y$.

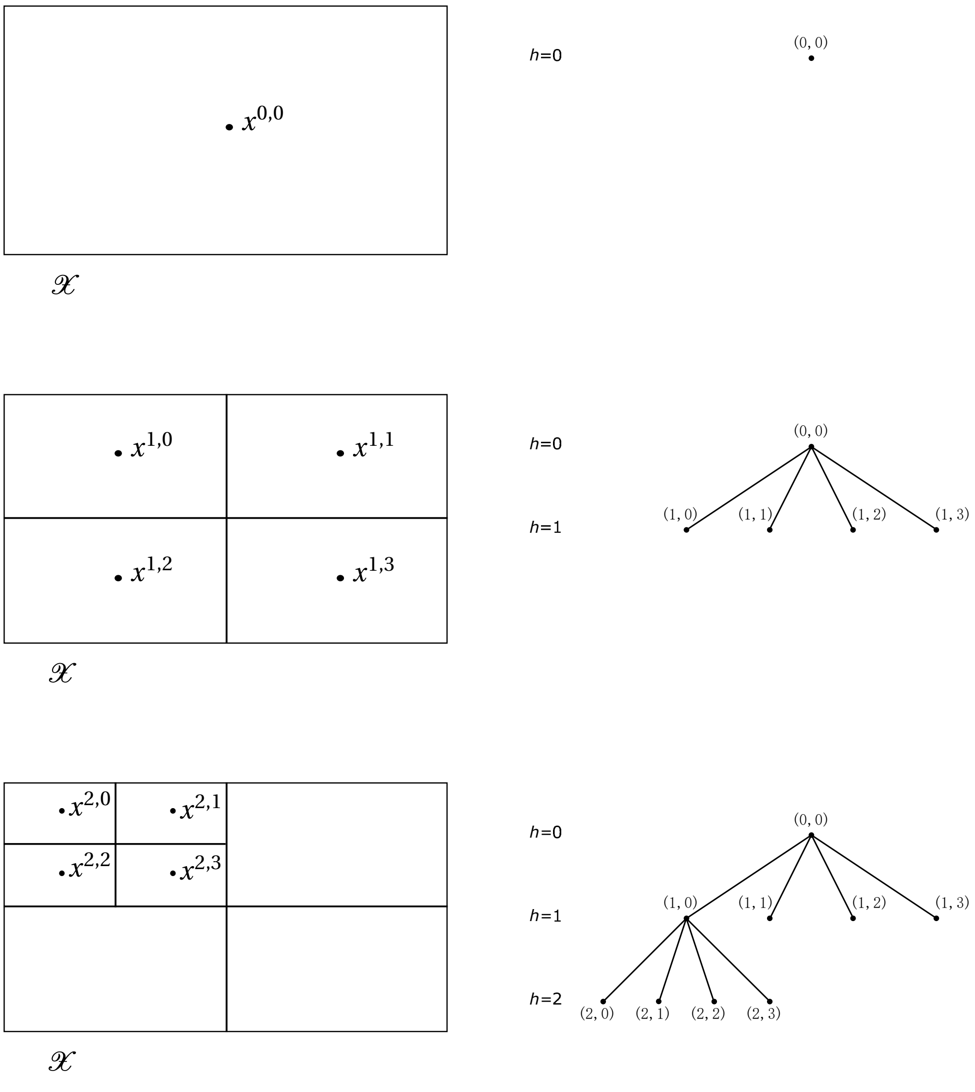


Figure 2.3: Hierarchical partitioning of the feasible set $\mathscr{X}$ represented by a tree.

Definition 2.9 (ℓ-ball) *An ℓ-ball of radius $r > 0$ centered at a point p in $\mathscr{X}$, denoted by $\mathfrak{B}_{p,r}$, is defined by*

$$\mathfrak{B}_{p,r} = \{x \in \mathscr{X} | \ell(x,p) \leq r\}.$$

We then consider the following assumptions:

Assumption 2.10 *There exists a semi-metric ℓ defined on $\mathscr{X}$ and at least one global optimizer $x^* \in \mathscr{X}$ of f (i.e., $f(x^*) = \min_{x \in \mathscr{X}} f(x)$) such that for all $x \in \mathscr{X}$,*

$$f(x) - f(x^*) \leq \ell(x, x^*).$$

Assumption 2.11 *There exists a decreasing sequence $\{\delta(h)\}_{h=0}^{\infty}$ with $\delta(h) > 0$, such that for any depth $h \in \{0,1,\ldots\}$, for any cell $X^{h,d}$ at depth h, we have*

$$\sup_{x \in X^{h,d}} \ell(x, x^{h,d}) \leq \delta(h),$$

where $\delta(h)$ is called the maximum diameter of the cells at depth h.

Assumption 2.12 *There exists a scalar $\nu > 0$ such that any cell $X^{h,d}$ at any depth h contains an ℓ-ball of radius $\nu\delta(h)$ centered in $x^{h,d}$.*

One main challenge of DOO is to design a semi-metric ℓ, a sequence $\{\delta(h)\}_{h=0}^{\infty}$, and a scalar ν that satisfy these assumptions.

Remark 2.13 Assumption 2.10 regards the local properties of f near one global optimum with respect to a semi-metric ℓ, guaranteeing that f does not decrease too fast around the global optimum. Assumptions 2.11-2.12 subsequently connect ℓ to the hierarchical partitioning that generates cells that shrink with further partitioning. Let $\delta(h)$ be the maximum diameter[1] of the cells at depth h. Loosely speaking, this means that the value of $\delta(h+1)/\delta(h)$ should be less than a given constant that is strictly smaller than 1. The scalar ν can be considered as the maximum ratio of the radius of the inscribed ball of any cell and the maximum distance between any two points in that cell. □

2.4.3 Deterministic optimistic optimization (DOO)

Deterministic optimistic optimization (DOO) algorithm is an application of the optimistic strategy in deterministic function optimization. Given a finite number n of iterations, DOO generates a sequence of feasible solutions during the iterations and returns the best solution $x(n)$ at the end of the algorithm. As shown in Figure 2.4, starting with the root node $\mathscr{T} = \{(0,0)\}$, DOO incrementally updates the tree $\mathscr{T}$ for iteration step $t = 1,\ldots,n$. For each cell $X^{h,d}$, define a b-value function, i.e.,

$$b^{h,d} = f(x^{h,d}) - \delta(h) \ .$$

At each iteration t, DOO select a leaf[2] of the current tree $\mathscr{T}$ with minimum $b^{h,d}$ value to expand by adding its K children to the current tree. Expanding a leaf (h,d) corresponds to

[1]The diameter of a cell is the maximum distance (measured by using the semi-metric ℓ) between any two points in that cell.

[2]A leaf of a tree is a node with no children. The set $\mathscr{L}$ contains the leaves of the tree $\mathscr{T}$.

Given: partitioning of $\mathscr{X}$, number n of iterations
Initialize the tree $\mathscr{T} \leftarrow \{(0,0)\}$ (root node)
for $t = 1$ to n **do**
  Select the leaf $(h^\dagger, d^\dagger) \in \mathscr{L}$ with minimum $b^{h^\dagger, d^\dagger}$ value
  Expand this node $(h^\dagger, d^\dagger)$ by adding its K children to $\mathscr{T}$
end for
Return $x(n) = \arg\min_{(h,d)\in\mathscr{T}} f(x^{h,d})$

Figure 2.4: Deterministic optimistic optimization (DOO) algorithm

splitting the cell $X^{h,d}$ into K subcells and evaluating the function f at the representative points of the children cells. Once the computational budget n is used, DOO returns the node of the tree $\mathscr{T}$ that yields the lowest function value of f, as the recommended solution. The returned result is an approximation of the global minimum of f. The performance of DOO is assessed by the difference between the approximation and the true optimal value. The analysis in the next subsection gives upper bounds on this difference.

2.4.4 Analysis of DOO

Let $(h_{\max}, d_{\max})$ be the deepest node that has been expanded by the algorithm up to n iterations. We have

$$f(x^*) \leq f(x(n)),$$

and

$$\begin{aligned} f(x(n)) &\leq f(x^{h_{\max}, d_{\max}}) \\ &\leq f(x^*) + \delta(h_{\max}) \ , \end{aligned}$$

i.e.,

$$f(x(n)) - f(x^*) \leq \delta(h_{\max}) \ .$$

So the returned solution $x(n)$ provides an upper bound $f(x(n))$ of the global minimum $f(x^*)$. In addition, the difference between the upper bound and the global minimum is bounded by $\delta(h_{\max})$.

The bound $\delta(h_{\max})$ provides a posterior guarantee on the performance of DOO and is obtained once the algorithm terminates. Moreover, the following analysis provides a priori guarantee on the performance. The performance of the algorithm depends on the complexity of the optimization problem, which may be expressed in terms of the quantity of the near-optimal solutions measured with the semi-metric ℓ.

From Assumptions 2.10-2.11, for any cell $X^{h,d}$ containing a global optimizer x^*, we have

$$\begin{aligned} b^{h,d} &= f(x^{h,d}) - \delta(h) \\ &\leq f(x^{h,d}) - \ell(x^*, x^{h,d}) \\ &\leq f(x^*) \ . \end{aligned}$$

So the b-value of any cell $X^{h',d'}$ for which $b^{h',d'} > f(x^*)$ is always greater than the b-value of a cell containing the optimal solution. At each iteration, the algorithm always selects the leaf with the smallest b-value. Consequently, only the cells satisfying $b^{h,d} \leq f(x^*)$ might be explored. The more cells satisfying $b^{h,d} \leq f(x^*)$, the slower the convergence speed of the algorithm. In general, the number of cells satisfying $b^{h,d} \leq f(x^*)$ will increase if the number of optimal solutions increases. Therefore, the algorithm is in general more efficient for problems with a unique optimal solution than for those where the optimal solution is not unique.

Let x^* be a global minimizer of f and for any $\varepsilon > 0$, let

$$\mathscr{X}_\varepsilon = \{x \in \mathscr{X} \,|\, f(x) - f(x^*) \leq \varepsilon\},$$

be the set of ε-near-optimal solutions.

Definition 2.14 *[100] The near-optimality dimension is the smallest $\eta \geq 0$ such that for any $\varepsilon > 0$, there exists a constant $C > 0$ such that the maximal number of disjoint ℓ-balls of radius $\nu\varepsilon$ with center in $\mathscr{X}_\varepsilon$ is less than $C\varepsilon^{-\eta}$.*

Theorem 2.15 *[100] Assume that there exist some constants $c > 0$ and $\gamma \in (0,1)$ such that $\delta(h) \leq c\gamma^h$ for any h. Let $x(n)$ be the solution returned after n iterations. Then we have:*

(i) If $\eta > 0$, then

$$f(x(n)) - f(x^*) \leq \left(\frac{C}{1-\gamma^\eta}\right)^{1/\eta} n^{-1/\eta}.$$

(ii) If $\eta = 0$, then

$$f(x(n)) - f(x^*) \leq c\gamma^{n/C-1}.$$

Remark 2.16 The near-optimality dimension actually characterizes the number of the ε-near-optimal solutions of f with respect to the semi-metric ℓ around the global optimum. Theorem 2.15 gives bounds on the suboptimality of the returned solution. For $\eta > 0$, the suboptimality bound decreases as a power of the computational budget n. The convergence speed of optimistic optimization is faster with smaller η. The best case is $\eta = 0$, which implies that the suboptimality bound decreases exponentially with n. Therefore, developing a semi-metric ℓ such that η is small is of great importance for optimistic optimization to be efficient. □

2.5 Optimistic planning algorithms

Besides the function optimization problems discussed in the previous section, the optimistic approach has also been applied to planning problems, resulting in optimistic planning algorithms. Optimistic planning algorithms optimize an infinite-horizon discounted reward function with the action space having a finite number of discrete actions. Optimistic planning algorithms return a sequence of actions as the recommended solution the length of which is influenced by the computational budget, the value of the discount factor, and the complexity of the problem. This is different from applying optimistic optimization algorithms in MPC, which consider a continuous feasible space and return a control sequence with a fixed length optimizing a fixed-horizon objective function. In [20], three types of optimistic planning algorithms have been reviewed, i.e., optimistic planning for deterministic systems [78], open-loop optimistic planning [17], and optimistic planning for sparsely stochastic systems [19]. Moreover, in [20] the theoretical guarantees on the performance of these algorithm are also provided. Recently, optimistic planning has been used for nonlinear networked control systems [22], and nonlinear switched systems [23]. In order to limit computations, optimistic planning with a limited number of action switches has been introduced in [96]. In this section, we present optimistic planning algorithms for solving an optimal control problem for discrete-time deterministic systems, i.e., optimistic planning for deterministic systems (OPD). This section is based on [20, 78, 101].

2.5.1 Optimistic planning for deterministic systems (OPD)

Optimistic planning for deterministic systems (OPD) [78, 101] is an algorithm that solves an optimal control problem for discrete-time deterministic systems described by an equation of the form

$$x(k+1) = f(x(k), u(k)),$$

where $f : X \times U \to X$ is the transition function, $x \in X$ is the state, and $u \in U$ is the action. The state space X is large (possibly infinite) and the action space U is finite with K possible actions, i.e., $U \triangleq \{u^1, \ldots, u^K\}$.

Given the initial state $x(0)$, OPD designs a control sequence $\boldsymbol{u} = (u(0), u(1), \ldots)$ *maximizing*[3] the following infinite-horizon discounted reward function:

$$J(\boldsymbol{u}, x(0)) = \sum_{k=0}^{\infty} \gamma^k R(k+1), \tag{2.23}$$

where $R(k) \in [0,1]$ is the reward for the transition from $x(k)$ to $x(k+1)$ as a result of $u(k)$ and where $\gamma \in (0,1)$ is the discount factor that is often used in the fields of dynamic programming and reinforcement learning and that expresses the difference in importance between future costs and present costs. The value of γ is usually selected close to 1. The optimal value of (2.23) is denoted as

$$J^*(x(0)) = \max_{\boldsymbol{u}} J(\boldsymbol{u}, x(0)).$$

For a given initial state, OPD explores the space of all possible control sequences $\boldsymbol{u}$. Define $\boldsymbol{u}_d = (u(0), \ldots, u(d-1))$ as a length d sequence with $d \in \{1, 2, \ldots\}$ and define $\boldsymbol{u}|_d$ as any infinite-length sequence of which the first d components coincide with $\boldsymbol{u}_d$. For any initial state $x(0)$, each $\boldsymbol{u}_d$ determines a state sequence $x(1), \ldots, x(d)$. Define

$$v(\boldsymbol{u}_d) = \sum_{k=0}^{d-1} \gamma^k R(k+1), \tag{2.24}$$

$$b(\boldsymbol{u}_d) = v(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma}. \tag{2.25}$$

The value $v(\boldsymbol{u}_d)$ is the sum of discounted rewards along the trajectory starting from the initial state $x(0)$ and applying the control sequence $\boldsymbol{u}_d$, and provides a lower bound of the value $J(\boldsymbol{u}|_d, x(0))$ for any $\boldsymbol{u}|_d$. On the other hand, note that $R(k) \in [0,1]$; hence,

$$J(\boldsymbol{u}|_d, x(0)) = v(\boldsymbol{u}_d) + \sum_{k=d}^{\infty} \gamma^k R(k+1)$$

[3] Now we maximize the reward function while before we minimized the objective function.

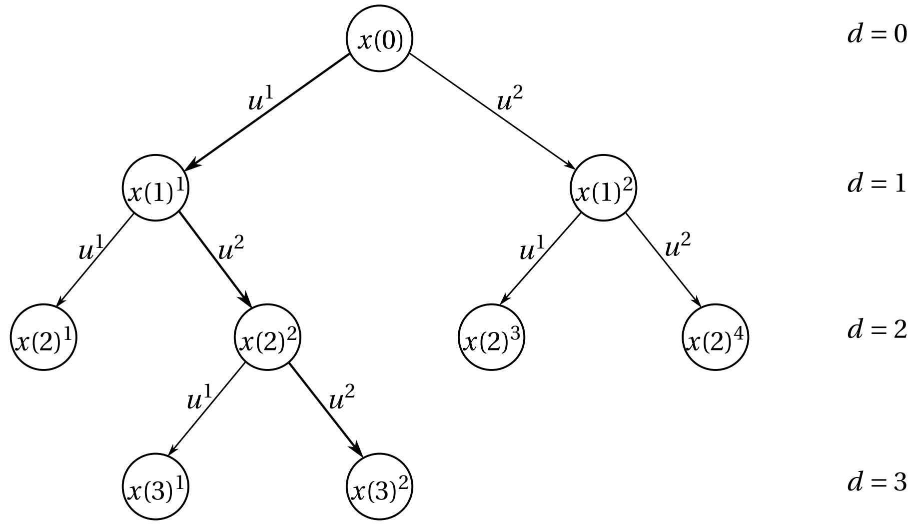


Figure 2.5: The tree representation of OPD with $K = 2$, i.e., $U = \{u^1, u^2\}$. The root node at depth $d = 0$ denotes the initial state $x(0)$. Each edge starting from a node at depth d corresponds to a control action $u(d)$. Each node corresponds to a reachable state $x(d)^i$, $i = 1, \dots, K^d$. The depth d corresponds to the time step. Any node at depth d is reached by a unique sequence $\boldsymbol{u}_d$ (e.g., the thick line for node $x(3)^2$) starting from $x(0)$.

$$\leq v(\boldsymbol{u}_d) + \sum_{k=d}^{\infty} \gamma^k \cdot 1$$
$$\leq v(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma}.$$

So $b(\boldsymbol{u}_d)$ provides an upper bound of $J(\boldsymbol{u}|_d, x(0))$ for any $\boldsymbol{u}|_d$.

The search process of OPD over the space of all possible control sequences $\boldsymbol{u}$ can be represented as a tree exploration process, as illustrated in Figure 2.5. Nodes of the tree correspond to reachable states; in particular, the root node is the initial state $x(0)$. Edges of the tree correspond to the possible control actions. Each node at some depth d is reached by a unique path through the tree, i.e., each node corresponds to a unique control sequence $\boldsymbol{u}_d = (u(0), \dots, u(d-1))$. Expanding a node means adding its K children to the current tree, i.e., generating transitions and rewards as well as computing the v and b-values for the K children. Given a finite number of node expansions, at each step, OPD always expands the most promising leaf, i.e., the control sequence $\boldsymbol{u}_d$ with the largest upper bound $b(\boldsymbol{u}_d)$. The algorithm terminates if the given number of node expansions n has been reached. Finally, the algorithm returns the control sequence $\boldsymbol{u}^*_{d'} = (u^*(0), u^*(1), \dots, u^*(d'-1))$ that maximizes the lower bound v where d' is the length of the returned optimal control sequence. The process of OPD is summarized in Figure 2.6.

OPD uses a receding-horizon scheme, so once $\boldsymbol{u}^*_{d'}$ has been computed, subsequently, only the first component $u^*(0)$ of $\boldsymbol{u}^*_{d'}$ is applied to the system, resulting in the state $x^*(1)$. At the next time step, $x^*(1)$ is used as the initial state and the whole process is repeated. From the viewpoint of the receding-horizon scheme, OPD can be seen as a variant of MPC. In MPC, a receding-horizon controller is obtained by repeatedly solving a finite-horizon open-loop optimal control problem and applying the first control input to the system. Using the current system state as the initial state, a control sequence is computed by optimizing an objective

Input: initial state $x(0)$, action space $U = \{u^1, \dots, u^K\}$, number of node expansions n
Initialize: $\mathscr{T} \leftarrow \{x(0)\}$
expand the root node by adding its K children to $\mathscr{T}$
$t \leftarrow 1$
while $t < n$
  expand the leaf with largest b-value
  $t \leftarrow t+1$
end while
Return $\boldsymbol{u}^*_{d'} = \arg\max_{\boldsymbol{u}_d \in \mathscr{L}(\mathscr{T})} v(\boldsymbol{u}_d)$ where $\mathscr{L}(\mathscr{T})$ is the set of leaves of $\mathscr{T}$

Figure 2.6: Optimistic planning for deterministic systems (OPD)

function over a finite horizon (prediction horizon). The whole procedure is repeated at the next step when new state measurements are available. Different from MPC, rather than a fixed-horizon setting OPD optimizes an infinite-horizon discounted objective function. The length of the returned control sequence is influenced by the computational budget, the value of the discount factor γ, and the complexity of the problem.

2.5.2 Analysis of OPD

Define the set of near-optimal nodes at depth d as follows:

$$\mathscr{T}^*_d = \left\{ \boldsymbol{u}_d \middle| J^*(x(0)) - v(\boldsymbol{u}_d) \le \frac{\gamma^d}{1-\gamma} \right\} .$$

OPD only expands the nodes in $\mathscr{T}^*_d$, $d \in \{0, 1, 2, \dots\}$, so the number of nodes in $\mathscr{T}^*_d$, denoted as $|\mathscr{T}^*_d|$, determines the efficiency of the algorithm. Define the asymptotic branching factor $\kappa \in [1, K]$ as $\kappa = \limsup_{d\to\infty} |\mathscr{T}^*_d|^{1/d}$, which characterizes the complexity of the problem. The following theorem summarizes the near-optimality analysis presented in [22, 78, 101].

Theorem 2.17 *Let the initial state $x(0)$ and the number of node expansions n be given.*

(i) Let $\boldsymbol{u}^*_{d'}$ be the d'-length sequence returned by the OPD algorithm and let $\boldsymbol{u}^*|_{d'}$ be any infinite-length sequence of which the first d' components coincide with $\boldsymbol{u}^*_{d'}$. Then we have

$$J^*(x(0)) - J(\boldsymbol{u}^*|_{d'}, x(0)) \le b(\boldsymbol{u}^*_{d'}) - v(\boldsymbol{u}^*_{d'}) \le \frac{\gamma^{d'}}{1-\gamma}.$$

(ii) If $\kappa > 1$, then

$$J^*(x(0)) - J(\boldsymbol{u}^*|_{d'}, x(0)) = O\Big(n^{-\frac{\log 1/\gamma}{\log \kappa}}\Big).$$

(iii) If $\kappa = 1$, then

$$J^*(x(0)) - J(\boldsymbol{u}^*|_{d'}, x(0)) = O\big(\gamma^{cn}\big),$$

where c is a constant. □

Remark 2.18 Theorem 2.17(i) provides an a posteriori bound on the near-optimality of the returned control sequence, while Theorem 2.17(ii)-(iii) imply a priori bound based on the complexity of the problem. The branching factor κ characterizes the number of nodes that will be expanded by the OPD algorithm. If $\kappa > 1$, then OPD needs a number of expansions $n = O(\kappa^d)$ to reach the depth d in the optimistic planning tree; if $\kappa = 1$, then $n = O(d)$ is required. Thus, $\kappa = 1$ is the ideal case where the number of near-optimal nodes at every depth is bounded by a constant independent of d and the a priori bound on the near-optimality decreases exponentially with n. □

2.6 Summary

In this chapter, we have summarized some basic background of max-plus algebra, max-plus linear (MPL) systems, and piecewise affine (PWA) systems. We have introduced model predictive control (MPC) for general nonlinear systems as well as for MPL systems and PWA systems. Moreover, we have presented optimistic optimization algorithms including the partitioning framework and some assumptions. We have particularly discussed the deterministic optimistic optimization (DOO) algorithm, which will be used to solve the MPC optimization problem encountered in the following chapters. Finally, we have introduced optimistic planning for deterministic systems (OPD) algorithms, which will be applied to receding-horizon control for MPL systems with discrete control variables in the next chapter.

Chapter 3

Optimistic optimization and planning for model-based control of MPL systems

In this chapter we deal with model-based control of max-plus linear (MPL) systems. We particularly consider the increments of the control inputs as control variables and investigate two cases where the control variables are respectively continuous valued and discrete valued. In the case of continuous control variables, we consider four types of output objective functions combined with just-in-time input objective functions and adapt an optimistic optimization algorithm to solve the model predictive control optimization problem for MPL systems by developing a dedicated semi-metric that satisfies the assumptions of optimistic optimization. Besides, in the case of discrete control variables, we address the infinite-horizon optimal control problem for MPL systems by using optimistic planning. More precisely, we consider a sum of discounted stage costs over an infinite horizon as the objective function. The resulting problem is solved by an optimistic planning algorithm.

3.1 Introduction

The state transitions of discrete-event systems (DES) are driven by occurrence of discrete events [28]. Events correspond to starting or ending some time-consuming activities. For example, an event may correspond to the arrival or departure of a train in a station, or the start or completion of a job on a machine. Typical examples of DES include railway networks, traffic control systems, flexible manufacturing systems, computer networks, and transportation systems. Due to the increasing complexity of these man-made systems, effective modeling tools are necessary for the analysis and control of DES. Max-plus algebra is a useful tool to model and analyze DES. Maximization and addition are two basic operations in the max-plus algebra. In conventional algebra, DES usually result in nonlinear systems, but there is a class of DES which can lead to linear systems in the max-plus algebra, called max-plus linear (MPL) systems [7, 36]. Many results for control of MPL systems have been achieved, e.g. [2, 14, 39, 58, 70, 77, 84, 95].

Model predictive control (MPC) is an advanced control design technique widely used in the process industry [26, 57, 116]. It is able to deal with multi-input multi-output systems and handle constraints on inputs and outputs. In MPC, an optimal control sequence is designed by solving an on-line optimization problem to minimize some given objective functions. The MPC framework has been extended to MPL systems in [47] (see Chapter 2 of this thesis for a brief introduction). For some special cases, the MPL-MPC problem can be formulated as a linear programming. However, in general, it results in a nonsmooth nonconvex optimization problem. To solve this problem, one approach is to recast it as a mixed integer linear programming (MILP) problem. Nonetheless, the computational complexity of most MILP algorithms grows in the worst case exponentially if the number of variables increases [123]. For the MILP problem resulting from the MPL-MPC problem, the number of auxiliary binary variables is in general proportional to the number of max operators (i.e., the prediction horizon and the number of inputs and outputs). Thus, the computation time required to solve the corresponding MILP problem will become unacceptable if the prediction horizon is large. As the event space corresponding to the prediction horizon should contain the crucial dynamics of the process, the prediction horizon can be very large for some MPL-MPC problems.

Optimistic optimization [101] is a class of algorithms that can find an approximation of the global optimal solution for nonlinear optimization problem. This method is called optimistic because the most promising solutions are examined first at each iteration. The main advantage of optimistic optimization is that one can specify the computational budget (e.g. the number of node expansions) in advance and guarantee bounds on the suboptimality with respect to the global optimum. Situations with a short control horizon and a long prediction horizon are rather common for DES control and it is useful to have a method to solve the corresponding MPC optimization problem without a significant influence of the prediction horizon. For a given MPL-MPC problem, the method using optimistic optimization in this chapter will be more efficient than the MILP method in the case of small control horizons and large prediction horizons.

Sometimes discrete control variables for MPL systems are required in practice. For example, for a manufacturing system it could happen that the raw materials are required to be fed to the manufacturing cell at 1 or 2 hours intervals; or for a railway network the departure times of trains might only be selected as multiples of 5 minutes. These constraints lead to discrete variables. In the optimal control problem given in this chapter, the objective function is a sum of discounted stage costs over an infinite horizon. Our goal is then to design a control sequence optimizing the infinite-horizon discounted objective function. The approach in this chapter is based on optimistic planning algorithms introduced below.

Optimistic planning is a class of planning algorithms originating in artificial intelligence applying the ideas of optimistic optimization [101]. This class of algorithms works for discrete-time systems with general nonlinear (deterministic or stochastic) dynamics and discrete control actions. Based on the current system state, a control sequence is obtained by optimizing an infinite-horizon sum of discounted bounded stage costs (or the expectation of these costs for the stochastic case). Optimistic planning uses a receding-horizon scheme and provides a characterization of the relationship between the computational budget and near-optimality. In optimistic planning for MPL systems with discrete control variables, we consider an infinite-horizon discounted objective function, which is more flexible than selecting a fixed finite-horizon objective function since the prediction horizon does not have to be fixed a priori. The length of the returned control sequence varies depending on the computational budget, the complexity of the problem, and the discount factor. Based on the standard geometric series, discounting is a simple way to obtain finite values for the total sum of stage costs over an infinite horizon. This is very convenient for comparing different infinite-length control sequences.

This chapter is organized as follows. In Section 3.2, the MPC problem for MPL systems with continuous control variables is presented. Generalized expressions of objective functions are given and a simple bound constraint on the input rate is considered. An optimistic optimization algorithm is applied to solve the proposed problem. In Section 3.3, we consider the infinite-horizon optimal control problem for MPL systems where the action space is discretized as a finite set. The objective function consists of a sum of discounted stage costs over an infinite horizon. We use an optimistic planning algorithm to address the considered problem. Finally, the chapter ends with concluding remarks and future research ideas.

3.2 Optimistic optimization for MPC of MPL systems with continuous control variables

Consider the following MPL system

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k), \tag{3.1}$$

$$y(k) = C \otimes x(k), \tag{3.2}$$

where the index k is the event counter, $x(k) \in \mathbb{R}_\varepsilon^{n_x}$ is the state, $u(k) \in \mathbb{R}_\varepsilon^{n_u}$ is the input, $y(k) \in \mathbb{R}_\varepsilon^{n_y}$ is the output, and where $A \in \mathbb{R}_\varepsilon^{n_x \times n_x}$, $B \in \mathbb{R}_\varepsilon^{n_x \times n_u}$, and $C \in \mathbb{R}_\varepsilon^{n_y \times n_x}$ are the system matrices. Let $y(k+j)$, $j = 1, 2, \ldots$ be the estimate of the output at event step $k+j$ based on the information available at event step k. Given a prediction horizon N_p, the estimation of the evolution of the MPL system from event step $k+1$ up to $k+N_\mathrm{p}$ can be presented as follows

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k), \tag{3.3}$$

with

$$\tilde{y}(k) = [y^T(k+1) \; \ldots \; y^T(k+N_\mathrm{p})]^T,$$
$$\tilde{u}(k) = [u^T(k) \; \ldots \; u^T(k+N_\mathrm{p}-1)]^T,$$

for appropriate H, $g(k)$ (see Section 2.3.2 for details of H, $g(k)$).

3.2.1 Objective function

The MPC framework has been extended to MPL systems in [47]. The considered objective function J consists of the weighted sum of an output cost and an input cost:

$$J = J_\mathrm{out} + \lambda J_\mathrm{in},$$

where the scalar $\lambda > 0$ is the trade-off between the output cost and the input cost. In this section, as a generalization of the costs presented in [47], we consider four more general output objective functions.

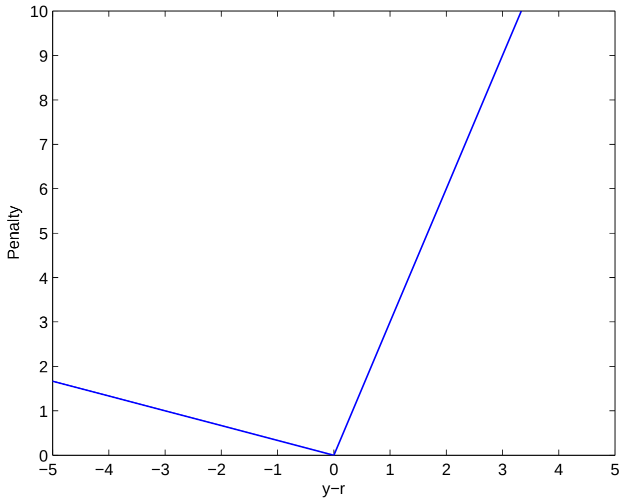


Figure 3.1: Output objective function $\max\big(\alpha(y-r),\beta(r-y)\big)$ *with* $\alpha=3,\ \beta=1/3$

Output objective function J_{out}

In a manufacturing system, a penalty may be incurred for every delay for urgent orders. In addition, an inventory cost may have to be paid for perishable goods. Therefore, we include both a tardiness and an earliness penalty in the output objective function with parameters to express the trade-off between the two kinds of penalties.

In particular, assume that r and y express the due date signal and completion time signal of the products respectively. Different penalty policies for the output objective function $\max(\alpha(y-r),\beta(r-y))$ can be achieved by choosing different parameters α,β with $\alpha,\beta\geq 0,\alpha+\beta>0$. If $\alpha>\beta$, the penalty for tardiness is higher than the one for earliness; if $\beta>\alpha$, the penalty for earliness is higher than the one for tardiness. Fig. 3.1 shows one specific penalty policy (with $\alpha>\beta$).

In this section, parameters α_l,β_l, $l=1,\ldots,n_y$ with $\alpha_l,\beta_l\geq 0$, $\alpha_l+\beta_l>0$ are introduced as weighting coefficients for the tardiness and earliness penalties with respect to a due date signal r. Denote

$$\Phi_{l,j,k}=\max\Big(\alpha_l\big(y_l(k+j)-r_l(k+j)\big),\beta_l\big(r_l(k+j)-y_l(k+j)\big)\Big),$$

where $y_l(k+j)$ is the l-th element of the estimate of the output at event step $k+j$, and $r_l(k+j)$ is the l-th element of the due date $r(k+j)$. Corresponding to the definitions of the 1-norm and the ∞-norm, we consider four different cases for J_{out}:

$$J_{\text{out}}^{1,1}(k)=\sum_{j=1}^{N_{\text{p}}}\sum_{l=1}^{n_y}\Phi_{l,j,k}\ , \qquad J_{\text{out}}^{1,\infty}(k)=\sum_{j=1}^{N_{\text{p}}}\max_{l=1,\ldots,n_y}\Phi_{l,j,k}\ ,$$

$$J_{\text{out}}^{\infty,1}(k)=\max_{j=1,\ldots,N_{\text{p}}}\sum_{l=1}^{n_y}\Phi_{l,j,k}\ , \qquad J_{\text{out}}^{\infty,\infty}(k)=\max_{j=1,\ldots,N_{\text{p}}}\max_{l=1,\ldots,n_y}\Phi_{l,j,k}\ .$$

It is easy to show that $J_{\text{out},1}$ and $J_{\text{out},2}$ in [47] are special cases of $J_{\text{out}}^{1,1}$.

Input objective function J_{in}

The control inputs of MPL systems often represent the time instants at which the control events occur, e.g., raw materials are fed into the production system. For the sake of just-in-time manufacturing and keeping low internal buffer level, it is better to maximize the control inputs. This is in contrast to the traditional case where the control input efforts should be minimized. Below are two objective functions that lead to the maximization of the inputs [47]:

$$J_{\mathrm{in}}^{1}(k) = -\sum_{j=1}^{N_{\mathrm{p}}}\sum_{s=1}^{n_u} u_s(k+j-1), \tag{3.4}$$

$$J_{\mathrm{in}}^{2}(k) = -\sum_{j=1}^{N_{\mathrm{p}}}\sum_{s=1}^{n_u} |u_s(k+j-1)|^2. \tag{3.5}$$

Relationship between $\tilde{u}$ and $\Delta\tilde{u}$

Since the control inputs correspond to the times of occurrence of input events, they are generally monotonically increasing. Hence, it is usually more convenient to consider the increments of the control inputs as control variables. Define the input rate

$$\Delta u(k) = u(k) - u(k-1).$$

In MPL-MPC, a control horizon N_{c} with $N_{\mathrm{c}} < N_{\mathrm{p}}$ is often introduced and the control input rate is taken to be constant from event step $k + N_{\mathrm{c}}$ on. Thus, the use of N_{c} reduces the computational burden. For an in-depth discussion about tuning of N_{c}, we refer the reader to [133]. Consequently, we assume

$$\Delta u(k+j) = \Delta u(k+N_{\mathrm{c}}-1),\ j = N_{\mathrm{c}},\ldots,N_{\mathrm{p}}-1.$$

Denote

$$\bar{u}(k-1) = \begin{bmatrix} u^T(k-1) & \cdots & u^T(k-1)\end{bmatrix}^T,$$

and define $L \in \mathbb{R}^{N_{\mathrm{p}} n_u \times N_{\mathrm{c}} n_u}$ as

$$L = \left[\begin{array}{ccccc} I_{n_u} & 0 & \cdots & 0 & 0 \\ I_{n_u} & I_{n_u} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ I_{n_u} & I_{n_u} & \cdots & I_{n_u} & I_{n_u} \\ \hline I_{n_u} & I_{n_u} & \cdots & I_{n_u} & 2I_{n_u} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ I_{n_u} & I_{n_u} & \cdots & I_{n_u} & (N_{\mathrm{p}}-N_{\mathrm{c}}+1)I_{n_u} \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\0\\\vdots\\0\end{array}}\right\} N_{\mathrm{c}} n_u \text{ rows} \\ \left.\vphantom{\begin{array}{c}0\\\vdots\\0\end{array}}\right\} (N_{\mathrm{p}}-N_{\mathrm{c}}) n_u \text{ rows} \end{array} \tag{3.6}$$

with I_{n_u} the $n_u \times n_u$ identity matrix. Then

$$\tilde{u}(k) = L\Delta\tilde{u}(k) + \bar{u}(k-1), \tag{3.7}$$

where

$$\Delta\tilde{u}(k) = \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k+N_\mathrm{c}-1) \end{bmatrix} = \begin{bmatrix} u(k) - u(k-1) \\ \vdots \\ u(k+N_\mathrm{c}-1) - u(k+N_\mathrm{c}-2) \end{bmatrix}.$$

Based on the definition of the objective function J, we know that J is a function of $\tilde{y}$ and $\tilde{u}$. Using (3.3) to eliminate $\tilde{y}$ from J_out, the eliminated J only depends on $\tilde{u}$ now. Then using (3.7) to replace $\tilde{u}$ by $\Delta\tilde{u}$, the resulting J only depends on $\Delta\tilde{u}$, denoted as J_Δ.

3.2.2 Constraints

Simple bound constraints on the input rate are common in practice, meaning that there is a minimum and a maximum separation between input events:

$$a \le \Delta\tilde{u}(k) \le b, \quad \text{for all } k, \tag{3.8}$$

with a, b real vectors of size $N_\mathrm{c} n_u \times 1$. The resulting feasible set is actually a hyperbox. Note that in practice the elements of a are always non-negative real values; hence, the constraint (2.3) is then automatically satisfied (namely the control input sequence is nondecreasing).

3.2.3 Problem formulation

Combining the material of previous subsections, we finally obtain the following MPL-MPC problem at event step k, for given $\sigma, \tau \in \{1, \infty\}$, $\omega \in \{1, 2\}$:

$$\min_{\Delta\tilde{u}(k)} J_\Delta^{\sigma,\tau,\omega}(k) = J_\mathrm{out}^{\sigma,\tau}(k) + \lambda J_\mathrm{in}^{\omega}(k) \tag{3.9}$$

subject to

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k), \tag{3.10}$$

$$\tilde{u}(k) = L\Delta\tilde{u}(k) + \bar{u}(k-1), \tag{3.11}$$

$$a \le \Delta\tilde{u}(k) \le b. \tag{3.12}$$

A finite optimal solution of the MPL-MPC problem (3.9)-(3.12) exists if the feasible set is bounded and closed and the objective function is finite for finite arguments. These conditions hold in general.

3.2.4 Optimistic optimization for the MPL-MPC problem

In this section, we adapt the deterministic optimistic optimization (DOO) algorithm for solving the MPL-MPC problem (3.9)-(3.12). More specifically, we develop a dedicated semi-metric ℓ that satisfies Assumptions 2.10-2.12 for the case that $\sigma = \tau = \omega = 1$, i.e.,

$$J_\Delta^{1,1,1} = J_\mathrm{out}^{1,1} + \lambda J_\mathrm{in}^{1} \;.$$

The expressions of ℓ for other cases can be derived similarly. Let

$$\mathcal{X} = \{\Delta\tilde{u}(k) | a \le \Delta\tilde{u}(k) \le b\},$$

be the hyperbox feasible set. Before proceeding further, we first present the following result.

Theorem 3.1 *Let* $\Delta\tilde{u}(k)$ *be an arbitrary input rate sequence vector and* $\Delta\tilde{u}^*(k)$ *be an optimal solution to problem* (3.9)-(3.12)*. Then it holds that*

$$\begin{aligned} & J_{\Delta}^{1,1,1}(\Delta\tilde{u}(k)) - J_{\Delta}^{1,1,1}(\Delta\tilde{u}^*(k)) \\ & \quad \le \sum_{l=1}^{n_y} \max(\alpha_l, \beta_l) \sum_{j=1}^{N_\mathrm{p}} \max_{i=1,\dots,jn_u} \left| L_{i,\cdot}(\Delta\tilde{u}(k) - \Delta\tilde{u}^*(k)) \right| + \lambda \left\| L(\Delta\tilde{u}(k) - \Delta\tilde{u}^*(k)) \right\|_1, \end{aligned} \tag{3.13}$$

where α_l, β_l *are as defined in Section 3.2.1 and* $L_{i,\cdot}$ *is the* i*-th row of* L *in* (3.6)*.*

Proof: Let $\tilde{u}(k)$ and $\tilde{u}^*(k)$ be the respective input sequence vectors corresponding to $\Delta\tilde{u}(k)$ and $\Delta\tilde{u}^*(k)$. Assume that $\tilde{y}(k)$ is the output sequence vector resulting from applying $\tilde{u}(k)$ to the system and $\tilde{y}^*(k)$ is the output sequence vector resulting from applying $\tilde{u}^*(k)$. Let

$$\tilde{l} = (j-1)n_y + l,$$

thus

$$\begin{aligned} \tilde{y}_{\tilde{l}}(k) &= y_l(k+j), \\ \tilde{y}^*_{\tilde{l}}(k) &= y^*_l(k+j), \\ \tilde{r}_{\tilde{l}}(k) &= r_l(k+j), \end{aligned}$$

for $l = 1,\dots,n_y$, $j = 1,\dots,N_\mathrm{p}$.

It is easy to verify that, for any $x, y, z \in \mathbb{R}$, we have

$$\max\big(\alpha(x-z), \beta(z-x)\big) - \max\big(\alpha(y-z), \beta(z-y)\big) \le \max(\alpha,\beta)|x-y|,$$

where α, β are non-negative real numbers. Hence, we have

$$J_{\mathrm{out}}^{1,1}(\Delta\tilde{u}(k)) - J_{\mathrm{out}}^{1,1}(\Delta\tilde{u}^*(k)) \le \sum_{j=1}^{N_\mathrm{p}} \sum_{l=1}^{n_y} \max(\alpha_l, \beta_l) \left| \tilde{y}_{\tilde{l}}(k) - \tilde{y}^*_{\tilde{l}}(k) \right|. \tag{3.14}$$

From (2.17), we have

$$\tilde{y}_{\tilde{l}}(k) = \max\big(H_{\tilde{l},\cdot} \otimes \tilde{u}(k),\ g_{\tilde{l}}(k)\big),$$

and

$$\tilde{y}^*_{\tilde{l}}(k) = \max\big(H_{\tilde{l},\cdot} \otimes \tilde{u}^*(k),\ g_{\tilde{l}}(k)\big),$$

where $H_{\tilde{l},\cdot}$ is the $\tilde{l}$-th row of H. Thus, we have

$$\left| \tilde{y}_{\tilde{l}}(k) - \tilde{y}^*_{\tilde{l}}(k) \right| \le \left| H_{\tilde{l},\cdot} \otimes \tilde{u}(k) - H_{\tilde{l},\cdot} \otimes \tilde{u}^*(k) \right|. \tag{3.15}$$

Denote

$$\begin{aligned} H_{\tilde{l},\cdot} \otimes \tilde{u}(k) &= \max_{w=1,\dots,jn_u} \big(H_{\tilde{l}w} + \tilde{u}_w(k)\big) \\ &= H_{\tilde{l}w_0} + \tilde{u}_{w_0}(k), \\ H_{\tilde{l},\cdot} \otimes \tilde{u}^*(k) &= \max_{z=1,\dots,jn_u} \big(H_{\tilde{l}z} + \tilde{u}^*_z(k)\big) \end{aligned}$$

$$
\begin{aligned}
&= H_{\tilde{l}z_0} + \tilde{u}^*_{z_0}(k) \\
&\geq H_{\tilde{l}w_0} + \tilde{u}^*_{w_0}(k).
\end{aligned}
\tag{3.16}
$$

Then we have

$$
\begin{aligned}
\left| H_{\tilde{l},\cdot} \otimes \tilde{u}(k) - H_{\tilde{l},\cdot} \otimes \tilde{u}^*(k) \right| &= \left| H_{\tilde{l}w_0} + \tilde{u}_{w_0}(k) - H_{\tilde{l}z_0} - \tilde{u}^*_{z_0}(k) \right| \\
&\overset{(3.16)}{\leq} \left| H_{\tilde{l}w_0} + \tilde{u}_{w_0}(k) - H_{\tilde{l}w_0} - \tilde{u}^*_{w_0}(k) \right| \\
&\leq \left| \tilde{u}_{w_0}(k) - \tilde{u}^*_{w_0}(k) \right| \\
&\leq \max_{i=1,\dots,jn_u} \left| \tilde{u}_i(k) - \tilde{u}^*_i(k) \right| \\
&\overset{(3.7)}{\leq} \max_{i=1,\dots,jn_u} \left| L_{i,\cdot}(\Delta\tilde{u}(k) - \Delta\tilde{u}^*(k)) \right|.
\end{aligned}
\tag{3.17}
$$

Therefore, from (3.14), (3.15) and (3.17), we have

$$
J^{1,1}_{\text{out}}(\Delta\tilde{u}(k)) - J^{1,1}_{\text{out}}(\Delta\tilde{u}^*(k)) \leq \sum_{l=1}^{n_y} \max(\alpha_i, \beta_i) \sum_{j=1}^{N_\text{p}} \max_{i=1,\dots,jn_u} \left| L_{i,\cdot}(\Delta\tilde{u} - \Delta\tilde{u}^*) \right|.
\tag{3.18}
$$

On the other hand,

$$
\begin{aligned}
J^1_{\text{in}}(\Delta\tilde{u}(k)) - J^1_{\text{in}}(\Delta\tilde{u}^*(k)) &= -\left[\sum_{j=1}^{N_\text{p}} \sum_{s=1}^{n_u} u_s(k+j-1) - \sum_{j=1}^{N_\text{p}} \sum_{s=1}^{n_u} u^*_s(k+j-1) \right] \\
&= \sum_{j=1}^{N_\text{p}} \sum_{s=1}^{n_u} \left[u^*_s(k+j-1) - u_s(k+j-1) \right] \\
&= \sum_{i=1}^{N_\text{p} n_u} \left[\tilde{u}^*_i(k) - \tilde{u}_i(k) \right] \\
&\leq \| \tilde{u}(k) - \tilde{u}^*(k) \|_1 \\
&\leq \| L(\Delta\tilde{u}(k) - \Delta\tilde{u}^*(k)) \|_1.
\end{aligned}
\tag{3.19}
$$

From (3.18)-(3.19), we deduce that (3.13) holds. □

According to Theorem 3.1, we can define $\ell^{1,1,1} : \mathscr{X} \times \mathscr{X} \to \mathbb{R}^+$, such that for any $\Delta\tilde{u}(k), \Delta\tilde{v}(k) \in \mathscr{X}$,

$$
\ell^{1,1,1}(\Delta\tilde{u}(k), \Delta\tilde{v}(k)) \triangleq \ell^{1,1}_{\text{out}}(\Delta\tilde{u}(k), \Delta\tilde{v}(k)) + \lambda \ell^1_{\text{in}}(\Delta\tilde{u}(k), \Delta\tilde{v}(k)),
\tag{3.20}
$$

with

$$
\begin{aligned}
&\ell^{1,1}_{\text{out}}(\Delta\tilde{u}(k), \Delta\tilde{v}(k)) = \sum_{l=1}^{n_y} \max(\alpha_l, \beta_l) \sum_{j=1}^{N_\text{p}} \max_{i=1,\dots,jn_u} \left| L_{i,\cdot}(\Delta\tilde{u}(k) - \Delta\tilde{v}(k)) \right|, \\
&\ell^1_{\text{in}}(\Delta\tilde{u}(k), \Delta\tilde{v}(k)) = \| L(\Delta\tilde{u}(k) - \Delta\tilde{v}(k)) \|_1,
\end{aligned}
$$

where $\lambda > 0$ and α_l, β_l are as defined in Section 3.2.1. Because L is not singular, it is easy to verify that the function $\ell^{1,1,1}$ defined by (3.20) is a semi-metric on $\mathscr{X}$. Therefore, Assumption 2.10 are satisfied for $\sigma = \tau = \omega = 1$.

Regarding the partitioning of $\mathscr{X} = \{\Delta\tilde{u}(k) | a \leq \Delta\tilde{u}(k) \leq b\}$, we take the center of $\mathscr{X}$ as the starting point (corresponding to the root node of the tree). At each iteration, we bisect each dimension of $\mathscr{X}$; so the number of branches K equals $2^{N_\mathrm{c} n_u}$. From (3.8), for any $\Delta\tilde{u}(k) \in X^{h,d}$ where $X^{h,d}$ is a cell at depth h with node index d and is characterized by its center $\Delta\tilde{u}^{h,d}(k)$, we have

$$\|\Delta\tilde{u}(k) - \Delta\tilde{u}^{h,d}(k)\|_\infty \leq \frac{1}{2^{h+1}} \|b-a\|_\infty \ , \tag{3.21}$$

$$\|\Delta\tilde{u}(k) - \Delta\tilde{u}^{h,d}(k)\|_1 \leq \frac{1}{2^{h+1}} \|b-a\|_1 \ . \tag{3.22}$$

Based on the proposed expression for $\ell^{1,1,1}$, we now derive the expressions for $\delta(h)$ and ν appearing in Assumptions 2.11-2.12. Corresponding to the superscript of $\ell^{1,1,1}$, the derived $\delta(h)$ and ν will be written as $\delta^{1,1,1}(h)$ and $\nu^{1,1,1}$.

Theorem 3.2 *Define*

$$\delta^{1,1,1}(h) = \frac{1}{2^{h+1}} \left[\delta^{1,1}_{\mathrm{out}} + \lambda \delta^{1}_{\mathrm{in}}\right], \tag{3.23}$$

for $h \in \{0, 1, \ldots\}$ *with*

$$\delta^{1,1}_{\mathrm{out}} = \frac{N_\mathrm{p}(N_\mathrm{p}+1)\|b-a\|_\infty}{2} \sum_{l=1}^{n_y} \max(\alpha_l, \beta_l), \tag{3.24}$$

$$\delta^{1}_{\mathrm{in}} = \|L(b-a)\|_1 \ . \tag{3.25}$$

Then for any $h \in \{0, 1, \ldots\}$, $d \in \{0, \ldots, K^h - 1\}$, *it holds that*

$$\sup_{\Delta\tilde{u}(k) \in X^{h,d}} \ell^{1,1,1}(\Delta\tilde{u}(k), \Delta\tilde{u}^{h,d}(k)) \leq \delta^{1,1,1}(h),$$

where $\Delta\tilde{u}^{h,d}(k)$ *is the center of the cell* $X^{h,d}$.

Proof: For any $\Delta\tilde{u}(k) \in X^{h,d}$, we have[1]

$$\begin{aligned}
\ell^{1,1}_{\mathrm{out}}(\Delta\tilde{u}(k), \Delta\tilde{u}^{h,d}(k)) &= \sum_{l=1}^{n_y} \max(\alpha_l, \beta_l) \sum_{j=1}^{N_\mathrm{p}} \max_{i=1,\ldots,j n_u} \left| L_{i,\cdot}(\Delta\tilde{u}(k) - \Delta\tilde{u}^{h,d}(k)) \right| \\
&\leq \sum_{l=1}^{n_y} \max(\alpha_l, \beta_l) \left[\sum_{j=1}^{N_\mathrm{p}} \max_{i=1,\ldots,j n_u} \|L_{i,\cdot}\|_1 \right] \|\Delta\tilde{u}(k) - \Delta\tilde{u}^{h,d}(k)\|_\infty \\
&\overset{(3.21)}{\leq} \sum_{l=1}^{n_y} \max(\alpha_l, \beta_l) \frac{N_\mathrm{p}(N_\mathrm{p}+1)}{2} \frac{\|b-a\|_\infty}{2^{h+1}} \\
&\overset{(3.24)}{\leq} \frac{1}{2^{h+1}} \delta^{1,1}_{\mathrm{out}} \ ,
\end{aligned}$$

and

$$\begin{aligned}
\ell^{1}_{\mathrm{in}}(\Delta\tilde{u}(k), \Delta\tilde{u}^{h,d}(k)) &= \|L(\Delta\tilde{u}(k) - \Delta\tilde{u}^{h,d}(k))\|_1 \\
&\overset{(3.22)}{\leq} \frac{1}{2^{h+1}} \|L(b-a)\|_1
\end{aligned}$$

[1] Note that for every $x, y \in \mathbb{R}^n$, we have $|x^T y| \leq \|x\|_1 \|y\|_\infty$ [62].

$$\overset{(3.25)}{\leq} \frac{1}{2^{h+1}}\delta^{1}_{\mathrm{in}} \ .$$

Thus if we define $\delta^{1,1,1}(h)$ as in (3.23), then

$$\sup_{\Delta\tilde{u}(k)\in X^{h,d}} \ell^{1,1,1}(\Delta\tilde{u}(k),\Delta\tilde{u}^{h,d}(k)) \leq \delta^{1,1,1}(h).$$

□

Theorem 3.3 *Choose $\nu^{1,1,1}$ such that*

$$0 < \nu^{1,1,1} \leq \frac{\rho \min\limits_{i=1,\ldots,N_{\mathrm{c}}n_u}(b_i - a_i)}{\delta^{1,1}_{\mathrm{out}} + \lambda\delta^{1}_{\mathrm{in}}} \ .$$

Then any cell $X^{h,d}$ at any depth h contains an ℓ-ball $\mathscr{B}^{h,d}$ of radius $\nu^{1,1,1}\delta^{1,1,1}(h)$ centered in $\Delta\tilde{u}^{h,d}$ where $0<\rho<1$ and $\delta^{1,1,1}(h)$, $\delta^{1,1}_{\mathrm{out}}$, and δ^{1}_{in} are as defined in (3.23)-(3.25).

Proof: According to Theorem 3.2, we can define a decreasing sequence $\{\delta^{1,1,1}(h)\}_{h=0}^{\infty}$ as in (3.23). Select a real number ρ with $0 < \rho < 1$. From (3.8), the ℓ-ball $\mathscr{B}^{h,d}$ of radius $\nu^{1,1,1}\delta^{1,1,1}(h)$ centered in $\Delta\tilde{u}^{h,d}$ is inside the cell $X^{h,d}$, if we choose $\nu^{1,1,1}$ such that

$$\nu^{1,1,1}\delta^{1,1,1}(h) \leq \rho\frac{\min\limits_{i=1,\ldots,N_{\mathrm{c}}n_u}(b_i - a_i)}{2^{h+1}} \ .$$

Then $\nu^{1,1,1}$ can be chosen as

$$\begin{aligned}\nu^{1,1,1} &\leq \frac{\rho \min\limits_{i=1,\ldots,N_{\mathrm{c}}n_u}(b_i - a_i)}{2^{h+1}\delta^{1,1,1}(h)}\\ &\leq \frac{\rho \min\limits_{i=1,\ldots,N_{\mathrm{c}}n_u}(b_i - a_i)}{\delta^{1,1}_{\mathrm{out}} + \lambda\delta^{1}_{\mathrm{in}}} \ .\end{aligned}$$

□

Up to now, we have proved that Assumptions 2.10-2.12 are satisfied for $\sigma=\tau=\omega=1$. In a similar way, we can obtain corresponding results for other cases. The analytic expressions for ℓ and $\delta(h)$ with different σ,τ,ω are presented as follows.

1) $\sigma=1$, $\tau=\infty$

$$\ell^{1,\infty}_{\mathrm{out}}(\Delta\tilde{u}(k),\Delta\tilde{v}(k)) = \max_{l=1,\ldots,n_y}\max(\alpha_l,\beta_l)\sum_{j=1}^{N_{\mathrm{p}}}\max_{i=1,\ldots,jn_u}\left|L_{i,\cdot}(\Delta\tilde{u}(k)-\Delta\tilde{v}(k))\right|,$$

$$\delta^{1,\infty}_{\mathrm{out}} = \frac{N_{\mathrm{p}}(N_{\mathrm{p}}+1)\|b-a\|_\infty}{2}\max_{l=1,\ldots,n_y}\max(\alpha_l,\beta_l) \ .$$

2) $\sigma=\infty$, $\tau=1$

$$\ell^{\infty,1}_{\mathrm{out}}(\Delta\tilde{u}(k),\Delta\tilde{v}(k)) = \sum_{l=1}^{n_y}\max(\alpha_l,\beta_l)\max_{j=1,\ldots,N_{\mathrm{p}}}\max_{i=1,\ldots,jn_u}\left|L_{i,\cdot}(\Delta\tilde{u}(k)-\Delta\tilde{v}(k))\right|,$$

$$\delta_{\text{out}}^{\infty,1} = N_{\text{p}} \|b-a\|_\infty \sum_{l=1}^{n_y} \max(\alpha_l, \beta_l) \quad .$$

3) $\sigma = \infty$, $\tau = \infty$

$$\ell_{\text{out}}^{\infty,\infty}(\Delta\tilde{u}(k), \Delta\tilde{v}(k)) = \max_{l=1,\dots,n_y} \max(\alpha_l, \beta_l) \max_{j=1,\dots,N_{\text{p}}} \max_{i=1,\dots,jn_u} \left| L_{i,\cdot}(\Delta\tilde{u}(k) - \Delta\tilde{v}(k)) \right|,$$
$$\delta_{\text{out}}^{\infty,\infty} = N_{\text{p}} \|b-a\|_\infty \max_{l=1,\dots,n_y} \max(\alpha_l, \beta_l) \quad .$$

4) $\omega = 2$

$$\ell_{\text{in}}^{2}(\Delta\tilde{u}(k), \Delta\tilde{v}(k)) = 2 \, \|Lb + \bar{u}(k-1)\|_2 \, \|L(\Delta\tilde{u}(k) - \Delta\tilde{v}(k))\|_2 \quad ,$$
$$\delta_{\text{in}}^{2} = 2 \, \|Lb + \bar{u}(k-1)\|_2 \, \|L(b-a)\|_2 \quad .$$

Remark 3.4 The computational complexity of DOO in our implementation is exponential in the control horizon N_{c}. On the other hand, the MPL-MPC problem can also be formulated as an MILP problem [8, 47]. The number of auxiliary binary variables that are used to convert the max operator into linear equations is proportional to the prediction horizon N_{p}. As a result, the complexity of state-of-the-art MILP algorithms is in the worst case exponential in N_{p} [123]. Therefore, DOO will be more efficient if $N_{\text{c}} \ll N_{\text{p}}$. □

3.2.5 Examples

In this section, we illustrate the proposed approach and the statement in Section 3.2.4 for MPL systems with randomly selected system matrices and for an industrial manufacturing system. All experiments are implemented in Matlab 2014b on an 3.1 GHz processor with 3.7 GB RAM.

Example 1: Random systems

Consider the MPL system (3.1)-(3.2) with $n_u = n_y = 1$. We will consider $n_x = 5, 10, 20$ and perform experiments for $N_{\text{c}} = 3, 4, 5$ and $N_{\text{p}} = N_{\text{c}} + 1, \dots, 60$. Assume that $\lambda = 0.01$, $u(-1) = 0$, $\sigma = \tau = \omega = 1$, and $-15 \le \Delta u(k) \le 15$ for all k. The elements of $A, B, C, x(0)$ are selected as random integers uniformly distributed in the interval $[0, 10]$, but some elements of $A, B, C, x(0)$ may be equal to ε with a probability 0.2. The increments of the reference sequence r are random integers uniformly distributed in the interval $[0, 10]$. For each $n_x \in \{5, 10, 20\}$, we generate 20 random $(A, B, C, x(0))$ combinations. For each choice of $(A, B, C, x(0))$, we generate 10 random reference sequences $\{r(k)\}_{k=1}^{N_{\text{p}}}$. The computational budget of DOO is set to 200 node expansions.

We compare the efficiency of our method with the MILP solvers [4] `cplex` and `glpk` for solving the problem (3.9)-(3.12). This comparison is fair because our method and the MILP solvers are all implemented in object code and called from Matlab. We specifically use the `cplex` solver inside the Tomlab toolbox of Matlab and the `glpk` solver called through the glpkmex interface.

The CPU time for each method is plotted using a logarithmic scale in Figure 3.2. We can see that, in Figure 3.2(a), for $N_{\text{c}} = 3$, the mean CPU time curves of DOO (`oo`) and the MILP solvers intersect at $N_{\text{p}} = 6$. For $N_{\text{c}} = 4$ and $N_{\text{c}} = 5$, the intersections of the mean CPU time curves for `oo` and for the MILP solvers occur at respectively $N_{\text{p}} = 9$ and $N_{\text{p}} = 14$ as shown in

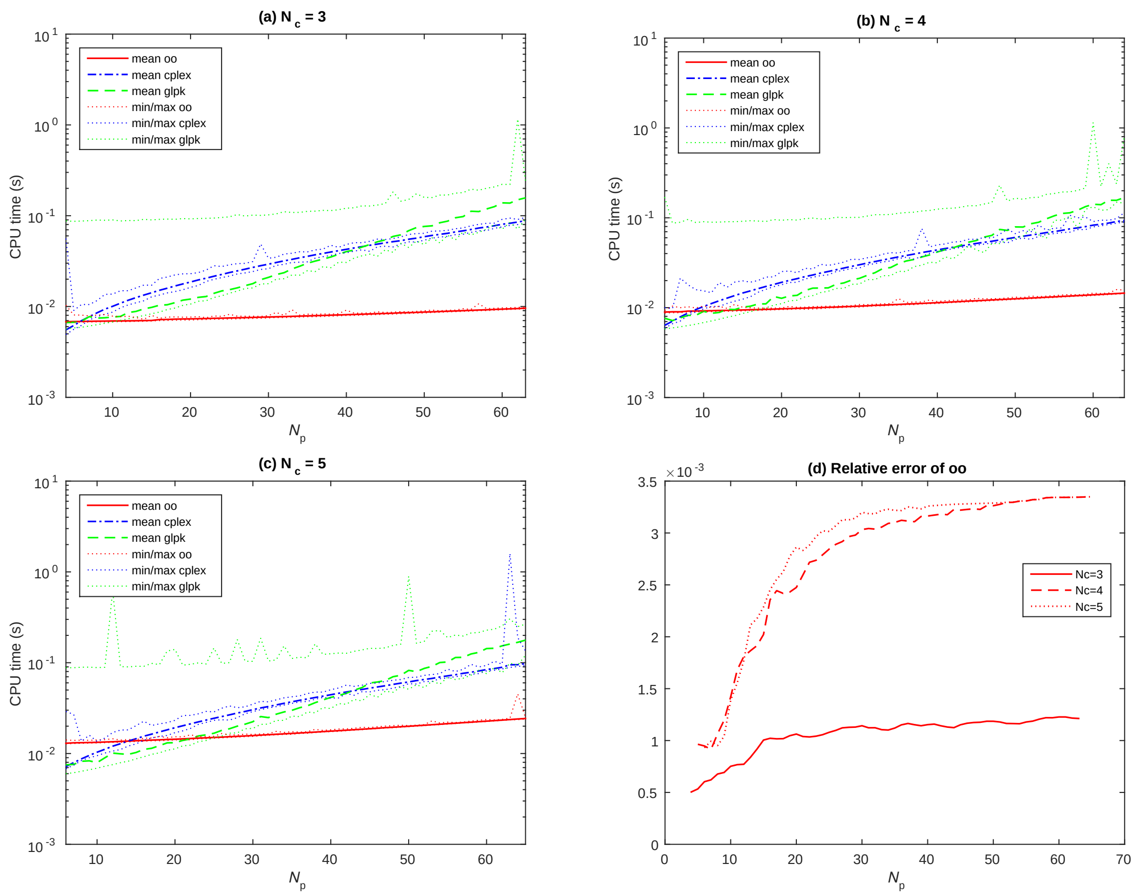


Figure 3.2: (a-c) The CPU time for DOO (`oo`), `cplex`, and `glpk` for $N_c = 3,4,5$*; (d) Relative error between `oo` and the MILP solvers.*

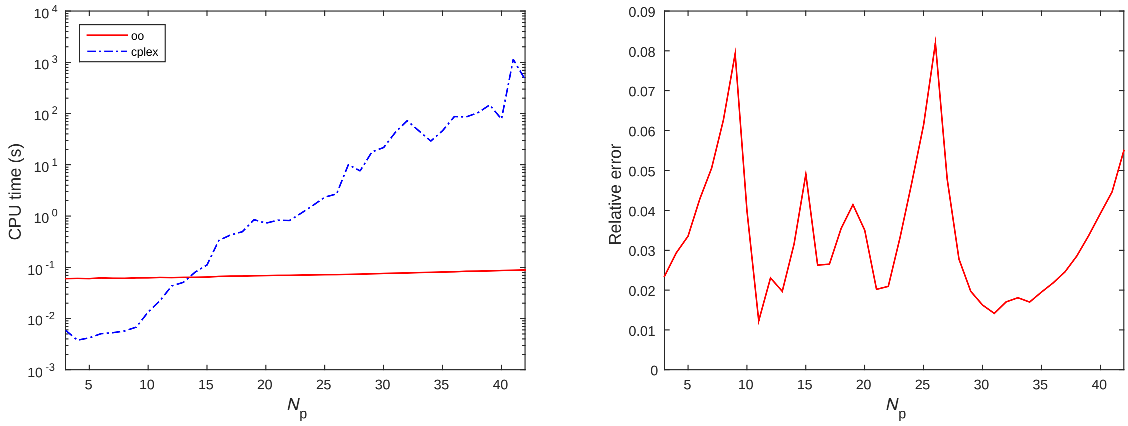


Figure 3.3: (a) The CPU time for DOO (`oo`) and `cplex`; (b) Relative error between `oo` and `cplex`.

Figure 3.2(b-c). Thus DOO is faster than MILP when N_p is about two or three times as large as N_c. We can also see that the computation time of the MILP solvers is exponential in N_p, while N_p has no significant influence on the computation time of DOO.

We also compute the relative error between the objective function value obtained by DOO and the best value among the two MILP solvers (see Figure 3.2(d)). The difference between the objective function values provided by the two MILP solvers are negligible, so it is not plotted. For each n_x and each combination of $A, B, C, x(0)$ and $r(k)$, the relative error of DOO is computed. The plotted relative errors are the average values over all instances. We can see that for each value of N_c considered, the average relative errors are less than 3.5×10^{-3}.

Example 2: Industrial manufacturing system

Now we consider the manufacturing unit for producing rubber tubes for automobile equipment presented in [5]. The dynamic behavior of this system is described by an MPL system with 19 states, 2 inputs, and 1 output (see [5] for details). Let $N_\mathrm{c} = 2$. We run experiments for $N_\mathrm{p} = N_\mathrm{c} + 1, \ldots, 40$ with $\lambda = 0.0001$, $\sigma = \tau = \omega = 1$, $u(-1) = [0\ 0]^T$ and $2 \leq \Delta u_i(k) \leq 8$, $i = 1, 2$, for all k. The increments of the reference sequence r are random integers uniformly distributed in the interval $[2, 10]$. We use DOO and the `cplex` solver to solve the corresponding MPL-MPC problem[2]. The computational budget of DOO is set to 700 node expansions. Figure 3.3 shows the CPU time for DOO and `cplex` and the relative error between the values of $J_\Delta^{1,1,1}$ provided by both methods. We can see that DOO is faster than `cplex` when $N_\mathrm{p} \geq 14$ and that the relative error between the objective function values is less than 9%.

[2]The `glpk` solver is not used for comparison because `cplex` is much faster than `glpk` when solving the resulting MILP problem for this example.

3.3 Optimistic planning for MPL systems with discrete control variables

In the previous section, we have used optimistic optimization to solve the finite-horizon optimal control problem for MPL systems with continuous control inputs. In this section, we propose to apply optimistic planning to solve the infinite-horizon optimal control problem for MPL systems where the action space is discretized as a finite set. Note that although the evolution of MPL systems is event-driven in contrast to time-driven as in a discrete-time system, optimistic planning can still be applied because of the analogy between descriptions of MPL systems and conventional linear time-invariant discrete-time systems. Also note that considering an infinite-horizon discounted objective function is more flexible than selecting a fixed finite-horizon objective function since the prediction horizon does not have to be fixed a priori.

3.3.1 Problem statement

Consider the following MPL system

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k), \tag{3.26}$$

$$y(k) = C \otimes x(k), \tag{3.27}$$

where k is the event counter, $x(k) \in \mathbb{R}_\varepsilon^{n_x}$ is the state, $u(k) \in \mathbb{R}_\varepsilon^{n_u}$ is the input, $y(k) \in \mathbb{R}_\varepsilon^{n_y}$ is the output and where the input $u(k)$ is rewritten as

$$u(k) = u(k-1) + \Delta u(k).$$

For the sake of simplicity, we consider the single input case (i.e., $n_u = 1$); however, an extension to multiple inputs can be made. We assume that the increments $\Delta u(k)$ of the inputs take values from a given finite set $U \triangleq \{u^1, \ldots, u^K\}$ with K the number of actions and with $u^i \geq 0$ for all i, and where U is called the action space.

Given a due date signal $\{r(k)\}_{k=0}^{\infty}$ with $r(k) \in \mathbb{R}^{n_y}$, a typical objective in optimal control for MPL systems is minimizing the due date error between the output event times and the due dates, e.g., the tardiness values $\max(y_l(k) - r_l(k), 0)$. So we consider the following stage cost:

$$\rho(k) = \sum_{l=1}^{n_y} \min\big(\max(y_l(k) - r_l(k), 0), g\big) + \lambda F(\Delta u(k)), \tag{3.28}$$

where $\lambda > 0$ is a trade-off between the delay of completion times with respect to the due date signal and the feeding rate. The positive scalar g is introduced to make $\rho(k)$ bounded, more specifically, g is a predefined value larger than $y_l(k) - r_l(k)$ for any l and k. For each element u^i of the finite set U, we assign a cost $F(u^i)$ according to some criterion. In addition, we make the following assumption in this section.

Assumption 3.5 *For any $i \in \{1, \ldots, K\}$, we have $F(u^i) \leq g$.*

If we consider a just-in-time setting, then the smaller the value of $\Delta u(k)$, the larger the value of its cost, i.e., F should be a positive monotonically nonincreasing function of $\Delta u(k)$. For example, assume that $U = \{u^1, u^2\}$, i.e., the next feeding time is after u^1 or u^2 time units and assume that $u^1 < u^2$, then we could have

$$F(\Delta u(k)) = \alpha_i g \quad \text{if } \Delta u(k) = u^i,$$

with $\alpha_1 > \alpha_2 > 0$ and $\alpha_1 + \alpha_2 = 1$. Another example could be:

$$F(\Delta u(k)) = g - \Delta u(k), \quad \text{with } g \geq \max(U).$$

It is easy to verify that $\rho(k)$ always belongs to the interval $[0, g + \lambda g]$.

Given initial conditions $x(0)$ and $u(-1)$, define an infinite-length control sequence $\Delta \boldsymbol{u} = (\Delta u(0), \Delta u(1), \ldots)$ and the infinite-horizon discounted objective function corresponding to $\Delta \boldsymbol{u}$:

$$J(\Delta \boldsymbol{u}, x(0), u(-1)) = \sum_{k=0}^{\infty} \gamma^k \rho(k+1).$$

Note that we have $J(\Delta \boldsymbol{u}, x(0), u(-1)) \in [0, \frac{g+\lambda g}{1-\gamma}]$ providing that Assumption 3.5 holds.

The infinite-horizon discounted optimal control problem for MPL systems with discrete inputs is now defined as follows:

$$\min_{\Delta \boldsymbol{u}} J(\Delta \boldsymbol{u}, x(0), u(-1)) \tag{3.29}$$

subject to

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k), \tag{3.30}$$

$$y(k) = C \otimes x(k), \tag{3.31}$$

$$u(k) = u(k-1) + \Delta u(k), \tag{3.32}$$

$$\Delta u(k) \in U \triangleq \{u^1, \ldots, u^K\}, \quad k = 0, 1, \ldots. \tag{3.33}$$

Note that (2.3) is automatically satisfied since $u^i \geq 0$ for all i.

3.3.2 Optimistic planning for MPL systems

In this section, we apply optimistic planning of deterministic systems (OPD) to solve the infinite-horizon discounted optimal control problem (3.29)-(3.33). We first define lower and upper bound functions similar to (2.24) and (2.25). The bounded stage cost (3.28) corresponds to a bounded reward function:

$$R(k) = 1 - \frac{\rho(k)}{g + \lambda g} \quad . \tag{3.34}$$

Furthermore, $R(k) \in [0, 1]$. Define

$$\bar{J}(\Delta \boldsymbol{u}, x(0), u(-1)) = \sum_{k=0}^{\infty} \gamma^k R(k+1). \tag{3.35}$$

The minimization problem (3.29)-(3.33) can now be translated into the following maximization problem:

$$\max_{\Delta \boldsymbol{u}} \bar{J}(\Delta \boldsymbol{u}, x(0), u(-1)) \tag{3.36}$$

subject to

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k), \tag{3.37}$$
$$y(k) = C \otimes x(k), \tag{3.38}$$
$$u(k) = u(k-1) + \Delta u(k), \tag{3.39}$$
$$\rho(k) = \sum_{l=1}^{n_y} \min\big(\max(y_l(k) - r_l(k), 0), g\big) + \lambda F(\Delta u(k)), \tag{3.40}$$
$$R(k) = 1 - \frac{\rho(k)}{g + \lambda g}, \tag{3.41}$$
$$\Delta u(k) \in U \triangleq \{u^1, \ldots, u^K\}, \quad k = 0, 1, \ldots. \tag{3.42}$$

Define

$$\Delta \boldsymbol{u}_d = (\Delta u(0), \ldots, \Delta u(d-1)),$$
$$v(\Delta \boldsymbol{u}_d) = \sum_{k=0}^{d-1} \gamma^k R(k+1),$$
$$b(\Delta \boldsymbol{u}_d) = v(\Delta \boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma}.$$

So $v(\Delta \boldsymbol{u}_d)$ and $b(\Delta \boldsymbol{u}_d)$ provide lower and upper bounds of $\bar{J}(\Delta \boldsymbol{u}|_d, x(0), u(-1))$ for any infinite-length sequence $\Delta \boldsymbol{u}|_d$ of which the first d components coincide with $\Delta \boldsymbol{u}_d$. When applying OPD to solve the problem (3.36)-(3.42), the upper bound function b is used to select the most promising control sequence (corresponding to the largest b-value among all leaves of the current tree) to expand. The lower bound function v is used for determining the best control sequence at the end of the algorithm.

Given initial conditions $x(0)$ and $u(-1)$, a due date signal $\{r(k)\}_{k=0}^{\infty}$, and the number of node expansions n, OPD returns a control sequence $\Delta \boldsymbol{u}^*_{d'}$ that maximizes the lower bound v function. The first action of $\Delta \boldsymbol{u}^*_{d'}$ is applied to the system and the whole process is repeated at each event step. In this way, a receding-horizon controller is obtained. The length d' of the returned sequence is the maximum depth reached by the algorithm for the given finite n. According to Theorem 2.17(i) (see also [22]), we have the following corollary for the near-optimality guarantee of the returned control sequence:

Corollary 3.6 *Let*

$$\bar{J}^*(x(0), u(-1)) \triangleq \max_{\Delta \boldsymbol{u}} \bar{J}(\Delta \boldsymbol{u}, x(0), u(-1)),$$

be the optimal value of the objective function in problem (3.36)-(3.42). *Let* $\Delta \boldsymbol{u}^*|_{d'}$ *be any infinite-length sequence of which the first* d' *components coincide with* $\Delta \boldsymbol{u}^*_{d'}$ *returned by OPD. Then we have*

$$\begin{aligned}&\bar{J}^*(x(0), u(-1)) - \bar{J}(\Delta \boldsymbol{u}^*|_{d'}, x(0), u(-1))\\ &\quad\le b(\Delta \boldsymbol{u}^*_{d'}) - v(\Delta \boldsymbol{u}^*_{d'})\\ &\quad\le \frac{\gamma^{d'}}{1-\gamma}.\end{aligned}$$

□

OPD applies the first component of $\Delta \boldsymbol{u}^*_{d'}$ to the system and generates a new control sequence at the next event step. Rather than recomputing a new control sequence at every event step, one can alternatively apply the first subsequence of length $\bar{d}$ of $\Delta \boldsymbol{u}^*_{d'}$ (with $\bar{d} \leq d'$) to the system and recompute the control sequence only every $\bar{d}$ event steps [22]. Recall that d' is the maximum depth reached by the algorithm for the fixed n. In order to obtain a control sequence with a sufficient length, the number of node expansions n should be large enough such that the length of the returned sequence $\Delta \boldsymbol{u}^*_{d'}$ is at least $\bar{d}$. In the worst case, the algorithm will explore all branches of the tree, so n should be larger than $\sum_{k=0}^{\bar{d}-1} K^k + 1$ to guarantee that at least one path has length $\bar{d}$. However, in general a smaller n can be selected because OPD explores the tree in an efficient way rather than evaluating all actions in the action space at each node expansion step.

3.3.3 Example

Consider the following MPL system from [103]:

$$x(k+1) = \begin{bmatrix} \varepsilon & 0 & \varepsilon & 9 \\ 4 & 3 & 4 & 5 \\ 8 & \varepsilon & 2 & 8 \\ 0 & 1 & \varepsilon & \varepsilon \end{bmatrix} \otimes x(k) \oplus \begin{bmatrix} 0 \\ 5 \\ 2 \\ 8 \end{bmatrix} \otimes u(k), \tag{3.43}$$

$$y(k) = \begin{bmatrix} 6 & 5 & 8 & \varepsilon \end{bmatrix} \otimes x(k). \tag{3.44}$$

Given a due date signal $r(k) = 50 + 6.5k$, and the initial conditions $x(0) = \begin{bmatrix} 6 & 12 & 9 & 14 \end{bmatrix}^T$ and[3] $u(-1) = 6$, we consider the following stage cost function

$$\rho(k) = \min\big(\max(y(k) - r(k), 0), g\big) + \lambda(g - \Delta u(k)), \tag{3.45}$$

with $g = 500$, $\lambda = 0.001$, $\Delta u(k) \in U = \{6, 8\}$, $K = 2$. The discount factor in (3.35) is $\gamma = 0.95$.

The optimistic planning based approach is implemented to obtain a receding-horizon controller for the MPL system (3.43)-(3.44). In addition, a finite-horizon approach is also implemented for comparison. More specifically, given a fixed finite horizon $d_\mathrm{N} = 10$, a full tree[4] is explored from the root node to the depth d_N. The finite-horizon approach returns a control sequence that maximizes the following function

$$\bar{J}_\mathrm{N} = \sum_{k=0}^{d_\mathrm{N}-1} \gamma^k R(k+1),$$

where R is the reward corresponding to (3.45).

The difference between y and r is used for comparing the optimistic planning approach and the finite-horizon approach. For each approach, we consider both applying the first action only and applying a subsequence of length $\bar{d}$ to the system once an optimal control sequence is obtained. Fig. 3.4 shows the results of applying the first action only with $n = 100$ for the optimistic planning approach and with $d_\mathrm{N} = 10$ for the finite-horizon approach.

[3] Considering a production system, the initial state $x(0)$ contains the starting times of the production processes for the 0-th cycle. The initial input $u(-1)$ represents the time at which a batch of raw material is fed to the system for the 0-th cycle.

[4] Here a full tree is a tree in which every node that is not a leaf node, has K children.

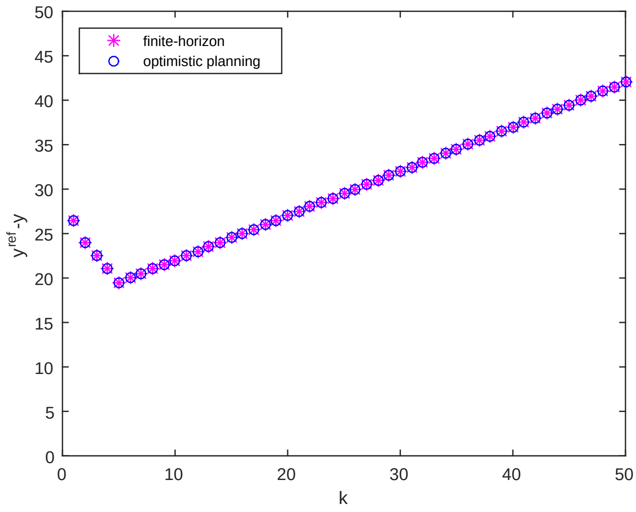


Figure 3.4: Tracking error for the closed-loop controlled system of the example of Section 3.3.4 when applying the first action only of the returned sequences for both approaches

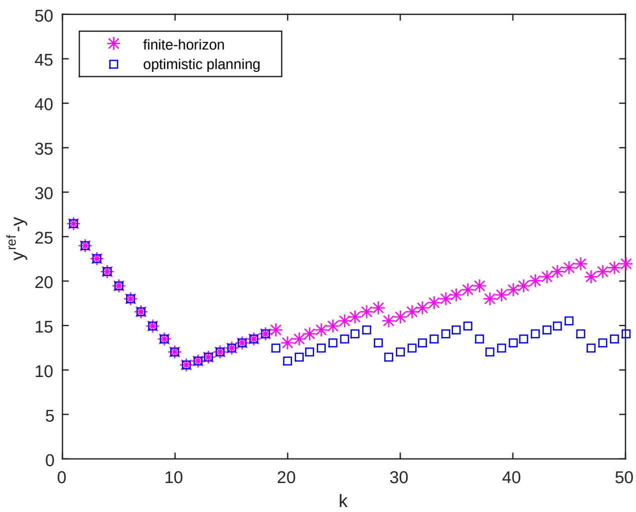


Figure 3.5: Tracking error for the closed-loop controlled system of the example of Section 3.3.4 when applying the first subsequence of length $\bar{d} = 9$ of the returned sequences for both approaches

We can see that the two approaches result in the same tracking error. Fig. 3.5 shows the results of applying a subsequence of length $\bar{d} = 9$ with $n = 500$ and $d_{\mathrm{N}} = 10$. We can see that in this case the optimistic planning approach gives a lower tracking error than the finite-horizon approach. In addition, for both approaches, the range of tracking errors by applying a subsequence is smaller than that by applying the first action only. Thus, for the considered MPL system (3.43)-(3.44), applying a subsequence of length $\bar{d} = 9$ yields a better tracking than applying the first action only for both approaches. However, this does not mean that applying a subsequence performs better for any experimental instance.

3.4 Conclusions

In this chapter, we have extended optimistic optimization and optimistic planning to model-based control for MPL systems.

In Section 3.2, we have generalized the expressions of the objective function given in [47, 133]. We have derived analytic expressions for the semi-metric required by the DOO algorithm for each objective function and extended the DOO algorithm to solve the MPL-MPC problem subject to bound constraints on the control variables. Based on the theoretical and numerical analysis, we found that the complexity of the proposed approach increases exponentially in the control horizon instead of the prediction horizon. This is in contrast to the worst-case complexity of the MILP method which is exponential in the prediction horizon. As illustrated by the numerical results, DOO is more efficient than MILP when the prediction horizon is large and the control horizon is small.

In Section 3.3, we have extended the OPD algorithm to the infinite-horizon optimal control problem for MPL systems with the control variable taking values in a finite set. The considered infinite-horizon discounted objective function aims at reducing the tracking error between the output signal and a due date signal. Within a limited computational budget, the OPD algorithm returns a control sequence the level of suboptimality of which can be characterized. In particular a bound can be derived for the difference between the optimal value of the objective function and the near-optimal value corresponding to the returned control sequence. The results of a numerical example show that for the given MPL system the proposed approach yields a better tracking than a finite-horizon approach when applying a subsequence of the returned control sequence.

We only considered the simple bounds on the input rate in Section 3.2. In the future, we will consider the case with general linear constraints on inputs and outputs. Moreover, we will focus on solving the robust optimal control problem for MPL systems with disturbances using (variants of) optimistic planning. We will also explore the infinite-horizon optimal control problem for other discrete-event and hybrid systems such as max-min-plus-scaling and piecewise affine systems.

Chapter 4

Optimistic optimization for MPC of continuous PWA systems

In this chapter we consider model predictive control (MPC) for discrete-time continuous piecewise affine systems with 1-norm and ∞-norm objective functions subject to linear constraints on the states and the inputs. We extend optimistic optimization to solve the resulting MPC optimization problem and derive analytic expressions for the core parameters required by optimistic optimization. We also discuss the level of suboptimality of the returned solution. The performance of the proposed approach is illustrated with a case study on adaptive cruise control.

4.1 Introduction

Piecewise affine (PWA) systems [127] are a subclass of hybrid systems, containing both continuous and discrete dynamics. PWA systems are defined by a polyhedral partition of the state and input space where each polyhedron is associated with an affine dynamical description. It has been proved [72] that continuous PWA systems are equivalent to other classes of hybrid systems, such as mixed logical dynamical (MLD) systems and max-min-plus-scaling (MMPS) systems. Based on this equivalence between continuous PWA systems and MLD systems, the MPC problem for continuous PWA systems can be written as a mixed-integer linear programming (MILP) problem [8]. However, the efficiency of solving the resulting MILP problem is limited by the number of integer variables. The number of integer variables is in general proportional to the value of the prediction horizon and the number of polyhedral partitions of the considered PWA system. The complexity of current MILP algorithms increases in the worst case exponentially if the number of integer variables increases. On the other hand, from the equivalence between continuous PWA systems and MMPS systems, the corresponding MPC optimization problem can be solved by a sequence of linear programming (LP) problems [48]. Nevertheless, the complexity of that approach is determined by the number of LP problems to be solved, which may increase rapidly if the prediction horizon increases. Therefore, developing an efficient approach with guaranteed performance for solving the continuous PWA-MPC optimization problem is the motivation of this chapter.

In this chapter, at each time step, a sequence vector of control inputs is computed by using optimistic optimization to solve a nonlinear, nonconvex optimization problem subject to linear constraints. The feasible set is transformed into a hyperbox by applying the penalty function method. Considering a 1-norm and ∞-norm objective function, we design a dedicated semi-metric and the analytic expressions for the parameters of optimistic optimization, which characterize the suboptimality of the solution in terms of the near-optimality dimension. We show that the near-optimality dimension of the resulting optimization problem is zero, which results in the suboptimality bound of the returned solution decreasing exponentially in the computational budget. Compared with the MILP method, which provides the true optimum, the solution returned by optimistic optimization given a finite computational budget is near-optimal, but optimistic optimization can be computationally efficient when the number of polyhedral partitions of the PWA system is large.

This chapter is organized as follows. In Section 4.2, the MPC problem for discrete-time PWA systems is formulated. In Section 4.3, the proposed optimistic-optimization-based approach is presented and the suboptimality is discussed. In Section 4.4, the effectiveness of the proposed approach is illustrated with an adaptive cruise control case study. Finally, Section 4.5 concludes this chapter.

4.2 MPC for continuous PWA systems

Consider the following discrete-time PWA system:

$$x(k+1) = A_i x(k) + B_i u(k) + g_i, \quad \text{for} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \mathscr{P}_i, \tag{4.1}$$

where the index k is the time counter[1], $x(k) \in \mathbb{R}^{n_x}$ is the state, $u(k) \in \mathbb{R}^{n_u}$ is the input, A_i, B_i, and g_i are the system matrices and vectors for the i-th region with $i \in \{1,\dots,N\}$ where N is the number of regions. Each region $\mathscr{P}_i$ is a polyhedron given as $\mathscr{P}_i = \{F_i x(k) + G_i u(k) \le h_i\}$ where F_i, G_i, and h_i are suitable matrices and vectors and $\{\mathscr{P}_i\}_{i=1}^{N}$ is a polyhedral partition of the state and input space.

As discussed in Section 2.2.2, the continuous PWA system (4.1) can equivalently be written as the MLD system (2.7) and the MMPS system (2.8).

Remark 4.1 In this chapter, we assume that the PWA system (4.1) is continuous, i.e., the right-hand side of (4.1) is continuous on the boundary of any two neighbouring regions. The advantage of considering continuous PWA systems is that (4.1) can equivalently be written as in the form of the MMPS system (2.8) without introducing additional auxiliary variables or extra constraints. □

4.2.1 Objective function and constraints

Let N_p and N_c be the prediction horizon and the control horizon. Define the sequence vectors

$$\tilde{x}(k) = \begin{bmatrix} x^T(k+1) & \cdots & x^T(k+N_\mathrm{p}) \end{bmatrix}^T,$$
$$\tilde{u}(k) = \begin{bmatrix} u^T(k) & \cdots & u^T(k+N_\mathrm{c}-1) \end{bmatrix}^T.$$

[1]Note that in Chapter 3, the index k is the event counter; while in this chapter, k represents the time counter.

Let r be a given reference signal. Define the control input increment

$$\Delta u(k) = u(k) - u(k-1). \tag{4.2}$$

In this chapter, we consider the following objective function

$$J(\tilde{u}(k)) = \|\tilde{x}(k) - \tilde{r}(k)\|_p + \lambda \|\Delta \tilde{u}(k)\|_q, \tag{4.3}$$

where $p, q \in \{1, \infty\}$, λ is a nonnegative scalar, and

$$\tilde{r}(k) = \begin{bmatrix} r^T(k+1) & \cdots & r^T(k+N_\mathrm{p}) \end{bmatrix}^T,$$
$$\Delta\tilde{u}(k) = \begin{bmatrix} \Delta u^T(k) & \cdots & \Delta u^T(k+N_\mathrm{c}-1) \end{bmatrix}^T.$$

Besides, we consider the following linear constraints on the state and the input:

$$P_k \tilde{x}(k) + Q_k \tilde{u}(k) \le b_k, \tag{4.4}$$
$$x_{\min} \le x(k+j) \le x_{\max}, \; j = 1, \dots, N_\mathrm{p}, \tag{4.5}$$
$$u_{\min} \le u(k+j-1) \le u_{\max}, \; j = 1, \dots, N_\mathrm{c}, \tag{4.6}$$

with $P_k \in \mathbb{R}^{n_\mathrm{c} \times n_x N_\mathrm{p}}$, $Q_k \in \mathbb{R}^{n_\mathrm{c} \times n_u N_\mathrm{c}}$, $b_k \in \mathbb{R}^{n_\mathrm{c}}$, $x_{\min}, x_{\max} \in \mathbb{R}^{n_x}$, and $u_{\min}, u_{\max} \in \mathbb{R}^{n_u}$.

4.2.2 Problem formulation

At time step k, the MPC problem for the system (4.1) can be written as

$$\min_{\tilde{u}(k)} J(\tilde{u}(k)) \tag{4.7}$$

subject to

$$\text{the prediction model (4.1), (2.7), or (2.8)} \tag{4.8}$$
$$P_k \tilde{x}(k) + Q_k \tilde{u}(k) \le b_k, \tag{4.9}$$
$$x_{\min} \le x(k+j) \le x_{\max}, \; j = 1, \dots, N_\mathrm{p}, \tag{4.10}$$
$$u_{\min} \le u(k+j-1) \le u_{\max}, \; j = 1, \dots, N_\mathrm{c}, \tag{4.11}$$
$$u(k+j) = u(k+N_\mathrm{c}-1), \; j = N_\mathrm{c}, \dots, N_\mathrm{p}-1. \tag{4.12}$$

An optimal control sequence vector $\tilde{u}(k)$ is obtained by solving the problem (4.7)-(4.12); subsequently, only the first control input $u(k)$ is applied to the system. At the next time step, this process is repeated.

Remark 4.2 In Section 2.2.2, we have stated that the PWA system (4.1) can equivalently be represented as an MLD system in the form of (2.7). If (2.7) is used as the prediction model, the PWA-MPC problem (4.7)-(4.12) can be recast into an MILP problem following the procedures in [8] where the number of variables and constraints is proportional to the product nNN_p. However, in practice, the worst-case complexity of the MILP problem is exponential in nNN_p. When the system (4.1) is continuous, another solution approach for the problem (4.7)-(4.12) has been presented in [48]. That approach is based on the equivalence between the PWA system (4.1) and the MMPS system (2.8) and consists in solving a sequence of LP problems based on writing the objective function (4.3) in the form (2.5). The number of LP problems is determined by the number of minimization operations in (2.5); the size of each LP problem relates to the number of maximization operations corresponding to each minimization operation. But the number of minimization and maximization operations may increase rapidly with growing N_p, which makes the approach less efficient. Therefore, we are motivated to alleviate the influence of N_p on the complexity. In the remaining part of this chapter, we will introduce an approach the complexity of which mainly depends on N_c instead of N_p. □

4.3 Optimistic optimization approach

In this section, we present the optimistic optimization approach for the PWA-MPC problem (4.7)-(4.12) provided that the PWA system (4.1) is continuous.

Recall the definitions of 1-norm and ∞-norm for vectors $x \in \mathbb{R}^n$:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|, \quad \|x\|_\infty = \max_{i=1,\ldots,n} |x_i|, \quad \text{and} \quad |x_i| = \max(x_i, -x_i).$$

According to the equivalence between the system (4.1) and (2.8), the objective function (4.3) can be transformed into an expression in the form of (2.4) or (2.5) (cf. Theorem 3.1 in [48]). Currently, we consider the max-min canonical form (2.4) while the results can easily be extended to (2.5) due to duality.

Using (2.8) as the prediction model, the state vector $\tilde{x}(k)$ can be eliminated from the objective function. Then the objective function (4.3) only has $\tilde{u}(k)$ as the independent variable:

$$J(\tilde{u}(k)) = \max_{i \in \mathcal{I}} \min_{j \in \mathcal{J}_i} \left\{ \alpha_{ijk}^T \tilde{u}(k) + \beta_{ijk} \right\}, \tag{4.13}$$

with $\alpha_{ijk} \in \mathbb{R}^{n_u N_\mathrm{c}}$, $\beta_{ijk} \in \mathbb{R}$. The parameter vectors α_{ijk} and the constant terms β_{ijk} can be computed from the known information at time step k, namely, the system matrices and vectors A_i, B_i, and g_i in (4.1), the reference sequence vector $\tilde{r}$, the current state $x(k)$, and the previous control input $u(k-1)$.

4.3.1 Penalty method

The feasible set defined by constraints (4.4)-(4.6) is a polytope. In order to easily guarantee the Assumptions 2.10-2.12 for optimistic optimization, we transform the problem into a problem with hyperbox constraints. Hence, we treat (4.4) and (4.5) as soft constraints and replace them by adding a penalty function to the objective function:

$$\begin{aligned} J_\mathrm{p}(\tilde{u}(k)) = \beta \cdot \max\Big(0, & \\ & \max_{i=1,\ldots,n_\mathrm{c}} \left(P_{i,\cdot}\tilde{x}(k) + Q_{i,\cdot}\tilde{u}(k) - b_i\right), \\ & \max_{j=1,\ldots,N_\mathrm{p}} \max_{l=1,\ldots,n_x} \left(x_l(k+j) - x_{\max,l}, x_{\min,l} - x_l(k+j)\right)\Big), \end{aligned} \tag{4.14}$$

where β is the penalty coefficient; $P_{i,\cdot}$ and $Q_{i,\cdot}$ are the i-th rows of P_k and Q_k respectively; b_i is the i-th element of b_k; $x_l(k+j)$, $x_{\min,l}$, and $x_{\max,l}$ are the l-th elements of $x(k+j)$, $x_{\min}$, and $x_{\max}$ respectively. So we have the new objective function

$$J_{\text{new}}(\tilde{u}(k)) = J(\tilde{u}(k)) + J_{\text{p}}(\tilde{u}(k)), \tag{4.15}$$

subject to the bound constraint (4.6). Consequently, the feasible set is actually an $n_u N_{\text{c}}$-dimensional hyperbox, i.e.,

$$\mathscr{U} \triangleq \left\{ \tilde{u} \middle| \begin{bmatrix} u_{\min}^T & \cdots & u_{\min}^T \end{bmatrix}^T \leq \tilde{u} \leq \begin{bmatrix} u_{\max}^T & \cdots & u_{\max}^T \end{bmatrix}^T \right\}.$$

By performing scaling and translation operations, the hyperbox $\mathscr{U}$ can be transformed into a unit hypercube $\mathscr{U}_{\text{c}}$. For the sake of simplicity of notation, we assume from now on that $\tilde{u}$ actually already belongs to a unit hypercube $\mathscr{U}_{\text{c}}$. Note that the new objective function can also be written in the form

$$J_{\text{new}}(\tilde{u}) = \max_{i \in \mathscr{I}} \min_{j \in \mathscr{J}_i} \{\hat{\alpha}_{ij}^T \tilde{u} + \hat{\beta}_{ij}\}, \tag{4.16}$$

with $\hat{\alpha}_{ij} \in \mathbb{R}^{n_u N_{\text{c}}}$, $\hat{\beta}_{ij} \in \mathbb{R}$. In the remaining part of this section the time counter k is omitted for the sake of simplicity.

4.3.2 Development and analysis

Now we design the semi-metric ℓ, the diameter $\delta(h)$, and the scalar ν for solving the continuous PWA-MPC problem (4.7)-(4.12) with the new objective function (4.16) using optimistic optimization. These parameters are required for the implementation of the deterministic optimistic optimization (DOO) algorithm and for the characterization of the suboptimality of the returned solution.

Theorem 4.3 *Define*

$$\bar{\alpha} \triangleq \max_{i,j} \|\hat{\alpha}_{ij}\|_2,$$

where $\hat{\alpha}_{ij}$ are the parameter vectors in (4.16). *Let $\tilde{u}^*$ be a global optimizer of the objective function J_{new} subject to $\tilde{u} \in \mathscr{U}_{\text{c}}$. Recalling the hierarchical partitioning framework presented in Section 2.4.1, let the branching number $K = D^{n_u N_{\text{c}}}$ where $n_u N_{\text{c}}$ is the dimension of the hypercube $\mathscr{U}_{\text{c}}$ and each edge of $\mathscr{U}_{\text{c}}$ is cut into D equal parts. Let $U^{h,d}$ be the cell at depth h with node index d and let $\tilde{u}^{h,d} \in U^{h,d}$ be the center of $U^{h,d}$.*

(i) If we define

$$\ell(\tilde{u}, \tilde{v}) = \bar{\alpha} \|\tilde{u} - \tilde{v}\|_2, \tag{4.17}$$

for any $\tilde{u}, \tilde{v} \in \mathscr{U}_{\text{c}}$, then ℓ is a semi-metric defined on $\mathscr{U}_{\text{c}}$ such that for any $\tilde{u} \in \mathscr{U}_{\text{c}}$, we have

$$J_{\text{new}}(\tilde{u}) - J_{\text{new}}(\tilde{u}^*) \leq \ell(\tilde{u}, \tilde{u}^*). \tag{4.18}$$

(ii) If we define

$$\delta(h) = \frac{\bar{\alpha}}{2} (n_u N_{\text{c}})^{1/2} / D^h, \tag{4.19}$$

then for any cell $U^{h,d}$ at any depth h, we have

$$\sup_{\tilde{u} \in U^{h,d}} \ell(\tilde{u}, \tilde{u}^{h,d}) \leq \delta(h). \tag{4.20}$$

(iii) Select $0 < \rho \le 1$. *If we define*

$$\nu = \rho (n_u N_\mathrm{c})^{-1/2}, \tag{4.21}$$

then any cell $U^{h,d}$ *contains an* ℓ*-ball with radius* $\nu\delta(h)$ *centered in* $\tilde{u}^{h,d}$.

Proof: (i) From Proposition 2.5, the objective function J_new is a continuous PWA function. It is easy to verify that the constant

$$\bar{\alpha} \triangleq \max_{i,j} \|\hat{\alpha}_{ij}\|_2,$$

is actually a Lipschitz constant[2] for J_new [56]. According to the Lipschitz continuity, we have

$$J_\mathrm{new}(\tilde{u}) - J_\mathrm{new}(\tilde{u}^*) \le \bar{\alpha} \|\tilde{u} - \tilde{u}^*\|_2,$$

for any $\tilde{u} \in \mathscr{U}_\mathrm{c}$. If we define the semi-metric as

$$\ell(\tilde{u}, \tilde{v}) = \bar{\alpha} \|\tilde{u} - \tilde{v}\|_2,$$

then the inequality (4.18) is satisfied.

(ii) Recall the hierarchical partitioning presented in Section 2.4.1. The feasible set $\mathscr{U}_\mathrm{c}$ is a unit hypercube, so the maximum distance between any two points in $\mathscr{U}_\mathrm{c}$ is $(n_u N_\mathrm{c})^{1/2}$. The cell $U^{h,d}$ at depth h of the partitioning is also a hypercube and the edge length of $U^{h,d}$ is $1/D^h$. Because $\tilde{u}^{h,d}$ is the center of the cell $U^{h,d}$, for any $\tilde{u} \in U^{h,d}$, we have

$$\|\tilde{u} - \tilde{u}^{h,d}\|_2 \le \frac{1}{2} (n_u N_\mathrm{c})^{1/2} / D^h.$$

Define

$$\delta(h) = \frac{\bar{\alpha}}{2} (n_u N_\mathrm{c})^{1/2} / D^h.$$

Therefore, for any $\tilde{u} \in U^{h,d}$, we have

$$\ell(\tilde{u}, \tilde{u}^{h,d}) = \bar{\alpha} \|\tilde{u} - \tilde{u}^{h,d}\|_2 \le \delta(h).$$

(iii) An ℓ-ball of radius $\nu\delta(h)$ centered in $\tilde{u}^{h,d}$ can be written as

$$\mathfrak{B} = \left\{ \tilde{u} \in \mathscr{U}_\mathrm{c} \middle| \ell(\tilde{u}, \tilde{u}^{h,d}) = \bar{\alpha} \|\tilde{u} - \tilde{u}^{h,d}\|_2 \le \nu\delta(h) \right\}.$$

Note that $\mathscr{U}_\mathrm{c}$ is a hypercube and so is the cell $U^{h,d}$. Thus, the center $u^{h,d}$ is also the center of the inscribed ball of $U^{h,d}$. Let $r(h)$ be the radius of the inscribed hyperball of $U^{h,d}$, so $r(h) = \frac{1}{2} L / D^h$. If we select ν such that

$$0 < \nu \le \frac{\bar{\alpha} r(h)}{\delta(h)},$$

then for any $\tilde{u} \in \mathfrak{B}$, we have

$$\|\tilde{u} - \tilde{u}^{h,d}\|_2 \le \frac{\nu\delta(h)}{\alpha} \le r(h).$$

[2] A Lipschitz constant of a function is a real number such that for every pair of points on the graph of this function, the absolute value of the slope of the line connecting them is not greater than this real number.

Hence, then we have $\mathfrak{B} \subset U^{h,d}$. Note that

$$\frac{\bar{\alpha} r(h)}{\delta(h)} = (n_u N_\mathrm{c})^{-1/2}.$$

Thus if we select a scalar $0 < \rho \leq 1$ and choose

$$\nu = \rho (n_u N_\mathrm{c})^{-1/2},$$

then $U^{h,d}$ contains an ℓ-ball of radius $\nu\delta(h)$ centered in $\tilde{u}^{h,d}$. □

Up to now, we have derived the expressions for all core parameters required by optimistic optimization. At each time step k, we apply optimistic optimization to solve the MPC optimization problem (4.7)-(4.12) to obtain a sequence of control inputs. To discuss the suboptimality of the returned solution, we compute the local near-optimality dimension for the objective function J_new over $\mathscr{U}_\mathrm{c}$. Denote the set of ε-near-optimal solutions as

$$\mathscr{U}_\varepsilon = \{\tilde{u} \in \mathscr{U}_\mathrm{c} | J_\mathrm{new}(\tilde{u}) - J_\mathrm{new}(\tilde{u}^*) \leq \varepsilon\}.$$

Theorem 4.4 *Let $\tilde{u}^*$ be a global optimizer of J_new subject to $\tilde{u} \in \mathscr{U}_\mathrm{c}$ and let $\tilde{u}^\natural$ be the solution returned by optimistic optimization after n iterations. If $\tilde{u}^*$ is a strict local minimizer of J_new, then the local near-optimality dimension is $\eta = 0$ and we have*

$$J_\mathrm{new}(\tilde{u}^\natural) - J_\mathrm{new}(\tilde{u}^*) \leq \frac{\bar{\alpha}}{2} (n_u N_\mathrm{c})^{1/2} D^{1-n/C},$$

for some constant $C > 0$.

Proof: Because J_new is a continuous PWA function and $\tilde{u}^* \in \mathscr{U}_\mathrm{c}$ is a strict local minimizer, there exists a $\delta_0 > 0$ such that the neighbourhood

$$\mathscr{N}(\tilde{u}^*, \delta_0) = \{\tilde{u} \in \mathscr{U}_\mathrm{c} | \|\tilde{u} - \tilde{u}^*\|_2 \leq \delta_0\},$$

consists of n_{δ_0} subregions where each region is associated with an affine function of J_new. Denote the affine functions active in the neighbourhood $\mathscr{N}(\tilde{u}^*, \delta_0)$ as

$$\alpha_i^T \tilde{u} + \beta_i, \; i = 1, \ldots, n_{\delta_0}.$$

For any $\tilde{u} \in \mathscr{N}(\tilde{u}^*, \delta_0)$, define the line that connects $\tilde{u}$ to $\tilde{u}^*$; so $J_\mathrm{new}(\tilde{u})$ decreases linearly when $\tilde{u}$ varies towards $\tilde{u}^*$ along this line. Furthermore, the smallest slope with which the objective function can decrease is given as

$$\underline{\alpha} = \min_{i=1,\ldots,n_{\delta_0}} \min_{j=1,\ldots,n_u N_\mathrm{c}} |\alpha_{i,j}|,$$

where $\underline{\alpha} > 0$ if $\tilde{u}^*$ is a strict local minimizer of J_new. Thus, for any $\tilde{u} \in \mathscr{N}(\tilde{u}^*, \delta_0)$, we have

$$J_\mathrm{new}(\tilde{u}) - J_\mathrm{new}(\tilde{u}^*) \geq \underline{\alpha} \|\tilde{u} - \tilde{u}^*\|_2. \tag{4.22}$$

For any $\delta_0 > 0$, there exists a $\varepsilon_0 > 0$ such that for any $\tilde{u} \in \mathscr{N}(\tilde{u}^*, \delta_0)$, we have

$$J_\mathrm{new}(\tilde{u}) - J_\mathrm{new}(\tilde{u}^*) \leq \varepsilon_0.$$

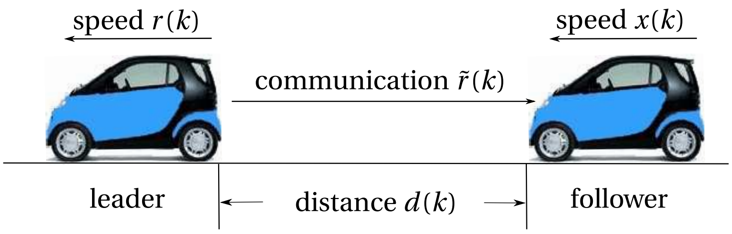


Figure 4.1: Adaptive cruise control set-up considered in the case study of Section 4.4

From (4.22), we have

$$\|\tilde{u} - \tilde{u}^*\|_2 \leq \varepsilon_0/\underline{\alpha}.$$

Furthermore, for any $\varepsilon \in (0, \varepsilon_0]$, for any $\tilde{u} \in \mathscr{U}_\varepsilon$, we have

$$\|\tilde{u} - \tilde{u}^*\|_2 \leq \varepsilon/\underline{\alpha}.$$

Namely, for any $\varepsilon \in (0, \varepsilon_0]$, the set $\mathscr{U}_\varepsilon$ is included in the set $\mathcal{N}(\tilde{u}^*, \varepsilon/\underline{\alpha})$. Note that the set $\mathcal{N}(\tilde{u}^*, \varepsilon/\underline{\alpha})$ is actually an ℓ-ball of radius $(\bar{\alpha}/\underline{\alpha})\varepsilon$, where the semi-metric ℓ is defined as in (4.17). Therefore, there exists a constant $C > 0$ such that the maximal number of disjoint ℓ-balls of radius $\nu\varepsilon$ with center in $\mathcal{N}(\tilde{u}^*, \varepsilon/\underline{\alpha})$ is less than C. Hence, based on Definition 2.14, we then have $\eta = 0$.

According to (4.19), if we define $c = \frac{\bar{\alpha}}{2}(n_u N_\mathrm{c})^{1/2}$ and $\gamma = \frac{1}{D}$, then we have $\delta(h) = c\gamma^h$. From Theorem 2.15(ii), we have

$$\begin{aligned} J_\mathrm{new}(\tilde{u}^\natural) - J_\mathrm{new}(\tilde{u}^*) &\leq c\gamma^{n/C-1} \\ &\leq \frac{\bar{\alpha}}{2}(n_u N_\mathrm{c})^{1/2} D^{1-n/C}. \end{aligned}$$

□

Remark 4.5 Theorem 4.4 shows that with the semi-metric (4.17), for the continuous PWA-MPC problem with the objective function (4.16) subject to (4.6) the near-optimality dimension is $\eta = 0$ when the optimizer is strict. This means that the optimization problem can be solved by optimistic optimization efficiently, converging quickly to the optimal solution. □

4.4 Example: Adaptive cruise control

In this section, we demonstrate the proposed approach with an adaptive cruise control problem for a road vehicle following a leader vehicle. We consider the setup introduced in [38]. The goal of a cruise controller is to track the velocity of the vehicle in front, guaranteeing secure driving and optimal usage of the brake system. The velocity of the leader vehicle is communicated to the follower vehicle and considered as a reference signal. As shown in Fig. 4.1, let $x(k)$ be the velocity of the follower vehicle at time step k. Let $r(k)$ be the velocity of the leader vehicle at time step k. A nonlinear model for the positive velocity of the follower vehicle is given in [38]. That model can be approximated by the following continuous PWA system:

$$x(k+1) = A_i x(k) + B_i u(k) + g_i, \text{ if } x \in (p_{i-1}, p_i], \tag{4.23}$$

with $i = 1,2$, $A_1 = 0.9883$, $B_1 = 4.598$, $g_1 = -0.0614$, $A_2 = 0.9655$, $B_2 = 4.5446$, $g_2 = 0.3711$, $p_0 = 0$, $p_1 = \frac{x_{\max}}{2}$, and $p_2 = x_{\max}$ where $x_{\max} = 37.5$ m/s is the maximum velocity and p_1 is the breakpoint for the least-squares fitting of the nonlinear friction. The control input $u(k)$ is the throttle/brake position at time step k.

Note that (4.23) is equivalent to the following MMPS system:

$$x(k+1) = \min\left(A_1 x(k) + B_1 u(k) + g_1,\ A_2 x(k) + B_2 u(k) + g_2\right). \tag{4.24}$$

Let $d(k)$ be the distance between two vehicles at time step k, so $d(k+1) = d(k) + (r(k) - x(k))T$ with T the sampling time. Due to safety and human comfort requirements, we add constraints on $d(k)$, $x(k)$, $u(k)$ for each time step k:

$$d_{\text{safe}} \le d(k+j),\ j = 1, \dots, N_{\text{p}}, \tag{4.25}$$

$$a_{\text{dec}} T \le x(k+j) - x(k+j-1) \le a_{\text{acc}} T,\ j = 1, \dots, N_{\text{p}}, \tag{4.26}$$

$$-\tau \le \Delta u(k+j-1) \le \tau,\ j = 1, \dots, N_{\text{c}}, \tag{4.27}$$

$$x_{\min} \le x(k+j) \le x_{\max},\ j = 1, \dots, N_{\text{p}}, \tag{4.28}$$

$$-u_{\max} \le u(k+j-1) \le u_{\max},\ j = 1, \dots, N_{\text{c}}, \tag{4.29}$$

where $d_{\text{safe}} = 10$ m corresponds to the safe following distance to reduce the risk of collision, $a_{\text{acc}} = 2.5$ m/s^2 and $a_{\text{dec}} = -1$ m/s^2 are the allowable acceleration and deceleration for human comfort, $\tau = 0.2$ is the maximum brake variation, $x_{\max} = 37.5$ m/s and $x_{\min} = 5$ m/s are the maximum and minimum velocities, and $u_{\max} = 1$ is the maximum brake.

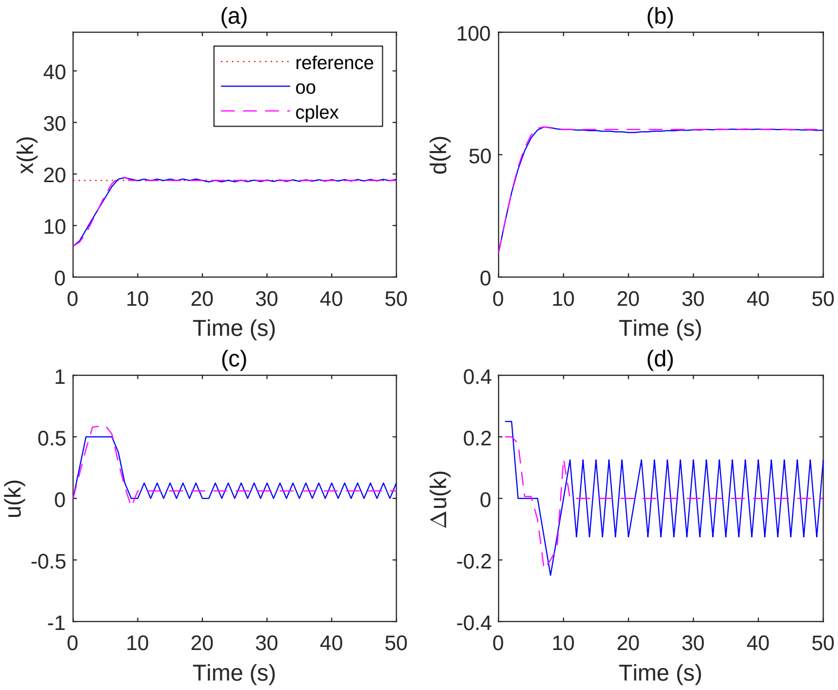
In order to minimize the velocity deviation between the follower and the leader vehicle and minimize the variation of the control input Δu, we consider the following objective function:

$$J(\tilde{u}(k)) = \|\tilde{x}(k) - \tilde{r}(k)\|_\infty + \lambda \|\Delta \tilde{u}(k)\|_1, \tag{4.30}$$

with the trade-off $\lambda = 0.05$ and $N_{\text{p}} = N_{\text{c}} = 2$. Based on (4.24), $\tilde{x}(k)$ and $\Delta\tilde{u}(k)$ in (4.30) can be substituted by $\tilde{u}(k)$. Moreover, the constraints (4.25)-(4.28) are replaced by adding a penalty function to the objective function. The penalty coefficient is selected as $\beta = 10$. The new objective function can be rewritten in the form of (4.16) and the resulting feasible set is a hypercube.

At each time step k, the MPC optimization problem is respectively solved by using the MILP method and the optimistic optimization approach. The corresponding MILP problem is solved by the `cplex` function (with the default settings) in the Tomlab optimization environment in Matlab. The optimistic optimization approach is implemented in Matlab. The termination criteria of optimistic optimization (`oo`) are a combination of the computational budget and the depth limitation. More specifically, given the number of node expansions $t_{\max}$, the number of evaluations (computational budget) of the objective function is $n = K t_{\max} + 1$ with $K = 2^{N_{\text{c}}}$ the branching number in the tree. In addition, the maximum depth of the resulting tree is limited as $h_{\max} = 10$. The algorithm will terminate and return the best solution if the computational budget is used or the maximum depth is reached.

(1) Constant reference velocity r



(2) Varying reference velocity r

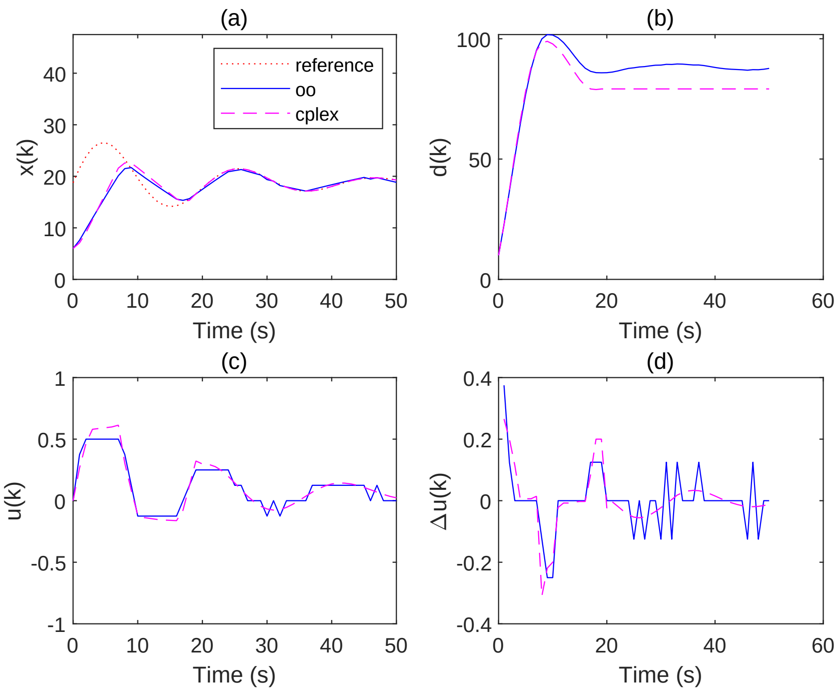


Figure 4.2: Simulation results of `cplex` *and optimistic optimization (*`oo`*) for the example of Section 4.4 for constant and varying reference velocities ($t_{\max}$ = 10 for* `oo`*): (a) Velocity of the follower vehicle; (b) Distance between the two vehicles; (c) Control input; (d) Throttle/Brake variation*

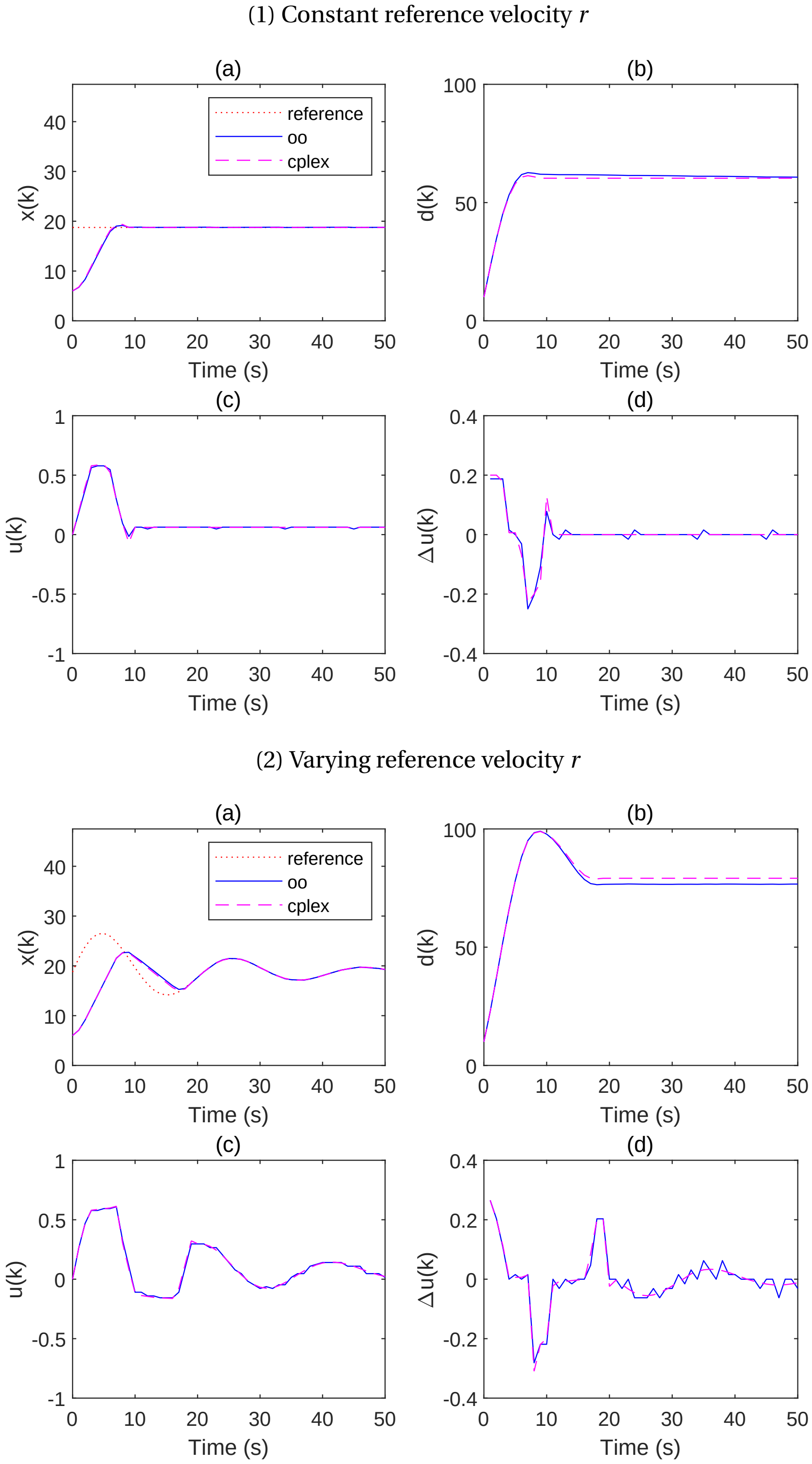


Figure 4.3: Simulation results of `cplex` and optimistic optimization (`oo`) for the example of Section 4.4 for constant and varying reference velocities ($t_{\max} = 100$ for `oo`): (a) Velocity of the follower vehicle; (b) Distance between the two vehicles; (c) Control input; (d) Throttle/Brake variation

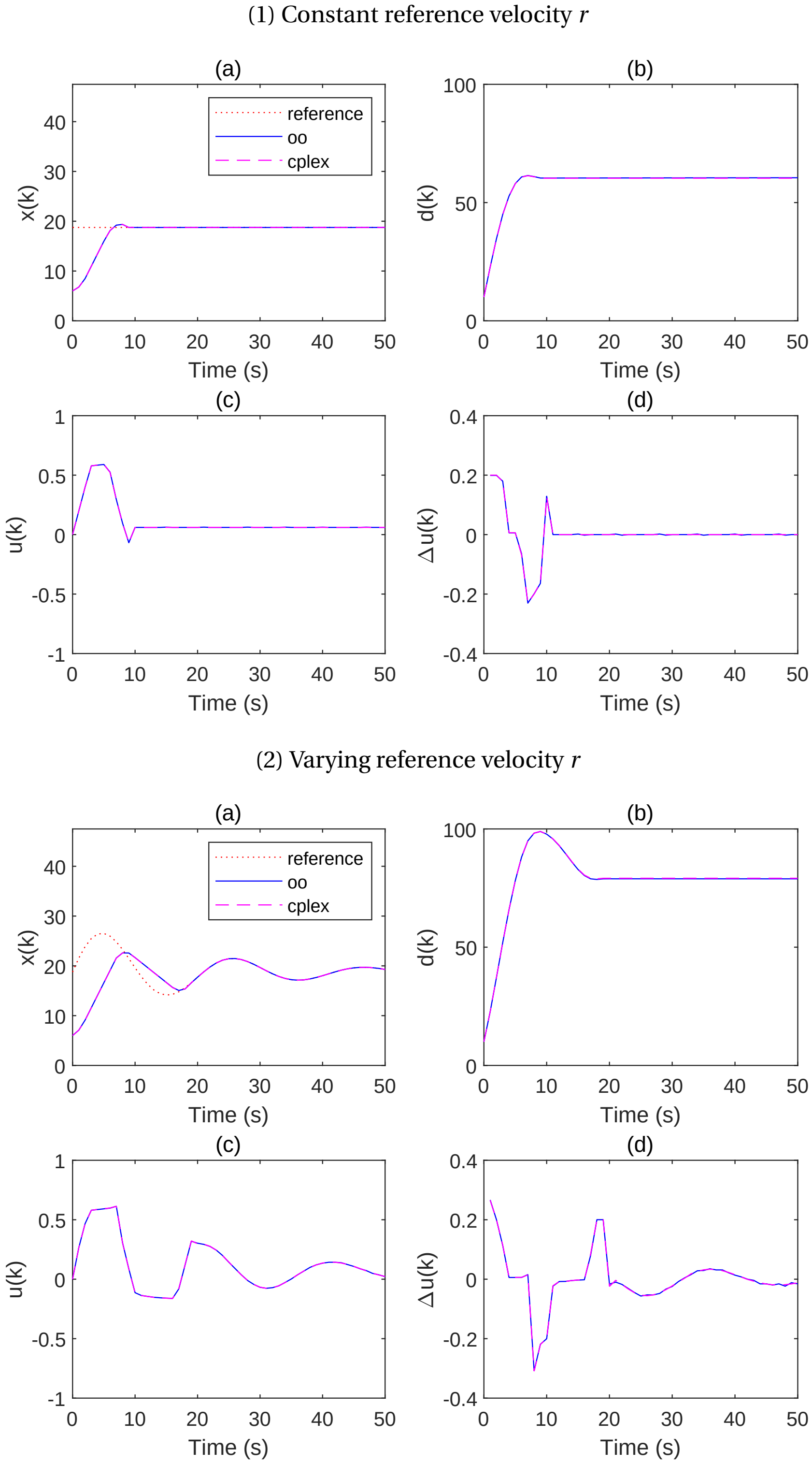


Figure 4.4: Simulation results of `cplex` *and optimistic optimization (*`oo`*) for the example of Section 4.4 for constant and varying reference velocities ($t_{\max}$ = 1000 for* `oo`*): (a) Velocity of the follower vehicle; (b) Distance between the two vehicles; (c) Control input; (d) Throttle/Brake variation*

Table 4.1: CPU times per step, closed-loop costs over the simulation period, and the relative error of `oo` *and* `cplex`

	$t_{\max} = 10$	$t_{\max} = 100$	$t_{\max} = 1000$	`cplex`
CPU time (s)	0.001	0.01	0.1	0.004
Constant r	45.8	41.96	39.98	39.76
	15.19%	5.51%	0.53%	0
Varying r	102.62	100.22	96.92	96.62
	6.21%	3.72%	0.3%	0

Fig. 4.2 shows the simulation results of adaptive cruise control for the follower vehicle tracking different reference velocities over the simulation horizon $[1, 50]$. The constant reference velocity is 18.75 m/s and the varying reference velocity is given as $r(k) = 10e^{-0.05k}\sin(0.3k) + 18.75$. The number of node expansions in optimistic optimization is $t_{\max} = 10$. We can see that the trajectory of the velocity of the follower vehicle controlled by optimistic optimization can track both types of reference velocities (Fig. 4.2(1a) and 4.2(2a)). However, the variation of the control input is not smooth, especially for the case with constant reference. Fig. 4.3 and 4.4 shows the simulation results when $t_{\max}$ is increased for optimistic optimization from 10 to 100 and 1000. We can see that the trajectories of the velocity and the distance resulting from optimistic optimization track the trajectories resulting from `cplex` better than the case in Fig. 4.2. Moreover, the control inputs solved by optimistic optimization are smoother and quite close to the control inputs solved by `cplex`. The closed-loop cost over the simulation period of optimistic optimization with $t_{\max} = 1000$ is 96.92 for the varying reference signal; the relative error compared with the cost of `cplex` is 0.3% (this relative error is computed as $100|(\text{cost}_{\texttt{cplex}} - \text{cost}_{\texttt{oo}})/\text{cost}_{\texttt{cplex}}|$). The closed-loop costs of optimistic optimization given different computational budgets and the relative error comparing with `cplex` are listed in Table 4.1. The relative error of closed-loop costs of optimistic optimization decreases if the computational budget increases. The average CPU times for optimistic optimization and `cplex` solving the optimization problem at each time step are also included in Table 4.1. Note that optimistic optimization will be faster if we would transfer the Matlab code into object code.

4.5 Conclusions

In this chapter, we have extended optimistic optimization to MPC for discrete-time continuous PWA systems and MMPS systems, which in general leads to an MILP problem. We have considered a 1-norm and ∞-norm objective function subject to a hyperbox feasible set. We have developed a dedicated semi-metric and other parameters required by optimistic optimization for the corresponding problem. In addition, a bound on the suboptimality of the returned solution with respect to a global optimum has been derived given a finite computational budget. A case study on adaptive cruise control has been implemented to illustrate the performance of the proposed approach.

In our future work, we will investigate the stochastic MPC for PWA systems with uncertainties. Moreover, we will also derive expressions for the core parameters of optimistic optimization considering a polytopic feasible set.

Chapter 5

Optimistic optimization of continuous nonconvex PWA functions

In the previous chapter we have considered model predictive control for continuous piecewise affine (PWA) systems with a 1-norm or ∞-norm objective function subject to a hyperbox feasible set. We have seen that the resulting model predictive control optimization problem actually involves the optimization of a continuous nonconvex PWA function over a hyperbox. The current chapter is an extension of the previous chapter. More precisely, we extend optimistic optimization to the global optimization problem of a continuous nonconvex PWA function over a polytope. Moreover, we replace the common assumptions of optimistic optimization by just one compact assumption and correspondingly adapt the definition for the near-optimality dimension. In addition, we provide a partitioning approach for a polytope by employing Delaunay triangulation and edgewise subdivision. For this partitioning, we derive the analytic expressions for the core parameters required by optimistic optimization for continuous PWA functions.

5.1 Introduction

Piecewise affine (PWA) functions are widely used in various fields for approximating nonlinearities, see [6, 111, 128]; they also appear as cost functions of numerous optimization problems, see [41, 104, 117]. During the last decades, optimization of PWA functions has been investigated by many researchers. A traditional technique for the optimization of a *convex* PWA function subject to linear constraints consists in transforming the problem into a single equivalent linear programming (LP) problem and then applying LP methods. Moreover, some LP methods have been extended to directly deal with the optimization of convex PWA function without resorting to LP reformulations, e.g., the simplex algorithm [55] and the interior point algorithm [29]. The optimization of a *nonconvex* PWA function is often recast as a mixed integer linear programming (MILP) problem [40, 141]. However, the worst-case complexity of MILP solvers grows exponentially with the number of polyhedral subregions of the PWA function, which usually make the problem solving process less efficient.

In this chapter we compress the common assumptions of optimistic optimization into a compact one and give a new definition for the near-optimality dimension, which is used for measuring the complexity of the optimization problem. Moreover, the linear constraints on the optimization variables are now considered as hard constraints and, for the first time in the literature on optimistic optimization, a polytopic feasible set is considered. This extension from a hyperbox feasible set to a polytopic one is not trivial but useful because a polytopic feasible set allows to include general affine constraints on the control variables rather than only single bound constraints. A partition of the given polytope is required to perform the search process. The partitioning should generate well-shaped cells that shrink with the depth. We first employ Delaunay triangulation to divide the polytope into a mesh of simplices and next repeatedly use edgewise subdivision to subdivide the simplices into smaller simplices that satisfy the requirements for optimistic optimization. For this partitioning approach, we develop analytic expressions for the core parameters of optimistic optimization based on the knowledge of the Lipschitz constants of the PWA objective function f. The effectiveness of the resulting algorithm is illustrated with numerical examples and the results show that using optimistic optimization algorithms for the optimization of a continuous and nonconvex PWA function over a given polytope is more efficient than transforming it into an MILP problem if the number of polyhedral subregions of the PWA function is large. The second example shows that the proposed approach is also efficient for the optimization of max-min-plus-scaling (MMPS) functions, which are equivalent to continuous PWA functions.

This chapter is organized as follows. In Section 5.2, we describe the optimization problem of continuous nonconvex PWA functions. In Section 5.3, we adapt the deterministic optimistic optimization (DOO) algorithm to the setting in [68]. In Section 5.4, we propose a partitioning approach for which we develop the analytic expressions for the core parameters of DOO. In Section 5.5, the proposed approach is assessed with numerical examples. Finally, Section 5.6 includes some conclusions and future work directions.

5.2 Problem statement

Consider the following optimization problem:

$$\min_{x\in\mathscr{P}} f(x) \tag{5.1}$$

subject to

$$Ax \le b. \tag{5.2}$$

The objective function $f:\mathscr{P}\to\mathbb{R}$ is a scalar-valued continuous PWA function where $\mathscr{P}\in\mathbb{R}^{n_x}$ is a polyhedron and there exists a polyhedral partition $\{\mathscr{P}_i\}_{i=1}^{N}$ of $\mathscr{P}$ such that f is affine on each $\mathscr{P}_i$, i.e.,

$$f(x)=\alpha_{(i)}^T x+\beta_{(i)},\ \forall x\in\mathscr{P}_i,$$

with $\alpha_{(i)}\in\mathbb{R}^{n_x}$, $\beta_{(i)}\in\mathbb{R}$, $i=1,\dots,N$. The matrix $A\in\mathbb{R}^{m\times n_x}$ and vector $b\in\mathbb{R}^m$ are the constraint matrix and vector. We assume that the feasible set

$$\mathscr{X}=\{x\in\mathbb{R}^{n_x}|Ax\le b\},$$

is nonempty and bounded. From Definition 2.2, $\mathscr{X}$ is a polytope. If f is convex, then the problem (5.1)-(5.2) is equivalent to a set of LP problems [92], which can be solved very efficiently.

In this chapter, we consider the case that f is continuous and nonconvex and that the number of polyhedral subregions N is much larger than n_x. For this case, one possible solution approach consists in transforming the problem (5.1)-(5.2) into an MILP problem. The number of auxiliary variables and linear constraints in the resulting MILP description is proportional to N. So the complexity of the resulting MILP problem grows in the worst case exponentially in N. In the next section, we will introduce an optimistic optimization algorithm for the problem (5.1)-(5.2). The knowledge of a Lipschitz constant of f is important for designing the two key parameters ν and ρ of optimistic optimization. For any $x, y \in \mathscr{P}_i$, we have

$$\begin{aligned} \left|f(x) - f(y)\right| &= \left|\alpha_{(i)}^T x + \beta_{(i)} - \alpha_{(i)}^T y - \beta_{(i)}\right| \\ &= \left|\alpha_{(i)}^T (x-y)\right| \\ &\le \|\alpha_{(i)}\|_2 \|x-y\|_2 \ . \end{aligned} \tag{5.3}$$

The last inequality is obtained from the property that for every $x, y \in \mathbb{R}^n$, we have $|x^T y| \le \|x\|_2 \|y\|_2$. It is easy to verify that $\max_{i=1,\ldots,N} \|\alpha_{(i)}\|_2$ is the smallest Lipschitz constant of f (see [56] and Proposition 2.2.7 in [122] for a proof).

5.3 Adaptation of DOO

In Section 2.4, we have introduced the background of the deterministic optimistic optimization (DOO) algorithm. In this section, we particularly adapt the assumptions presented in [100] to a compact one like the one in [68]. In this section, we consider f and $\mathscr{X}$ as a general objective function and a general feasible space.

Four necessary assumptions are stated in [100] (written as Assumptions 2.10-2.12 in Section 2.4.2) regarding the function f and the partitioning used in the DOO algorithm. Those assumptions are expressed in terms of a semi-metric. However, as discussed in [68], this semi-metric actually just seems to link the function and the partitioning and it is not used in the implementation of the algorithm. So in [68] the assumptions for the DOO algorithm are merged into a single one by discarding the semi-metric. In this chapter, we use the setting in [68] and make the following assumption where two parameters ν and ρ are introduced to directly relate f to the partitioning.

Assumption 5.1 *Given the partitioning of $\mathscr{X}$, let d_h^* be the index of the cell at depth h containing a global optimizer x^*, i.e., $x^* \in X^{h,d_h^*}$, and let x^{h,d_h^*} be the representative point of the cell X^{h,d_h^*}. Then there should exist $\nu > 0$ and $\rho \in (0,1)$ such that for any $h \in \{0,1,\ldots\}$, we have*

$$f(x^{h,d_h^*}) - f(x^*) \le \nu\rho^h.$$

The process of DOO is summarized in Figure 2.4. Adapting DOO for Assumption 5.1, at each iteration t, DOO selects a a leaf of the current tree with the minimum value $f(x^{h,d}) - \nu\rho^h$ to expand. Assumption 5.1 implies that any cell containing x^* satisfies

$$f(x^{h,d_h^*}) - \nu\rho^h \le f(x^*).$$

Consequently, a cell $X^{h',d'}$ such that

$$f(x^{h',d'}) - \nu\rho^{h'} > f(x^*),$$

will never be selected to split because there always exists a cell containing x^* such that

$$f(x^{h,d_h^*}) - \nu\rho^h < f(x^{h',d'}) - \nu\rho^{h'}.$$

More specifically, DOO only expands nodes of the set

$$I \triangleq \bigcup_{h\geq 0} I_h \ ,$$

where

$$I_h = \{(h,d) | f(x^{h,d}) - f(x^*) \leq \nu\rho^h\}.$$

The elements of I_h can be considered as $\nu\rho^h$-near-optimal solutions. A measure (called near-optimality dimension) is defined in [100] to characterize the number of near-optimal solutions and to derive bounds on the difference between the optimal solution and the solution returned by the algorithm. In this section, we adapt the definition of near-optimality dimension in [68] to make it equivalent to the definition in [100].

Definition 5.2 *The near-optimality dimension of f is the smallest $\eta > 0$ such that there exists a positive constant C such that the maximum number of cells $X^{h,d}$ at any depth h for which $f(x^{h,d}) - f(x^*) \leq \nu\rho^h$ is less than $C(\nu\rho^h)^{-\eta}$.*

With this near-optimality dimension, the results in [100] about bounds on the suboptimality still hold.

Theorem 5.3 *For a given finite number n of iterations, let x^* be a global minimizer and let $x(n)$ be the solution returned by the algorithm after n iterations.*
(i) Let $(h_{\max}, d_{\max})$ be the deepest node that has been expanded by the algorithm up to n iterations. Then we have

$$f(x(n)) - f(x^*) \leq \nu\rho^{h_{\max}}.$$

(ii) If $\eta > 0$, then

$$f(x(n)) - f(x^*) \leq \left(\frac{C}{1-\rho^\eta}\right)^{1/\eta} n^{-1/\eta}.$$

(iii) If $\eta = 0$, then

$$f(x(n)) - f(x^*) \leq \nu\rho^{n/C-1}.$$

Proof: (i) Since DOO only expands the nodes of the set I, we have

$$f(x^{h_{\max},d_{\max}}) - \nu\rho^{h_{\max}} \leq f(x^*).$$

Note that $x(n)$ is the returned solution with minimum function value of f among the expanded nodes, so

$$f(x(n)) \leq f(x^{h_{\max},d_{\max}}).$$

Since x^* is a global minimizer, we have

$$f(x^*) \leq f(x(n)).$$

Hence,

$$\begin{aligned} f(x(n)) - \nu\rho^{h_{\max}} &\leq f(x^{h_{\max},d_{\max}}) - \nu\rho^{h_{\max}} \\ &\leq f(x^*). \end{aligned}$$

Furthermore, $f(x(n)) - \nu\rho^{h_{\max}}$ and $f(x(n))$ are respectively a lower and an upper bound of $f(x^*)$. In addition, the distance between the two bounds is bounded by $\nu\rho^{h_{\max}}$.

(ii) From Definition 5.2, we have

$$|I_h| \leq C(\nu\rho^h)^{-\eta}.$$

Define an indicator function $\mathbf{1}_{I_h}(h,d)$ as: if (h,d) has been expanded, $\mathbf{1}_{I_h}(h,d) = 1$, else $\mathbf{1}_{I_h}(h,d) = 0$. When $\eta > 0$, the number of node expansions n satisfies

$$\begin{aligned} n &= \sum_{h=0}^{h_{\max}} \sum_{d=0}^{K^h-1} \mathbf{1}_{I_h}(h,d) \\ &\leq \sum_{h=0}^{h_{\max}} |I_h| \\ &\leq C\nu^{-\eta} \sum_{h=0}^{h_{\max}} (\rho^{-\eta})^h \\ &\leq C\nu^{-\eta} \frac{\rho^{-\eta(h_{\max}+1)} - 1}{\rho^{-\eta} - 1} \\ &\leq C\nu^{-\eta} \frac{\rho^{-\eta h_{\max}} - \rho^{\eta}}{1 - \rho^{\eta}} \\ &\leq C\nu^{-\eta} \frac{\rho^{-\eta h_{\max}}}{1 - \rho^{\eta}}. \end{aligned}$$

Thus, we have

$$(\nu\rho^{h_{\max}})^{\eta} \leq \frac{C}{n(1-\rho^{\eta})}.$$

Combined with (i), this yields,

$$f(x(n)) - f(x^*) \leq \left(\frac{C}{1-\rho^{\eta}}\right)^{1/\eta} n^{-1/\eta}.$$

(iii) When $\eta = 0$, we have

$$\begin{aligned} n &\leq C\nu^{-\eta} \sum_{h=0}^{h_{\max}} (\rho^{-\eta})^h \\ &\leq C(h_{\max} + 1). \end{aligned}$$

Thus, we have

$$h_{\max} \geq \frac{n}{C} - 1.$$

Since $\rho \in (0,1)$, we obtain

$$f(x(n)) - f(x^*) \leq \nu\rho^{h_{\max}} \leq \nu\rho^{n/C-1}.$$

□

5.4 Optimistic optimization of PWA functions

In this section, we first develop a partitioning approach for the polytopic feasible set

$$\mathcal{X} = \{x \in \mathbb{R}^{n_x} | Ax \leq b\}.$$

Standard partitioning works well for hypercubes which are one class of regular polytopes (having high degree of symmetry). However, the polytope considered in our problem can be irregular with arbitrary shape. So we need to divide the polytope into a collection of simplices and then subdivide each simplex into smaller simplices. There are many methods in literature to refine simplices. The method used in this chapter can to a great extent maintain the shape of the simplices and the volume of the refined simplices decrease with a fixed rate. Those properties of the partitioning scheme in this chapter allow us to develop expressions for the parameters of DOO.

Definition 5.4 (Simplex) *An m-simplex $\mathcal{S} \subset \mathbb{R}^n$ with $0 \leq m \leq n$ is the convex hull of $m+1$ affinely independent points $v_0, \dots, v_m \in \mathbb{R}^n$, which are its vertices. It can be written as*

$$\mathcal{S} = \Big\{ \sum_{i=0}^{m} \lambda_i v_i \Big| \lambda_i \geq 0, i = 0, \dots, m, \sum_{i=0}^{m} \lambda_i = 1 \Big\}.$$

If $m = n$, the set $\mathcal{S}$ is simply called a simplex of $\mathbb{R}^n$. Let $e_i = v_i - v_{i-1}$, $i = 1, \dots, n$. The n-dimensional volume of $\mathcal{S}$ is

$$\mathrm{vol}(\mathcal{S}) = \frac{1}{n!} \big| \det(e_1, e_2, \dots, e_n) \big|. \tag{5.4}$$

5.4.1 Hierarchical partition of a polytope

The partitioning of $\mathcal{X}$ consists of two stages: (i) dividing the polytope $\mathcal{X}$ into a collection of simplices; (ii) subdividing each simplex into smaller simplices. We propose to use Delaunay triangulation for the first stage and next to use edgewise subdivision repeatedly for the second stage.

Delaunay triangulation [31] divides a polytope into a mesh of high-quality simplices where high-quality means that in the division process, simplices with very short edges are created as little as possible. This property assures the regularity of the partitioning of $\mathcal{X}$.

Edgewise subdivision [51] divides a simplex $\mathcal{S}$ of $\mathbb{R}^{n_x}$ into k^{n_x} n_x-simplices, where each edge of $\mathcal{S}$ is cut into k equal pieces (k is an integer parameter that can be selected). A ready-to-implement algorithm for edgewise subdivision is presented in [64]. Below we present the definition and some properties of edgewise subdivision. Those properties are needed in the next section for the development of expressions for the parameters ν and ρ of DOO.

Definition 5.5 (Congruence class) *[13] Two non-degenerate[1] simplices $\mathcal{S}, \mathcal{S}'$ are called congruent to each other if there exists a translation vector $v \in \mathbb{R}^{n_x}$, a scaling factor $c > 0$, and an orthogonal matrix $Q \in \mathbb{R}^{n_x \times n_x}$ such that[2] $\mathcal{S}' = v + cQ\mathcal{S}$. In that case $\mathcal{S}$ and $\mathcal{S}'$ are said to*

[1] An m-simplex $\mathcal{S}$ is called degenerate if its m-dimensional volume equals 0.

[2] We define $v + cQ\mathcal{S} = \{v + cQx | x \in \mathcal{S}\}$.

Table 5.1: List of Symbols

$\mathcal{X}$	polytopic feasible set
$\{\mathcal{X}_s \mid s = 1, \ldots, N_t\}$	simplicial mesh of $\mathcal{X}$
k	subsimplices of edgewise subdivision shrinks by the factor $1/k$
n_x	dimension of $\mathcal{X}$
K	k^{n_x}, branching factor of optimistic optimization
h	subdivision depth
d	index of simplices at depth h
$X_s^{h,d}$	simplex at depth h of the edgewise subdivision of $\mathcal{X}_s$
$L_s^{h,d}$	maximum edge length of $X_s^{h,d}$
$r_s^{h,d}$	inradius of $X_s^{h,d}$
$x_s^{h,d}$	incenter of $X_s^{h,d}$
N_s	number of congruence classes of the edgewise subdivision of $\mathcal{X}_s$
$C_{s,i}$, $i = 1, \ldots, N_s$	representative simplices of the congruence classes
γ_s	ratio between the maximum and minimum volumes among the representative simplices for $\mathcal{X}_s$
$\rho_{s,i}$	inradius of $C_{s,i}$
ρ_s	minimum of $\rho_{s,i}$
$v_{s,0}, \ldots, v_{s,n}$	vertices of $\mathcal{X}_s$
$v_{s,0}^{h,d}, \ldots, v_{s,n}^{h,d}$	vertices of $X_s^{h,d}$

be elements of the same congruence class.

Properties of edgewise subdivision. For every integer $k \geq 1$, the edgewise subdivision of a simplex $\mathcal{S} \subset \mathbb{R}^{n_x}$ has the following properties [51]:

(i) all generated simplices have the same n_x-dimensional volume;

(ii) all generated simplices fall into at most $n_x!/2$ congruence classes;

(iii) the faces of $\mathcal{S}$ are subdivided with the same k and the same method;

(iv) repeated subdivision has the same effect as increasing k.

The property (iv) that repeated subdivision has the same effect as increasing k, means that instead of dividing an n_x-simplex $\mathcal{S}$ into k^{n_x} n_x-simplices and subsequently subdividing each subsimplex into l^{n_x} n_x-simplices, we can subdivide $\mathcal{S}$ into $(kl)^{n_x}$ n_x-simplices and reach the same result.

5.4.2 PWA optimistic optimization

In this section, for the partitioning approach given in Section 5.4.1, we develop analytic expressions for the parameters ν and ρ satisfying Assumption 5.1 for applying DOO to solve the problem (5.1)-(5.2). Some of the symbols that occur frequently in this section are listed in Table 5.1.

By performing Delaunay triangulation, the feasible set $\mathcal{X}$ is divided into a mesh of simplices $\{\mathcal{X}_s \mid s = 1, \ldots, N_t\}$. Every simplex $\mathcal{X}_s$ in the simplicial mesh is taken as the original simplex on which repeated edgewise subdivision is performed. Properties (i)-(iv) of edgewise subdivision given in Section 5.4.1 are essential for the proofs in the rest of this section. For any integer $k \geq 1$, edgewise subdivision divides $\mathscr{X}_s$ into k^{n_x} n_x-simplices; so the maximum number K of child cells of a parent cell equals k^{n_x}.

Note that $h \in \{0, 1, \ldots\}$ is the depth of the subdivision (indicator of the recursion of edgewise subdivision) and $d \in \{0, \ldots, K^h - 1\}$ is the index of a simplex at a given depth h. Let $X_s^{h,d}$ be a simplex at depth h generated by repeated edgewise subdivision of $\mathscr{X}_s$. Let $L_s^{h,d}, r_s^{h,d}, x_s^{h,d}$ be the maximum edge length, inradius (i.e., the radius of the inscribed hyper-ball) and incenter (i.e., the center of the inscribed hyper-ball) of $X_s^{h,d}$. Let $N_\mathrm{c} \leq n_x!/2$ be the number of congruence classes that all simplices generated by repeated edgewise subdivision of $\mathscr{X}_s$ fall into (see Property (ii)). Note that the simplices in each congruence class are the same up to translation, scaling, and rotation. Let $C_{s,i}$, $i = 1, \ldots, N_\mathrm{c}$, be a representative simplex[3] of each congruence class. Define the ratio between the maximum and minimum volumes among the representative simplices for $\mathscr{X}_s$ as

$$\gamma_s = \max_{i,j=1,\ldots,N_\mathrm{c}} \frac{\mathrm{vol}(C_{s,i})}{\mathrm{vol}(C_{s,j})} \quad . \tag{5.5}$$

Let $\tau_{s,i}$ be the inradius of $C_{s,i}$ and denote

$$\tau_s = \min_{i=1,\ldots,N_\mathrm{c}} \tau_{s,i} \quad . \tag{5.6}$$

Let $v_{s,0}, \ldots, v_{s,n_x}$ be the vertices of $\mathscr{X}_s$. Let $v_{s,0}^{h,d}, \ldots, v_{s,n_x}^{h,d}$ be the vertices of $X_s^{h,d}$. Define

$$\begin{aligned} e_{s,i} &= v_{s,i} - v_{s,i-1} \quad , \\ e_{s,i}^{h,d} &= v_{s,i}^{h,d} - v_{s,i-1}^{h,d} \quad , \end{aligned}$$

with $i = 1, \ldots, n_x$. Then taking into account the proof of the independence lemma in [51] as well as the fact that repeated subdivision has the same effect as increasing k (see Property (iv)), there exists a permutation $\pi_s^{h,d}$ of $\{1, \ldots, n_x\}$ such that

$$e_{s,i}^{h,d} = \frac{1}{k^h} e_{s,\pi_s^{h,d}(i)} \quad .$$

Note that we have

$$v_{s,i}^{h,d} - v_{s,0}^{h,d} = e_{s,i}^{h,d} + e_{s,i-1}^{h,d} + \cdots + e_{s,1}^{h,d} \quad . \tag{5.7}$$

Now select an arbitrary edge of $X_s^{h,d}$ and let $v_{s,i}^{h,d}$ and $v_{s,j}^{h,d}$ with $j > i$ be the corresponding vertices. By (5.7), we have

$$\begin{aligned} \left| v_{s,j}^{h,d} - v_{s,i}^{h,d} \right| &= \left| e_{s,j}^{h,d} + e_{s,j-1}^{h,d} + \cdots + e_{s,i+1}^{h,d} \right| \\ &= \frac{1}{k^h} \left| e_{s,\pi_s^{h,d}(j)} + e_{s,\pi_s^{h,d}(j-1)} + \cdots + e_{s,\pi_s^{h,d}(i+1)} \right|. \end{aligned}$$

[3] A representative simplex of a congruence class is defined here as the simplex resulting from scaling any simplex in the class such that its maximum edge length equals 1.

Define

$$\theta_{s,\min} = \min_{i=1,\ldots,n_x} |e_{s,i}| \ , \qquad \theta_{s,\max} = \sum_{i=1}^{n_x} |e_{s,i}| \ . \tag{5.8}$$

Note that $\theta_{s,\min} > 0$. Then we have

$$\frac{1}{k^h}\theta_{s,\min} \leq \left| v_{s,j}^{h,d} - v_{s,i}^{h,d} \right| \leq \frac{1}{k^h}\theta_{s,\max} \ . \tag{5.9}$$

Lemma 5.6 *Denote*

$$L_{s,h} = \max_{d \in D_h} L_s^{h,d},$$

and

$$r_{s,h} = \min_{d \in D_h} r_s^{h,d},$$

where $D_h = \{0, \ldots, K^h - 1\}$ is the index set of simplices at depth h. Then we have

$$\frac{L_{s,h+1}}{L_{s,h}} \leq \frac{1}{k}\gamma_s^{1/n_x} \ , \qquad \frac{r_{s,h}}{L_{s,h}} \geq \frac{\theta_{s,\min}\tau_s}{\theta_{s,\max}} \ , \tag{5.10}$$

where γ_s, τ_s, $\theta_{s,\min}$ and $\theta_{s,\max}$ are as defined in (5.5), (5.6), (5.8) *and $1/k$ is the factor of edgewise subdivision.*

Proof: Let $X_s^{h,d'}$ be the simplex that has the maximum edge length $L_{s,h}$ among all simplices at depth h and assume that $X_s^{h,d'}$ belongs to congruence class i with a representative simplex $C_{s,i}$. By definition the maximum edge length of $C_{s,i}$ equals 1.

From Property (iv), repeated subdivision is equivalent to increasing k; so a division at depth h actually corresponds to selecting k^h instead of k. Moreover, from Property (i), we have

$$\text{vol}(X_s^{h,d'}) = \frac{\text{vol}(\mathscr{X}_s)}{k^{hn_x}}.$$

Scaling $X_s^{h,d'}$ with a factor $1/L_{s,h}$ scales every column in the matrix of which the determinant is taking in the volume formula (5.4), resulting in a multiplication with $(1/L_{s,h})^{n_x}$ compared to the original expression. Hence, we have

$$\text{vol}(C_{s,i}) = \left(\frac{1}{L_{s,h}}\right)^{n_x} \text{vol}(X_s^{h,d'}) = \left(\frac{1}{L_{s,h}}\right)^{n_x} \frac{\text{vol}(\mathscr{X}_s)}{k^{hn_x}}. \tag{5.11}$$

Likewise let $X_s^{h+1,d''}$ be the simplex that has the maximum edge length $L_{s,h+1}$ among all simplices at depth $h+1$ and assume that $X_s^{h+1,d''}$ belongs to congruence class j with a representative simplex $C_{s,j}$. So

$$\text{vol}(C_{s,j}) = \left(\frac{1}{L_{s,h+1}}\right)^{n_x} \frac{\text{vol}(\mathscr{X}_s)}{k^{(h+1)n_x}}. \tag{5.12}$$

Thus (5.11) and (5.12) result in

$$\left(\frac{L_{s,h+1}}{L_{s,h}}\right)^{n_x} = \frac{1}{k^{n_x}} \frac{\text{vol}(C_{s,i})}{\text{vol}(C_{s,j})}, \tag{5.13}$$

and thus

$$\frac{L_{s,h+1}}{L_{s,h}} = \frac{1}{k}\Big(\frac{\mathrm{vol}(C_{s,i})}{\mathrm{vol}(C_{s,j})}\Big)^{1/n_x} \le \frac{1}{k}\gamma_s^{1/n_x} \quad .$$

This completes the proof of the first inequality in (5.10).

Let $X_s^{h,d^\sharp}$ be the simplex that has the shortest inradius $r_{s,h}$ among all simplices at depth h and assume that $X_s^{h,d^\sharp}$ belongs to congruence class l with a representative simplex $C_{s,l}$. The maximum edge length of $C_{s,l}$ equals 1 and the inradius of $C_{s,l}$ is $\tau_{s,l}$. Thus, we have

$$r_{s,h} = L_s^{h,d^\sharp}\tau_{s,l} \quad .$$

Due to (5.6), we also have

$$r_{s,h} \ge L_s^{h,d^\sharp}\tau_s \quad .$$

Note that (5.9) implies that

$$\frac{1}{k^h}\theta_{s,\min} \le L_s^{h,d} \le \frac{1}{k^h}\theta_{s,\max} \quad , \forall d \in D_h \quad .$$

Hence,

$$r_{s,h} \ge L_s^{h,d^\sharp}\tau_s \ge \frac{1}{k^h}\theta_{s,\min}\tau_s \quad ,$$

and thus

$$\frac{r_{s,h}}{L_{s,h}} \ge \frac{\frac{1}{k^h}\theta_{s,\min}\tau_s}{L_{s,h}} \ge \frac{\frac{1}{k^h}\theta_{s,\min}\tau_s}{\frac{1}{k^h}\theta_{s,\max}} \ge \frac{\theta_{s,\min}\tau_s}{\theta_{s,\max}} \quad .$$

This completes the proof. □

Theorem 5.7 *Denote*

$$\alpha = \max_{i=1,\ldots N} \|\alpha_{(i)}\|_2 \quad ,$$

and

$$\nu_s = \alpha L_{s,0} \quad , \qquad \rho_s = \frac{L_{s,h+1}}{L_{s,h}} \quad ,$$

where $L_{s,h}$ is as defined in Lemma 5.6. Let

$$\nu = \max_{s=1,\ldots,N_t} \nu_s \quad , \qquad \rho = \max_{s=1,\ldots,N_t} \rho_s \quad .$$

If k is selected as an integer that is strictly larger than $\max_{s=1,\ldots N_t} \gamma_s^{1/n_x}$, then for any cell X^{h,d_h^*} that contains a global optimizer x^* with the incenter selected as the representative point x^{h,d_h^*} of the cell X^{h,d_h^*}, we have $\nu > 0$, $\rho \in (0,1)$, and

$$f(x^{h,d_h^*}) - f(x^*) \le \nu\rho^h.$$

Proof: From (5.13), we can conclude that $\rho_s = \frac{L_{s,h+1}}{L_{s,h}}$ does not depend on h. Note that with the given definitions of ν and ρ, they are naturally positive constants. Moreover, if k is selected as an integer that is strictly larger than $\max_{s=1,\ldots N_t}\gamma_s^{1/n_x}$, then, from Lemma 5.6, for any s, we have

$$\rho_s \le \frac{1}{k}\gamma_s^{1/n_x} < 1 \quad .$$

So $\nu > 0$ and $\rho \in (0,1)$. Assume that x^* is contained in a cell X_s^{h,d_h^*} and the incenter of X_s^{h,d_h^*} is selected as the representative point x_s^{h,d_h^*}. Then we have

$$\begin{aligned} f(x_s^{h,d_h^*}) - f(x^*) &\overset{(5.3)}{\leq} \alpha \| x_s^{h,d_h^*} - x^* \|_2 \\ &\leq \alpha L_s^{h,d_h^*} \\ &\leq \alpha L_{s,h} \ . \end{aligned}$$

From $\rho_s = \frac{L_{s,h+1}}{L_{s,h}}$, we have

$$L_{s,h} = (\rho_s)^h L_{s,0} \ . \tag{5.14}$$

Thus,

$$f(x_s^{h,d_h^*}) - f(x^*) \leq \alpha (\rho_s)^h L_{s,0} \ .$$

From $\nu_s = \alpha L_{s,0}$, we have

$$f(x_s^{h,d_h^*}) - f(x^*) \leq \nu_s (\rho_s)^h.$$

Let $\nu = \max_{s=1,\ldots,N_t} \nu_s$ and $\rho = \max_{s=1,\ldots,N_t} \rho_s$. Therefore, for any cell X^{h,d_h^*} that contains x^*, we have

$$f(x^{h,d_h^*}) - f(x^*) \leq \nu \rho^h.$$

This completes the proof. $\square$

Lemma 5.8 *Let*

$$\sigma = \mu_1 \mu_2 \min_{s=1,\ldots,N_t} \sigma_s \ ,$$

where

$$\mu_1 = \min_{s',s''=1,\ldots,N_t} \frac{L_{s'',0}}{L_{s',0}} \ , \quad \mu_2 = \min_{s,s''=1,\ldots,N_t} \frac{L_{s,h}}{L_{s'',h}} \ ,$$

and σ_s is a positive constant such that

$$0 < \sigma_s \leq \frac{\tau_s \theta_{s,\min}}{\alpha \theta_{s,\max}}.$$

Then any cell $X^{h,d}$ at any depth h contains a ball of radius $\sigma \nu \rho^h$ centered in $x^{h,d}$, denoted as

$$\mathfrak{B}(x^{h,d}, \sigma \nu \rho^h) = \left\{ x \in \mathscr{X} \middle| \| x - x^{h,d} \|_2 \leq \sigma \nu \rho^h \right\} \subset X^{h,d},$$

where ν and ρ are defined as in Theorem 5.7.

Proof: First we prove that μ_2 is independent of h. Similar to the proof of Lemma 5.6, we get

$$\begin{aligned} \mathrm{vol}(X_s^{h,d'}) &= \frac{\mathrm{vol}(\mathscr{X}_s)}{k^{hn_x}} \\ &= (L_{s,h})^{n_x} \mathrm{vol}(C_{s,i}), \quad i \in \{1, \ldots, N_s\}, \end{aligned}$$

and

$$\mathrm{vol}(X_{s''}^{h,d''}) = \frac{\mathrm{vol}(\mathscr{X}_{s''})}{k^{hn_x}}$$

$$= (L_{s'',h})^{n_x} \operatorname{vol}(C_{s'',j}), \quad j \in \{1, \ldots, N_{s''}\}.$$

Hence, we have

$$\frac{L_{s,h}}{L_{s'',h}} = \Big(\frac{\operatorname{vol}(X_s^{h,d'}) \operatorname{vol}(C_{s'',j})}{\operatorname{vol}(X_{s''}^{h,d''}) \operatorname{vol}(C_{s,i})}\Big)^{1/n_x},$$

which is independent of h. So μ_2 is independent of h.

Now, we prove that $\sigma \nu \rho^h \le \sigma_s \nu_s (\rho_s)^h$ for any $s = 1, \ldots, N_t$, where ν_s and ρ_s are defined as in Theorem 5.7. The inequality to be proved is rewritten as $\sigma \le \sigma_s \frac{\nu_s(\rho_s)^h}{\nu \rho^h}$. Let s' and s'' denote the indices such that $\nu_{s'} = \max\limits_{1,\ldots,N_t} \nu_s$ and $\rho_{s''} = \max\limits_{1,\ldots,N_t} \rho_s$. Thus we have $\nu = \nu_{s'}$, $\rho = \rho_{s''}$, and

$$\begin{aligned}
\frac{\nu_s(\rho_s)^h}{\nu \rho^h} &= \frac{\nu_s(\rho_s)^h}{\nu_{s'}(\rho_{s''})^h} \\
&= \frac{\alpha L_{s,0}(\rho_s)^h}{\alpha L_{s',0}(\rho_{s''})^h} \\
&= \frac{L_{s,h}}{L_{s',0}(\rho_{s''})^h} \\
&= \frac{L_{s'',0} L_{s,h}}{L_{s',0} L_{s'',0}(\rho_{s''})^h} \\
&= \frac{L_{s'',0} L_{s,h}}{L_{s',0} L_{s'',h}} \\
&\ge \mu_1 \mu_2 \ .
\end{aligned}$$

If $\sigma = \mu_1 \mu_2 \min\limits_{s=1,\ldots,N_t} \sigma_s$, then we have $\sigma \nu \rho^h \le \sigma_s \nu_s (\rho_s)^h$ for any $s = 1, \ldots, N_t$.

Finally, we prove that $\mathfrak{B}(x^{h,d}, \sigma \nu \rho^h) \subset X^{h,d}$. For any $x \in \mathfrak{B}(x^{h,d}, \sigma \nu \rho^h)$, we have

$$\begin{aligned}
\|x - x^{h,d}\| &\le \sigma \nu \rho^h \\
&\le \sigma_s \nu_s (\rho_s)^h \\
&\le \frac{\tau_s \theta_{s,\min}}{\alpha \theta_{s,\max}} \alpha L_{s,0} (\rho_s)^h \\
&\overset{(5.14)}{\le} \frac{\tau_s \theta_{s,\min}}{\theta_{s,\max}} L_{s,h} \\
&\overset{(5.10)}{\le} r_{s,h} \ ,
\end{aligned}$$

where $r_{s,h}$ defined in Lemma 5.6 is the minimum among the inradii of simplices at depth h. Therefore, $x \in \mathfrak{B}(x^{h,d}, \sigma \nu \rho^h)$ implies that $x \in X^{h,d}$. This completes the proof. □

Theorem 5.7 gives analytic expressions for the parameters ν and ρ required by DOO. Lemma 5.8 guarantees that the subsimplices generated by the developed partitioning approach do not become too slim with very short edges.

Remark 5.9 In Theorem 5.7, the parameter α requires the knowledge of the Lipschitz constants of the PWA function f. Actually, it may not always be possible to find the smallest Lipschitz constant of a general PWA objective function. In this case, an upper bound on the Lipschitz constants is also acceptable, but note that a larger α results in a larger ν and consequently results in a larger number of cells such that $f(x^{h,d}) - f(x^*) \leq \nu\rho^h$. As a result, the algorithm may waste time on exploring too many unnecessary cells, which will lower the degree of optimality of the resulting solution for the predefined computational budget. □

In Section 5.2 of [132], it is shown that functions defined over a finite-dimensional and bounded space $\mathscr{X}$ have a near-optimality dimension equal to 0 if the functions have an upper and lower envelope around one global maximizer x^* of the same order, i.e., there exists constants $c \in (0,1)$ and $\delta > 0$, such that for all $x \in \mathscr{X}$:

$$\min(\delta, c\ell(x, x^*)) \leq f(x^*) - f(x) \leq \ell(x, x^*), \tag{5.15}$$

where ℓ is a semi-metric. Clearly, the condition (5.15) is satisfied by the continuous nonconvex PWA functions considered in this chapter. So the near-optimality dimension in our problem is equal to 0.

5.5 Examples

In this section, we evaluate the optimistic optimization approach and compare it with other methods.

Example 6.1

The instances considered include 60 randomly generated continuous PWA functions $f : \mathbb{R}^2 \to \mathbb{R}$ in which the vector pairs $\alpha_{(i)} \in \mathbb{R}^2$, $\beta_{(i)} \in \mathbb{R}$ contain pseudorandom values drawn from the standard normal distribution $\mathscr{N}(0,1)$ with $i = 1, \ldots, N$ where N is also random.

Below we compare the efficiency of the DOO algorithm, the MILP method, and the DIRECT algorithm [83]. DIRECT is a Lipschitz algorithm not requiring the knowledge of the Lipschitz constant. It uses an optimistic splitting technique similar to the optimistic optimization algorithm.

The corresponding MILP problem is derived based on the techniques in [8] and solved with the `intlinprog` function in the Matlab Optimization Toolbox and the `cplex` function called using the Tomlab interface toolbox for Matlab. DOO is implemented as a function in Matlab (called `pwadoo`). Note that `pwadoo` and `intlinprog` are both Matlab functions and `cplex` is implemented in object code, which implies that it will in general run much faster than a equivalent program written in Matlab. DIRECT is performed using the `glbDirect` solver in Tomlab and is implemented in object code.

Figure 5.1 shows the semi logarithmic plot of CPU time (average over 10 runs) for different solvers as a function of N. The function values of f returned from different solvers are denoted as f_{int}, f_{cpl}, f_{oo}, and f_{dir}, where f_{int} and f_{cpl} of every instance are equal. The iteration in `pwadoo` (`glbDirect`) is stopped if the gap between f_{cpl} and f_{oo} (f_{dir}) is less than 5% (the gap is calculated as $100|(f_{\text{oo}} - f_{\text{cpl}})/f_{\text{cpl}}|$ and $100|(f_{\text{dir}} - f_{\text{cpl}})/f_{\text{cpl}}|$). We can see that `pwadoo` is faster than `intlinprog` and even `cplex` for 80% of the instances. Figure 5.2 shows the relative error of `pwadoo` and `glbDirect` given different number of iterations for all 60 PWA function instances. We can see that the rate of convergence of `pwadoo` is slower than `glbDirect`. This is because the Lipschitz constant is used in the DOO algorithm.

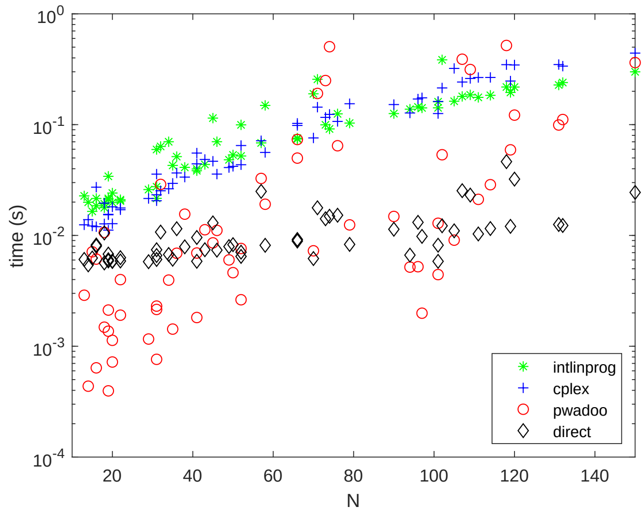


Figure 5.1: CPU time of `intlinprog`, `cplex`, `pwadoo` and `glbDirect` for the optimization of PWA functions (N is the number of polyhedral subregions of PWA functions)

The experiments show that DOO finds an approximation solution close to the optimal solution requiring computation time less than that of the MILP solvers taking to find the optimal solution. Hence, we propose to use DOO instead of the MILP method to solve the optimization problem of the PWA function for the case that N is much larger than the dimension of the feasible set.

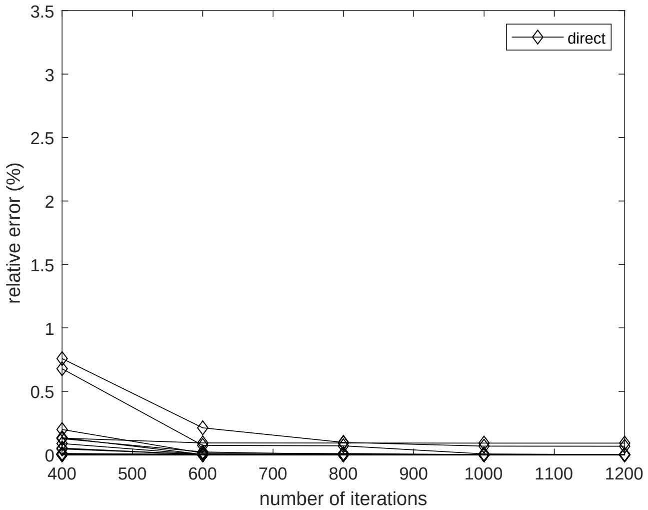
Example 6.2

Any continuous PWA function can be represented as a min-max or max-min composition of its affine components [108], which is similar to the canonical form of max-min-plus-scaling (MMPS) functions. As presented in [48], the optimization of MMPS functions can be written as a finite set of LP problems where the worst-case complexity is largely determined by the number of affine terms in equivalent canonical form of the MMPS expression. We consider an MMPS function written as

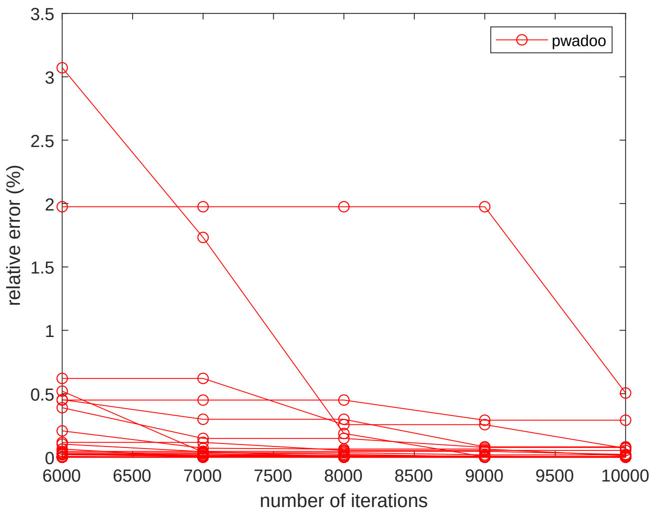
$$g(x) = \min_{i=1\ldots M} \max_{j=1\ldots M} \{\alpha_{(i,j)}^T x + \beta_{(i,j)}\}, \quad \forall x \in \mathbb{R}^2,$$

where $\alpha_{ij} \in \mathbb{R}^2, \beta_{ij} \in \mathbb{R}$ contains pseudorandom values drawn from the standard normal distribution. We use the `linprog` function of Tomlab to solve the set of LPs resulting from the minimization problem of g. The optimistic optimization approach is implemented as a function in Matlab (called `mmpsdoo`).

Figure 5.3 shows the semi-logarithmic plot of the CPU time (average over 10 runs) of the LP approach and the optimistic optimization approach for increasing n. The gap between the function value g_{lp} returned by `linprog` and g_{oo} returned by `mmpsdoo` is restricted to 5% (the gap is calculated as $100|(g_{\text{lp}} - g_{\text{oo}})/g_{\text{lp}}|$). We can see that using the `mmpsdoo` function is more efficient than solving a sequence of LPs.



number of iterations	400	600	800	1000	1200
CPU time (s)	0.0633	0.0922	0.1259	0.1610	0.1965



number of iterations	6000	7000	8000	9000	10000
CPU time (s)	0.5365	0.6909	0.8685	1.0663	1.2954

Figure 5.2: Relative error and average CPU times of `pwadoo` and `glbDirect` for all 60 PWA function instances

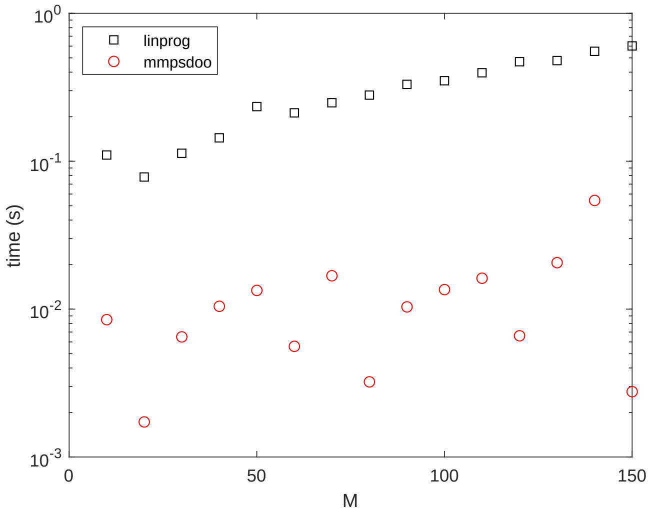


Figure 5.3: CPU time of linprog and optimistic optimization (`mmpsdoo`) for the optimization of MMPS functions (M is the number of max and min operations of MMPS functions)

5.6 Conclusions

In this chapter, we have considered the optimization of a continuous nonconvex PWA function over a polytope. We have proposed an optimistic-optimization-based approach to solve the given problem. In particular, by employing Delaunay triangulation and edgewise subdivision, we have constructed a partition of the feasible set satisfying the requirements for optimistic optimization. We have also derived the analytic expressions for the core parameters. Numerical examples have been implemented to test the proposed approach. Compared with the MILP based methods, the optimistic-optimization-based approach is more efficient especially for large problems.

The partitioning scheme developed in this chapter is the only way we have found currently satisfying all the requirements of optimistic optimization. In the future, we will search for other suitable partitioning schemes. In addition, the proposed algorithm is formulated in a deterministic setting. We will also investigate a stochastic setting. Furthermore, we will investigate the performance of the optimistic optimization algorithms which does not require the knowledge of the Lipschitz constant, such as the simultaneous optimistic optimization (SOO) algorithm, for solving the optimization problem of PWA functions.

Chapter 6

MPC for stochastic MPL systems with chance constraints

In this chapter we consider model predictive control for max-plus linear systems with stochastic uncertainties the distribution of which is supposed to be known. We consider linear constraints on the inputs and the outputs. Due to the uncertainties, these linear constraints are formulated as probabilistic or chance constraints, i.e., the constraints are required to be satisfied with a predefined probability level. Two methods based on Boole's inequality and Chebyshev's inequality respectively are introduced to transform the chance constraint into a reduced form that can be evaluated efficiently. The simulation results for a production system example show that the two proposed methods are faster than a Monte Carlo simulation method and yield lower closed-loop costs than the nominal model predictive control method.

6.1 Introduction

Due to model mismatch or disturbances, uncertainties are often considered in the prediction model of model predictive contro (MPC). Many results have been achieved in the area of robust MPC dealing with the situation that the uncertainties are assumed to be deterministic and bounded, see e.g., [9, 97] and the references therein. On the other hand, for the situation that the uncertainties are characterized as random variables, stochastic MPC [54, 99] has emerged as a useful control design method where usually the expected value of a cost criterion is optimized subject to input, state, or output constraints. Due to the probabilistic nature of the uncertainties, those constraints are usually formulated as chance constraints, i.e., the probability of constraint violation is limited to a predefined probability level. Stochastic MPC takes advantage of the knowledge of the probability distributions of the uncertainties and is based on stochastic programming and chance-constrained programming [25, 30, 53, 142].

In contrast to conventional linear systems, where uncertainties are usually modelled by adding an extra term in the system equations, uncertainties in max-plus linear (MPL) systems are usually included in the system matrices [7]. The MPC framework has been extended to stochastic max-plus linear (SMPL) systems in [134]. The expected value of the outputs is used in the objective criterion and in the constraint. Some results about MPC for SMPL systems can be found in [52, 120, 137, 138]. To the author's best knowledge currently [120] is the only result in literature that has considered the chance-constrained MPC problem for SMPL systems. In [120], the chance constraints are approximated and substituted with a finite number of pointwise constraints at independently generated scenarios of the uncertainties. The approach in [120] is different from the methods developed in this chapter as we transform the chance constraints into a reduced form based on some probabilistic inequalities.

In particular, in this chapter we develop approaches for solving the chance-constrained MPC problem based on probabilistic inequalities and properties of SMPL systems. More specifically, if the chance constraints are monotonically nondecreasing as a function of the outputs (i.e., the coefficients of the outputs in the linear constraints are nonnegative), we rewrite the chance constraints into an equivalent max-affine form, namely, the maximum of some correlated random variables. Those correlated random variables are affine functions of the uncertainties of the SMPL system. Based on the resulting max-affine form, we develop two methods for transforming the chance constraints into a reduced form. In the first method, based on Boole's inequality, the probability of the maximum of correlated random variables is decomposed into the sum of probabilities of a single random variable. In the second method, we provide sufficient conditions for applying the multidimensional Chebyshev inequality to transform the chance constraints into constraints that are linear in the control inputs. The approaches developed in this chapter are assessed with a production system example and compared with a Monte Carlo (MC) simulation method and the nominal MPC method. The results show that the two proposed methods generally take less computation time than the MC simulation method to achieve a similar performance. The nominal MPC method is faster than the other methods, but it yields a worse performance.

This chapter is organized as follows. Section 6.2 provides preliminaries about p-norms, probabilistic inequalities and the definition of max-affine functions. A brief introduction to SMPL systems is given in Section 6.3. The MPC problem formulation with chance constraints for SMPL systems is presented in Section 6.4. Two approaches for solving the proposed problem are developed in Section 6.5 and illustrated with a production system example in Section 6.6. Finally, Section 6.7 concludes the chapter.

6.2 Probabilistic inequalities

This section is based on [34, 82, 119].

Definition 6.1 (Joint probability distribution) *Let $X = [X_1, \dots, X_n]^T$ be a random vector and $x = [x_1, \dots, x_n]^T$ be a realization of X. Let f_X be the joint probability density function of X. For any set $D \in \mathbb{R}^n$, the probability that a realization of X falls inside D is then*

$$\Pr\{X \in D\} = \int_D f_X(x_1, \dots, x_n)\,\mathrm{d}x_1 \cdots \mathrm{d}x_n.$$

Theorem 6.2 (Boole's inequality) *For a countable collection of events $A_1, A_2, A_3, \dots$, we have*[1]

$$\Pr\Big(\bigcup_i A_i\Big) \le \sum_i \Pr(A_i). \tag{6.1}$$

[1] The event $A_1 \bigcup A_2$ would occur if A_1 or A_2 occurs.

Definition 6.3 (Expected value) *Let f_X be the joint probability density function of a random vector $X = [X_1, \ldots, X_n]^T$. The expected value of a function g of X is defined as*

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(x) f_X(x_1, \ldots, x_n)\,\mathrm{d}x_1 \cdots \mathrm{d}x_n,$$

provided that $\mathbb{E}[|g(X)|] \le \infty$.

Theorem 6.4 (Jensen's inequality) *Let φ be an integrable, concave function of a random variable v. Then*

$$\mathbb{E}[\varphi(v)] \le \varphi(\mathbb{E}[v]). \tag{6.2}$$

Theorem 6.5 (Multidimensional Chebyshev inequality) *Let $X = [X_1, \ldots, X_n]^T$ be a random vector with mean $\mu_X = \mathbb{E}[X]$ and covariance matrix $\Sigma_X = \mathbb{E}[(X-\mu)(X-\mu)^T]$. If Σ_X is positive definite, then for any $a > 0$ we have*

$$\Pr\left\{(X-\mu_X)^T \Sigma_X^{-1}(X-\mu_X) \le a\right\} \ge 1 - \frac{n}{a}. \tag{6.3}$$

Theorem 6.6 *Let X be a random vector with mean μ_X and covariance matrix Σ_X. Let $B \in \mathbb{R}^{m\times n}$ be a real matrix. Then the linear combination $Y = BX$ satisfies*

$$\begin{aligned}
\mu_Y &= \mathbb{E}[Y] = \mathbb{E}[BX] = B\mu_X,\\
\Sigma_Y &= \mathrm{Cov}(Y) = \mathrm{Cov}(BX) = B\Sigma_X B^T.
\end{aligned}$$

Definition 6.7 (Max-affine function) *A max-affine function f of $x \in \mathbb{R}_\varepsilon^n$ is a function of the form*

$$f(x) = \max_{i=1,\ldots,n} (\alpha_i^T x + \xi_i),$$

with constant coefficients $\alpha_i \in \mathbb{R}^n$ and $\xi_i \in \mathbb{R}$.

6.3 Stochastic MPL systems

The random vector $w(k) \in \mathbb{R}^{n_w}$ collects uncertainties at event step k caused by disturbances or model mismatch. Just as in [137] we adopt the following assumption in this chapter:

Assumption 6.8 *At any event step k, the components of $w(k)$ are independent and identically distributed random variables with a given probability distribution, i.e., $\{w_i(k) : i\}$ is a collection of i.i.d. random variables. In addition, the uncertainties at different event steps are independent, i.e., $w(0), w(1)\ldots$ are mutually statistically independent.*

Consider a stochastic max-plus linear (SMPL) system [134] of the form

$$x(k) = A(w(k)) \otimes x(k-1) \oplus B(w(k)) \otimes u(k), \tag{6.4}$$

$$y(k) = C(w(k)) \otimes x(k), \tag{6.5}$$

where k is the event counter, $u(k) \in \mathbb{R}_\varepsilon^{n_u}$ and $y(k) \in \mathbb{R}_\varepsilon^{n_y}$ are the input and output of the system consisting of the time instants at which the input and output events occur for the k-th cycle, and $x(k) \in \mathbb{R}_\varepsilon^{n_x}$ is the state of the system representing the time instants at which the internal processes of the system start for the k-th cycle.

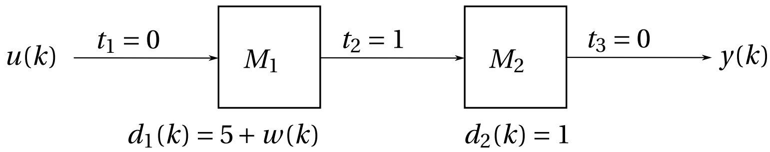


Figure 6.1: A production system

Typically, the entries of the uncertain system matrices $A(w(k))$, $B(w(k))$, and $C(w(k))$ consist of sums of internal process times and transportation times [7]. In general, the components of $w(k)$ correspond to perturbations in these process and transportation times. Due to the possibility of breakdown or delay of machines and transporters, the process and transportation times might be disturbed by uncertainties. Since the machines and the transport systems in the production system usually work independently, the uncertainties occurring at different machines and transporters are usually independent. For the sake of simplicity, we assume that the uncertainties at the current cycle do not influence the uncertainties at the next cycle. Instead of modeling uncertainties by adding an extra max-plus-algebraic term in (6.4) and (6.5), uncertainties should rather be modeled as an additive term to these system matrices. Then, the entries of the uncertain system matrices are max-affine functions of $w(k)$.

As an example, we consider the production system presented in [134] (see Figure 6.1). This system consists of two machines M_1 and M_2 where raw materials are fed into M_1, afterwards intermediate products are fed into M_2, and finally the finished goods leave the production system. Just as in [134] we assume that the transportation times are constant (i.e., $t_1 = t_3 = 0, t_2 = 1$) and so is the processing time of M_2 (i.e., $d_2(k) = 1$). Here $x_i(k)$ represents the time instant at which machine i starts for the k-th time. The system matrices of the corresponding SMPL model are given as follows:

$$A = \begin{bmatrix} d_1(k-1) & \varepsilon \\ d_1(k-1) + d_1(k) + t_2(k) & d_2(k-1) \end{bmatrix},$$

$$B = \begin{bmatrix} t_1(k) \\ d_1(k) + t_1(k) + t_2(k) \end{bmatrix}, \quad C = \begin{bmatrix} \varepsilon & d_2(k) + t_3(k) \end{bmatrix}.$$

6.4 MPC for stochastic MPL systems

In [134], the MPC framework has been extended to SMPL systems in which the expected value of the outputs is used in the objective function and in the constraints. In this section, we give a brief introduction to this framework and formulate the MPC problem using chance constraints instead of using expected value of the outputs in the constraints.

6.4.1 Prediction of future outputs

Define

$$\tilde{u}(k) = \begin{bmatrix} u^T(k) & \cdots & u^T(k+N_\mathrm{p}-1) \end{bmatrix}^T,$$
$$\tilde{y}(k) = \begin{bmatrix} y^T(k) & \cdots & y^T(k+N_\mathrm{p}-1) \end{bmatrix}^T,$$
$$\tilde{w}(k) = \begin{bmatrix} w^T(k-1) & \cdots & w^T(k+N_\mathrm{p}-1) \end{bmatrix}^T,$$

where N_p is the prediction horizon. By using successive substitution on (6.4)-(6.5), the prediction of the future outputs is given by

$$\tilde{y}(k) = \tilde{C}(\tilde{w}(k)) \otimes x(k-1) \oplus \tilde{D}(\tilde{w}(k)) \otimes \tilde{u}(k).$$

The detailed expressions of $\tilde{C}(\tilde{w}(k))$ and $\tilde{D}(\tilde{w}(k))$ are given by [134]:

$$\tilde{C}(\tilde{w}(k)) = \begin{pmatrix} \tilde{C}_1(\tilde{w}(k)) \\ \vdots \\ \tilde{C}_{N_\mathrm{p}}(\tilde{w}(k)) \end{pmatrix},$$

$$\tilde{D}(\tilde{w}(k)) = \begin{pmatrix} \tilde{D}_{11}(\tilde{w}(k)) & \cdots & \tilde{D}_{1N_\mathrm{p}}(\tilde{w}(k)) \\ \vdots & \ddots & \vdots \\ \tilde{D}_{N_\mathrm{p}1}(\tilde{w}(k)) & \cdots & \tilde{D}_{N_\mathrm{p}N_\mathrm{p}}(\tilde{w}(k)) \end{pmatrix},$$

where for $i, j = 1, \ldots, N_\mathrm{p}$,

$$\tilde{C}_i(\tilde{w}(k)) = C(k+i-1) \otimes A(k+i-1) \otimes \cdots \otimes A(k),$$

$$\tilde{D}_{ij}(\tilde{w}(k)) = \begin{cases} C(k+i-1) \otimes A(k+i-1) \otimes \cdots \otimes A(k+j) \otimes B(k+j-1), & \text{if } i > j, \\ C(j+i-1) \otimes B(k+j-1), & \text{if } i = j, \\ \varepsilon, & \text{if } i < j. \end{cases}$$

Note that the entries of $\tilde{C}(\tilde{w}(k))$ and $\tilde{D}(\tilde{w}(k))$ are max-affine functions of $\tilde{w}(k)$ [134]. So the components of $\tilde{y}(k)$ are max-affine functions of $\tilde{w}(k)$ and $\tilde{u}(k)$. Since $\tilde{w}(k)$ is a random vector, $\tilde{y}(k)$ is also a random vector.

6.4.2 Objective function

Define an objective function J that reflects the input and output cost functions from event step k to $k + N_\mathrm{p} - 1$:

$$J(k) = J_\mathrm{out}(k) + \lambda J_\mathrm{in}(k),$$

with the scalar $\lambda \geq 0$ the trade-off between J_out and J_in. In MPC, one aims to design an optimal control sequence $u(k), \ldots, u(k + N_\mathrm{p} - 1)$ that minimizes $J(k)$ subject to constraints on the inputs and the outputs. Different choices for J_out and J_in are given in [47]. In this chapter J_out and J_in are chosen as

$$J_\mathrm{out}(k) = \sum_{j=0}^{N_\mathrm{p}-1} \sum_{i=1}^{n_y} \eta_i(k+j),$$

$$J_\mathrm{in}(k) = -\sum_{j=0}^{N_\mathrm{p}-1} \sum_{l=1}^{n_u} u_l(k+j),$$

where

$$\eta_i(k) = \max(y_i(k) - r_i(k), 0),$$

reflects the delay between the completion time y and the due-date signal r. The selected J_in corresponds to the just-in-time rule. Note that the results in this chapter can be easily extended to other cases such as 1-norm and ∞-norm objective functions used in [47].

Note that J_{out} is random. To obtain a deterministic objective function, the expected value of $J(k)$ is used as the objective function [134]. Moreover, J is actually the maximum of some correlated random variables and it is difficult to get an analytic expression for the distribution of J. So the expected value $\mathbb{E}[J(k)]$ cannot be computed analytically. In this chapter, $\mathbb{E}[J(k)]$ will be computed or approximated by different methods, namely, MC simulation and MC integration. We will combine each method for $\mathbb{E}[J(k)]$ with the methods for chance constraints developed in the next section and compare the efficiency and performance of every combination for a production system example (see Section 6.6).

6.4.3 Chance constraints

Consider the following linear constraints:

$$G\tilde{u}(k) + H\tilde{y}(k) \leq h(k), \tag{6.6}$$

where $G \in \mathbb{R}^{c\times N_\text{p} n_u}$ and $H \in \mathbb{R}^{c\times N_\text{p} n_y}$ are constant matrices and $h(k) \in \mathbb{R}^c$ is a vector depending on the known information at event step k, i.e., the state and input at previous event step and the due-date sequence vector $\tilde{r}(k) = \begin{bmatrix} r^T(k) & \cdots & r^T(k+N_\text{p}-1)\end{bmatrix}^T$.

Note that (6.6) is random due to the uncertainties $\tilde{w}(k)$. To reformulate the random constraints (6.6), we require that (6.6) is satisfied for sufficiently many realizations of $\tilde{w}(k)$, namely,

$$\Pr\left\{G\tilde{u}(k) + H\tilde{y}(k) \leq h(k)\right\} \geq 1-\epsilon, \tag{6.7}$$

where $\epsilon \in (0,1)$ is the probability of possible violation of (6.6). In other words, we require that (6.6) is satisfied at least with a probability $1-\epsilon$. The probabilistic constraint (6.7) is usually called chance constraint.

6.4.4 Problem formulation

Now we combine the material of previous subsections. At step k, the chance-constrained MPC problem for SMPL systems is then defined as follows:

$$\min_{\tilde{u}(k)} \mathbb{E}[J(k)] \tag{6.8}$$

subject to

$$(6.4)-(6.5), \tag{6.9}$$
$$\Pr\left\{G\tilde{u}(k) + H\tilde{y}(k) \leq h(k)\right\} \geq 1-\epsilon, \tag{6.10}$$
$$u(k+j) \geq u(k+j-1),\ j = 0,\ldots,N_\text{p}-1. \tag{6.11}$$

The constraint (6.11) is added since the $u(k),\ldots,u(k+N_\text{p}-1)$ correspond to consecutive event occurrence times.

In general, problem (6.8)-(6.11) is a nonlinear nonconvex optimization problem. For decreasing the computational burden, we aim to transform the problem into reduced forms. In this chater, $\mathbb{E}[J(k)]$ will be approximated by MC simulation [118] and MC integration [46] respectively. Moreover, MC simulation will also be used to deal with the chance constraint and compared with the two approaches developed in the next section.

Remark 6.9 The SMPL-MPC problem was first defined in [134] where the linear constraint (6.6) was reformulated as

$$G\tilde{u}(k) + H\mathbb{E}[\tilde{y}(k)] \leq h(k),$$

instead of the chance constraint (6.7). This means that (6.6) might actually be violated but the extent of violation is uncertain. Therefore, we are motivated to consider using (6.7). □

Remark 6.10 If G and H are block diagonal matrices, then the chance constraint (6.7) can be equivalently written as

$$\Pr\left\{G_j u(k+j-1) + H_j y(k+j-1) \leq h_j(k),\ j = 1,\dots,N_\mathrm{p}\right\} \geq 1-\epsilon, \tag{6.12}$$

where $G_j \in \mathbb{R}^{c_j \times n_u}$, $H_j \in \mathbb{R}^{c_j \times n_y}$, $h_j(k) \in \mathbb{R}^{c_j}$, and

$$G = \begin{bmatrix} G_1 & & \\ & \ddots & \\ & & G_{N_\mathrm{p}} \end{bmatrix},\ H = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_{N_\mathrm{p}} \end{bmatrix},\ h(k) = \begin{bmatrix} h_1(k) \\ \vdots \\ h_{N_\mathrm{p}}(k) \end{bmatrix}.$$

Note that (6.12) involves joint chance constraints.

Alternatively, one can consider individual chance constraints:

$$\Pr\left\{G_{j,i} u(k+j-1) + H_{j,i} y(k+j-1) \leq h_{j,i}(k)\right\} \geq 1-\epsilon,\ i = 1,\dots,c_j,\ j = 1,\dots,N_\mathrm{p}, \tag{6.13}$$

where $G_{j,i}, H_{j,i}, h_{j,i}(k)$ are the i-th rows of $G_j, H_j, h_j(k)$.

The joint chance constraints (6.12) mean that the linear constraints on the inputs and the outputs are satisfied simultaneously from event step k to $k+N_\mathrm{p}-1$ with a probability $1-\epsilon$, while the individual chance constraints (6.13) only limit the probability of violation of every constraint at each event step to ϵ. In this chapter, we consider (6.7). However, the method developed here can also deal with MPC problems for SMPL systems with joint chance constraints (6.12) or individual chance constraints (6.13). □

6.5 Chance-constrained MPC for stochastic MPL systems

In this section, we develop approaches for solving the chance-constrained MPC problem (6.8)-(6.11).

6.5.1 Max-affine form of chance constraints

First we rewrite the chance constraint (6.10) into a max-affine form. We have

$$\begin{aligned}\Pr\left\{G\tilde{u}(k) + H\tilde{y}(k) \leq h(k)\right\} &= \Pr\left\{G\tilde{u}(k) + H\tilde{y}(k) - h(k) \leq 0\right\} \\ &= \Pr\left\{\max_{i=1,\dots,c}\left(G\tilde{u}(k) + H\tilde{y}(k) - h(k)\right)_i \leq 0\right\}.\end{aligned}$$

Note that the vector $G\tilde{u}(k)+H\tilde{y}(k)-h(k)$ only contains affine operations on the components of $\tilde{u}(k)$ and $\tilde{y}(k)$.

Assumption 6.11 *H has nonnegative entries.*

Recall that the components of $\tilde{y}(k)$ are max-affine functions of $\tilde{w}(k)$ and $\tilde{u}(k)$. Assuming that H has nonnegative entries, therefore, each component of $G\tilde{u}(k) + H\tilde{y}(k) - h(k)$ is also a max-affine function of $\tilde{w}(k)$ and $\tilde{u}(k)$. Let $m = \sum_{i=1}^{c} n_i$ where n_i is the number of affine expressions appearing in the maximization for the i-th component of $G\tilde{u}(k) + H\tilde{y}(k) - h(k)$. Hence, we have

$$\Pr\Big\{ \max_{i=1,\dots,c} (G\tilde{u}(k) + H\tilde{y}(k) - h(k))_i \le 0 \Big\} = \Pr\Big\{ \max_{i=1,\dots,m} (z_i(k)) \le 0 \Big\},$$

with

$$z(k) = \Lambda \tilde{w}(k) + \Gamma \tilde{u}(k) + \Xi(k), \tag{6.14}$$

for some appropriately defined matrices and vectors $\Lambda \in \mathbb{R}^{m \times n_{\tilde{w}}}$, $\Gamma \in \mathbb{R}^{m \times N_\mathrm{p} n_u}$, $\Xi(k) \in \mathbb{R}^m$ where $n_{\tilde{w}} = (N_\mathrm{p} + 1) n_w$. Therefore, the chance constraint (6.10) is equivalent to

$$\Pr\Big\{ \max_{i=1,\dots,m} (z_i(k)) \le 0 \Big\} \ge 1 - \epsilon, \tag{6.15}$$

if H has nonnegative elements.

According to (6.14), the components of $z(k)$ are generally not independent and it is difficult to get an analytic expression for the distribution of their maximum. Although the probability in (6.15) can be computed by numerical integration based on MC simulation [46], the computational load is usually heavy. In the following subsections we will introduce two methods to transform (6.15) into a reduced form that can be evaluated efficiently.

6.5.2 Method 1: based on Boole's inequality

In this subsection, we apply Boole's inequality to convert the multivariate constraints (6.15) into several univariate constraints that can be evaluated efficiently.

Theorem 6.12 *If*

$$\sum_{i=1}^{m} \Pr\{z_i(k) > 0\} \le \epsilon, \tag{6.16}$$

then

$$\Pr\Big\{ \max_{i=1,\dots,m} (z_i(k)) \le 0 \Big\} \ge 1 - \epsilon.$$

Proof: We have

$$\Pr\Big\{ \max_{i=1,\dots,m} (z_i(k)) \le 0 \Big\} = 1 - \Pr\Big\{ \max_{i=1,\dots,m} (z_i(k)) > 0 \Big\}.$$

According to the Boole's inequality (6.1), we have

$$\Pr\Big\{ \max_{i=1,\dots,m} (z_i(k)) > 0 \Big\} \le \sum_{i=1}^{m} \Pr\{z_i(k) > 0\}.$$

So if

$$\sum_{i=1}^{m} \Pr\{z_i(k) > 0\} \le \epsilon,$$

then

$$\Pr\Big\{ \max_{i=1,\dots,m} (z_i(k)) \le 0 \Big\} \ge 1 - \epsilon.$$

□

We can see that Theorem 6.12 does not require Assumption 6.8. Based on Theorem 6.12, the optimal control sequence at step k can be calculated by solving the optimization problem (6.8)-(6.9), (6.11) and (6.16), which can be solved more efficiently than the original optimization problem (6.8)-(6.11). However, (6.16) is more conservative than the original chance constraint (6.10).

6.5.3 Method 2: based on Chebyshev's inequality

Theorem 6.12 transforms the chance constrain (6.15) into a reduced form, but the resulting constraints are still nonlinear. Now we introduce an alternative method applying the multidimensional Chebyshev inequality to transform (6.15) into linear constraints on the control inputs and we propose a sufficient condition for applying such method.

According to Assumption 6.8, the components of $\tilde{w}(k) \in \mathbb{R}^{n_{\tilde{w}}}$ are independent and identically distributed random variables. Let $\mu_{\tilde{w}}$ and $\Sigma_{\tilde{w}}$ be the mean vector and covariance matrix of $\tilde{w}(k)$. Define

$$\mu_z(k) = \Lambda\mu_{\tilde{w}} + \Gamma\tilde{u}(k) + \Xi(k), \tag{6.17}$$

$$\Sigma_z = \Lambda\Sigma_{\tilde{w}}\Lambda^T. \tag{6.18}$$

From Theorem 6.6, $\mu_z(k)$ and Σ_z are the mean vector and covariance matrix of $z(k)$.

Theorem 6.13 *Assume that $\Sigma_z \in \mathbb{R}^{m\times m}$ is a positive definite matrix*[2]. *Let $\lambda_{\min}(\Sigma_z^{-1}) > 0$ be the smallest eigenvalue of the matrix Σ_z^{-1}. Let*

$$\bar{\mu}_z(k) = \max_{i=1,\ldots,m} \mu_{z,i}(k).$$

If $\bar{\mu}_z(k) < 0$ and

$$\frac{m}{(\bar{\mu}_z(k))^2\lambda_{\min}(\Sigma_z^{-1})} \le \epsilon,$$

then

$$\Pr\left\{\max_{i=1,\ldots,m}(z_i(k)) \le 0\right\} \ge 1-\epsilon.$$

Proof: For the sake of simplicity, in this proof, we will write z, μ_z instead of $z(k), \mu_z(k)$. Consider

$$\begin{aligned}\max(z_1,\ldots,z_m) &= \max(z_1-\bar{\mu}_z,\ldots,z_m-\bar{\mu}_z)+\bar{\mu}_z\\ &\le \max(z_1-\mu_{z,1},\ldots,z_m-\mu_{z,m})+\bar{\mu}_z\\ &\le \max(|z_1-\mu_{z,1}|,\ldots,|z_m-\mu_{z,m}|)+\bar{\mu}_z\\ &=: \|z-\mu_z\|_\infty+\bar{\mu}_z\\ &\le \|z-\mu_z\|_2+\bar{\mu}_z \ .\end{aligned} \tag{6.19}$$

For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, the smallest eigenvalue $\lambda_{\min}(A)$ has a property that

$$\lambda_{\min}(A)x^Tx \le x^TAx,$$

[2]Note that every covariance matrix is symmetric and positive semi-definite.

for all $x \in \mathbb{R}^n$ [62]. If Σ_z is positive definite, so is Σ_z^{-1}; then we have $\lambda_{\min}(\Sigma_z^{-1}) > 0$ and

$$\lambda_{\min}(\Sigma_z^{-1})\|z-\mu_z\|_2^2 \le (z-\mu_z)^T\Sigma_z^{-1}(z-\mu_z). \tag{6.20}$$

Combining (6.19) and (6.20), we have

$$\begin{aligned}
\Pr\{\max(z_1,\ldots,z_m) \le 0\} &\ge \Pr\{\|z-\mu_z\|_2 \le -\bar{\mu}_z\} \\
&\ge \Pr\{\|z-\mu_z\|_2^2 \le \bar{\mu}_z^2\} \\
&\ge \Pr\{\lambda_{\min}(\Sigma_z^{-1})\|z-\mu_z\|_2^2 \le \lambda_{\min}(\Sigma_z^{-1})\bar{\mu}_z^2\} \\
&\ge \Pr\{(z-\mu_z)^T\Sigma_z^{-1}(z-\mu_z) \le \lambda_{\min}(\Sigma_z^{-1})\bar{\mu}_z^2\}.
\end{aligned} \tag{6.21}$$

From the multidimensional Chebyshev inequality (6.3), we have

$$\Pr\{(z-\mu_z)^T\Sigma_z^{-1}(z-\mu_z) \le \lambda_{\min}(\Sigma_z^{-1})\bar{\mu}_z^2\} \ge 1 - \frac{m}{\bar{\mu}_z^2\lambda_{\min}(\Sigma_z^{-1})} \ . \tag{6.22}$$

If

$$\frac{m}{\bar{\mu}_z^2\lambda_{\min}(\Sigma_z^{-1})} \le \epsilon,$$

therefore, from (6.21) and (6.22), we have

$$\Pr\{\max(z_1,\ldots,z_m) \le 0\} \ge 1-\epsilon.$$

$\square$

Based on Theorem 6.13, the chance constraint (6.15) can be transformed into the following constraint:

$$\mu_z(k) \le -\Big(\frac{m}{\epsilon\lambda_{\min}(\Sigma_z^{-1})}\Big)^{1/2} \ .$$

By substituting (6.17), we obtain

$$\Gamma\tilde{u}(k) \le -\Lambda\mu_{\tilde{w}} - \Xi(k) - \Big(\frac{m}{\epsilon\lambda_{\min}(\Sigma_z^{-1})}\Big)^{1/2} \ . \tag{6.23}$$

Note that this constraint is linear in $\tilde{u}(k)$. Thus the optimal control sequence at step k can be calculated by solving the optimization problem (6.8)-(6.9), (6.11), and (6.23) where (6.11) and (6.23) are both linear constraints.

Remark 6.14 It is important to know that the sufficient condition for this transformation into linear constraints is $\Sigma_z > 0$ (i.e., Σ_z is positive definite). From Assumption 6.8, $\Sigma_{\tilde{w}}$ is positive definite. So from (6.18), Σ_z is positive definite if Λ is a full-row rank matrix, i.e., Λ has rank m. However, in practice, Λ is not always full-row rank and it can even have zero rows. In that case, an alternative procedure is to separate the zero rows from Λ and to divide the remaining part of Λ into several block matrices along the row dimension , i.e., we can assume without loss of generality that Λ has the following format:

$$\Lambda = \begin{bmatrix} \mathbf{0} \\ \Lambda_1 \\ \vdots \\ \Lambda_s \end{bmatrix},$$

such that every block matrix Λ_l is full-row rank. Then we have

$$z(k) = \begin{bmatrix} z^0(k) \\ z^1(k) \\ \vdots \\ z^s(k) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \Lambda_1 \\ \vdots \\ \Lambda_s \end{bmatrix} \tilde{w}(k) + \begin{bmatrix} \Gamma_0 \\ \Gamma_1 \\ \vdots \\ \Gamma_s \end{bmatrix} \tilde{u}(k) + \begin{bmatrix} \Xi_0(k) \\ \Xi_1(k) \\ \vdots \\ \Xi_s(k) \end{bmatrix}.$$

On the one hand, if

$$\Gamma_0 \tilde{u}(k) + \Xi_0(k) > 0,$$

then

$$\Pr\Big\{ \max_{i=1,\dots,m} (z_i(k)) \leq 0 \Big\} = 0.$$

On the other hand, if

$$\Gamma_0 \tilde{u}(k) + \Xi_0(k) \leq 0,$$

then

$$\Pr\Big\{ \max_{i=1,\dots,m} (z_i(k)) \leq 0 \Big\} = \Pr\left\{ \begin{bmatrix} \Lambda_1 \\ \vdots \\ \Lambda_s \end{bmatrix} \tilde{w}(k) + \begin{bmatrix} \Gamma_1 \\ \vdots \\ \Gamma_s \end{bmatrix} \tilde{u}(k) + \begin{bmatrix} \Xi_1(k) \\ \vdots \\ \Xi_s(k) \end{bmatrix} \leq 0 \right\},$$

and the linear constraint (6.23) becomes

$$\begin{aligned} &\Gamma_0 \tilde{u}(k) \leq -\Xi_0(k), \\ &\Gamma_1 \tilde{u}(k) \leq -\Lambda_1 \mu_{\tilde{w}} - \Xi_1(k) - \left(\frac{ms}{\epsilon \lambda_{\min}(\Sigma_{z,1}^{-1})} \right)^{1/2}, \\ &\vdots \\ &\Gamma_s \tilde{u}(k) \leq -\Lambda_s \mu_{\tilde{w}} - \Xi_s(k) - \left(\frac{ms}{\epsilon \lambda_{\min}(\Sigma_{z,s}^{-1})} \right)^{1/2}, \end{aligned} \tag{6.24}$$

with $\Sigma_{z,l} = \Lambda_l \Sigma_{\tilde{w}} \Lambda_l^T$, $l = 1, \dots, s$.

The linear constraints (6.24) guarantee the following constraints:

$$\begin{aligned} &\Pr\{z^0(k) > 0\} = 0, \\ &\Pr\{z^1(k) > 0\} \leq \epsilon / s, \\ &\vdots \\ &\Pr\{z^s(k) > 0\} \leq \epsilon / s. \end{aligned} \tag{6.25}$$

Consequently,

$$\sum_{l=1}^{s} \Pr\{z^l(k) > 0\} \leq \epsilon. \tag{6.26}$$

Similarly to the proof of Theorem 6.12, according to Boole's inequality (6.1), then we have

$$\begin{aligned}\Pr\Big\{\max_{i=1,\dots,m}(z_i(k))\le 0\Big\} &= 1-\Pr\Big\{\max_{i=1,\dots,m}(z_i(k))>0\Big\}\\ &\overset{(6.1)}{\ge} 1-\sum_{l=1}^{s}\Pr\{z^l(k)>0\}\\ &\overset{(6.26)}{\ge} 1-\epsilon.\end{aligned}$$

□

In the chance-constrained MPC problem for SMPL systems, the matrix $\Lambda\in\mathbb{R}^{m\times n_{\tilde w}}$ in (6.14) is usually tall, namely, $m>n_{\tilde w}$ (see the example in Section 6.6). In that case, Λ will not be full-row rank and thus Σ_z is positive semi-definite (i.e., $\Sigma_z\ge 0$).

Theorem 6.15 *Assume that $\Sigma_z\in\mathbb{R}^{m\times m}$ has rank n_Z and $n_Z<m$. Let $\lambda_{\max}(\Sigma_z)$ be the largest eigenvalue of Σ_z. Let*

$$\bar\mu_z(k)=\max_{i=1,\dots,m}\mu_{z,i}(k).$$

If $\bar\mu_z(k)<0$ and

$$\frac{m\lambda_{\max}(\Sigma_z)}{(\bar\mu_z(k))^2}\le\epsilon,$$

then

$$\Pr\Big\{\max_{i=1,\dots,m}(z_i(k))\le 0\Big\}\ge 1-\epsilon.$$

Proof: For the sake of simplicity, in this proof, we will write $\tilde w,z,\mu_z$ instead of $\tilde w(k),z(k),\mu_z(k)$. Compute the singular value decomposition of Σ_z:

$$\Sigma_z=\begin{bmatrix}U_1 & U_2\end{bmatrix}\begin{bmatrix}S_1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}U_1^T\\ U_2^T\end{bmatrix},$$

with $U_1\in\mathbb{R}^{m\times n_Z}$, $U_2\in\mathbb{R}^{m\times(m-n_Z)}$, and $S_1\in\mathbb{R}^{n_Z\times n_Z}>0$. Note that

$$S_1=\mathrm{diag}(s_1,\dots,s_{n_Z}),$$

where s_i are the non-zero eigenvalues of Σ_z. Thus

$$\lambda_{\max}(\Sigma_z)=\max_{i=1,\dots,n_Z}s_i\ .$$

Introduce a dummy random variable $\tilde v\in\mathbb{R}^{m-n_Z}$ where $\tilde v$ is not correlated with $\tilde w$, satisfying $\mathbb{E}[\tilde v]=0$ and $\mathbb{E}[\tilde v\tilde v^T]=\delta I$ where $0\le\delta<\lambda_{\max}(\Sigma_z)$. Define

$$z'=U_2\tilde v.$$

Then we have

$$\mathbb{E}[z']=0,$$

and

$$\mathbb{E}[z'z'^T]=\delta U_2U_2^T\ge 0.$$

Define $z^+ = z + z'$. First note that $\mathbb{E}[z^+] = \mu_{z^+} = \mu_z$. Furthermore,

$$\begin{aligned}
\Sigma_{z^+} &\triangleq \mathbb{E}[(z + z' - \mu_z)(z + z' - \mu_z)^T] \\
&= \mathbb{E}[(z-\mu_z)(z-\mu_z)^T + z'(z-\mu_z)^T + (z-\mu_z)z'^T + z'z'^T] \\
&= \mathbb{E}[(z-\mu_z)(z-\mu_z)^T] + \mathbb{E}[z'z'^T] \\
&= \Sigma_z + \delta U_2 U_2^T \\
&= \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & \delta I \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix},
\end{aligned}$$

where Σ_{z^+} is invertible. Then we derive

$$\Sigma_{z^+}^{-1} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1^{-1} & 0 \\ 0 & \delta^{-1} I \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix},$$

and since $0 \le \delta < \lambda_{\max}(\Sigma_z)$,

$$\begin{aligned}
\lambda_{\min}(\Sigma_{z^+}^{-1}) &= \min\left(\frac{1}{s_1}, \ldots, \frac{1}{s_{n_Z}}\right) \\
&= \frac{1}{\lambda_{\max}(\Sigma_z)} \quad .
\end{aligned}$$

Similarly to the proof of Theorem 6.13, we have

$$\begin{aligned}
\Pr\left\{\max(z_1^+, \ldots, z_m^+) \le 0\right\} &\ge \Pr\left\{(z^+ - \mu_z)^T \Sigma_{z^+}^{-1} (z^+ - \mu_z) \le \lambda_{\min}(\Sigma_{z^+}^{-1}) \bar{\mu}_z^2\right\} \\
&\ge \Pr\left\{(z^+ - \mu_z)^T \Sigma_{z^+}^{-1} (z^+ - \mu_z) \le \frac{\bar{\mu}_z^2}{\lambda_{\max}(\Sigma_z)}\right\}.
\end{aligned}$$

Note that

$$\lim_{\delta \to 0} (z^+ - \mu_z)^T \Sigma_{z^+}^{-1} (z^+ - \mu_z) = (z - \mu_z)^T \Sigma_{z^+}^{-1} (z - \mu_z),$$

and so for $\delta \to 0$ we find

$$\Pr\left\{\max(z_1, \ldots, z_m) \le 0\right\} \ge \Pr\left\{(z - \mu_z)^T \Sigma_{z^+}^{-1} (z - \mu_z) \le \frac{\bar{\mu}_z^2}{\lambda_{\max}(\Sigma_z)}\right\}.$$

From the multidimensional Chebyshev inequality (6.3), we have

$$\Pr\left\{(z - \mu_z)^T \Sigma_{z^+}^{-1} (z - \mu_z) \le \frac{\bar{\mu}_z^2}{\lambda_{\max}(\Sigma_z)}\right\} \ge 1 - \frac{m \lambda_{\max}(\Sigma_z)}{\bar{\mu}_z^2} \quad .$$

So if

$$\frac{m \lambda_{\max}(\Sigma_z)}{\bar{\mu}_z^2} \le \epsilon,$$

then, we have

$$\Pr\left\{\max(z_1, \ldots, z_m) \le 0\right\} \ge 1 - \epsilon.$$

□

6.5.4 Discussion

For Method 1, we need to know the respective distributions of $z_1(k),\ldots,z_m(k)$ instead of the distribution of their maximum; and for Method 2, we need to know the mean vector and covariance matrix of $z(k)$. Based on (6.14), $z(k)$ is an affine function of $\tilde{w}(k)$. Therefore, to apply the two methods developed in this chapter, we require $\tilde{w}(k)$ to be random variables the distribution of which is preserved or known under summation and multiplication by a scalar, such as the normal distribution, the Poisson distribution, and the gamma distribution [110].

6.6 Example

In this section, we consider the production system presented in [134] (see Figure 6.1). We assume that the processing time of M_1 is perturbed by a random variable $w(k)$: $d_1(k) = 5 + w(k)$ where $w(k)$ has a normal distribution with expected value 0 and variance 2.

The initial state is $x(0) = \begin{bmatrix} 0 & 10 \end{bmatrix}^T$, $u(0) = 0$, the prediction horizon is chosen as $N_\mathrm{p} = 3$, and the trade-off between the output and input costs is selected as $\lambda = 10^{-5}$. At each event step k, an MPC optimization problem in the form of (6.8)-(6.11) is solved. The experiment is performed for $k = 1,\ldots,50$. We consider the following chance constraint:

$$\Pr\{y(k+j) - r(k+j) \le h, j = 0,\ldots,N_\mathrm{p}-1\} \ge 1-\epsilon,$$

which is equivalent to

$$\Pr\Big\{\max_{i=1,\ldots,19}(z_i(k)) \le 0\Big\} \ge 1-\epsilon,$$

with $z(k) = \Lambda\tilde{w}(k) + \Gamma\tilde{u}(k) + \Xi(k)$ where the detailed expressions of Λ, Γ, and $\Xi(k)$ are as follows

$$\Lambda = \begin{bmatrix} 1&1&0&0\\0&0&0&0\\0&1&0&0\\1&1&1&0\\0&1&1&0\\1&1&0&0\\0&0&0&0\\0&1&0&0\\0&0&1&0\\1&1&1&1\\0&1&1&1\\0&0&1&1\\1&1&1&0\\0&1&1&0\\1&1&0&0\\0&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0&0&0\\0&0&0\\1&0&0\\0&0&0\\1&0&0\\0&0&0\\0&0&0\\1&0&0\\0&1&0\\0&0&0\\1&0&0\\0&1&0\\0&0&0\\1&0&0\\0&0&0\\0&0&0\\1&0&0\\0&1&0\\0&0&1 \end{bmatrix}, \quad \Xi(k) = \begin{bmatrix} \begin{bmatrix} 12+x_1(k-1)\\ 2+x_2(k-1)\\ 7 \end{bmatrix} - r(k) - 10 \\ \begin{bmatrix} 17+x_1(k-1)\\ 12\\ 13+x_1(k-1)\\ 3+x_2(k-1)\\ 8\\ 7 \end{bmatrix} - r(k+1) - 10 \\ \begin{bmatrix} 22+x_1(k-1)\\ 17\\ 12\\ 18+x_1(k-1)\\ 13\\ 14+x_1(k-1)\\ 4+x_2(k-1)\\ 9\\ 8\\ 7 \end{bmatrix} - r(k+2) - 10 \end{bmatrix}.$$

We consider two different cases: (i) $r(k) = 10+30\cdot k$, $\epsilon = 0.5$, $h = 20$; (ii) $r(k) = 10+75\cdot k$, $\epsilon = 0.2$, $h = 50$. The Boole method (Method 1) and the Chebyshev method (Method 2) developed in Section 6.5 are applied to deal with the chance constraint and compared with two other methods: an MC simulation method and the nominal MPC method. For each case, we solve the chance-constrained MPC problem (6.8)-(6.11) in closed loop for $k = 1,\dots,50$ and run the experiment 10 times, each time with a different realization of w. For each round, the same realization is used for all methods. Table 6.1 lists the mean computation time for the entire simulation over the 10 realizations and the mean closed-loop costs over the 10 realizations. The closed-loop costs are computed as $J_{\mathrm{clp}} = \sum_{k=1}^{50}\big(\max(y(k)-r(k),0)-\lambda u(k)\big)$. Figure 6.2 and 6.3 show the mean tracking error over 10 realizations for all methods.

The nominal MPC method consists in computing the optimal control sequence by using the deterministic MPL system as the prediction model and considering deterministic linear constraints. The MC simulation method consists in approximating $E[J(k)]$ and the chance constraint by using random samples. When using the Boole method or the Chebyshev method to deal with the chance constraint, we consider two different ways to compute the value of $E[J(k)]$, namely, MC integration and MC simulation.

From Table 6.1, we can see that for both cases, although the nominal MPC method is faster than the other methods, it yields higher closed-loop costs. This is because the outputs resulting from nominal MPC violates the due dates at many event steps (see Figure 6.2 the purple solid line with circle markers).

The MC simulation method generally achieves the lowest closed-loop costs, but it takes a longer computation time, e.g., for case (i), it takes 225 seconds with $5\cdot 10^3$ random samples resulting in $J_{\mathrm{clp}} = -0.3817$. The outputs resulting from MC simulation are always below the due dates (see Figure 6.2 the black solid line with diamond markers).

When using the Boole method or the Chebyshev method to deal with the chance constraint, using MC simulation for computing $E[J(k)]$ is better than using MC integration in terms of computation time. Moreover, given the same number of samples, compared with only using MC simulation, the computation time of the combination of the Boole method and MC simulation decreases by about 40% and the computation time of the combination of the Chebyshev method and MC simulation decreases by about 50%.

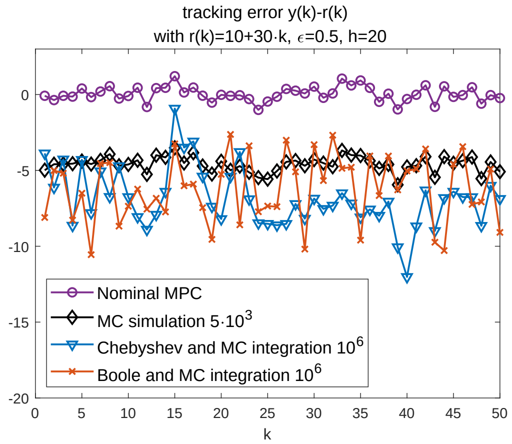
6.7 Conclusions

We have considered the chance-constrained MPC problem for stochastic max-plus linear systems and developed two methods to deal with the chance constraints. Method 1 converts the chance constraint into several univariate constraints by applying Boole's inequality. Method 2 uses Chebyshev's inequality and transforms the chance constraint into linear constraints on the control inputs. The two methods are assessed with a production system and compared with two other methods: MC simulation and nominal MPC. The results show that the two methods are faster than MC simulation while achieving a similar performance and yield a better performance than nominal MPC.

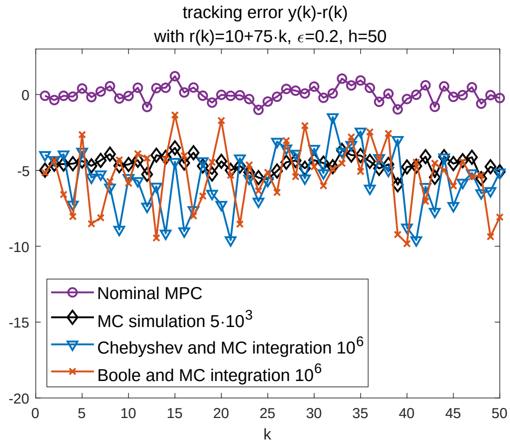
In the future, one possible improvement of Method 2 is to find some optimal way to allocate the probability level ϵ of constraint violation to each of the inequalities in (6.24) (note that in the current chapter ϵ is allocated uniformly). Moreover, more extensive simulations will be implemented.

Table 6.1: The computation time and closed-loop costs J_{clp} using different methods (The number following MC simulation and MC integration indicates the number of random samples used)

Case (i): $r(k) = 10 + 30 \cdot k, \quad \epsilon = 0.5, \quad h = 20$						
Methods					Time [s]	J_{clp}
Constraint		$\mathbb{E}[J(k)]$				
Nominal MPC		Nominal MPC			0.94	29.6440
MC simulation	10^3	MC simulation	10^3		50	-0.3196
	$5 \cdot 10^3$		$5 \cdot 10^3$		225	-0.3817
	10^4		10^4		444	-0.3816
Boole		MC integration	$6 \cdot 10^5$		1271	0.8227
			10^6		2055	-0.3667
			$2 \cdot 10^6$		4194	-0.3807
		MC simulation	10^3		39	-0.3196
			$5 \cdot 10^3$		117	-0.3817
			10^4		216	-0.3816
Chebyshev		MC integration	$6 \cdot 10^5$		1240	-0.1135
			10^6		2020	-0.1190
			$2 \cdot 10^6$		4266	-0.3798
		MC simulation	10^3		26	-0.3195
			$5 \cdot 10^3$		104	-0.3817
			10^4		189	-0.3816
Case (ii): $r(k) = 10 + 75 \cdot k, \quad \epsilon = 0.2, \quad h = 50$						
Methods					Time [s]	J_{clp}
Constraint		$\mathbb{E}[J(k)]$				
Nominal MPC		Nominal MPC			0.77	29.0703
MC simulation	10^3	MC simulation	10^3		51	-0.8933
	$5 \cdot 10^3$		$5 \cdot 10^3$		230	-0.9554
	10^4		10^4		425	-0.9553
Boole		MC integration	$6 \cdot 10^5$		1352	-0.2058
			10^6		2251	-0.4716
			$2 \cdot 10^6$		4625	-0.9538
		MC simulation	10^3		39	-0.8933
			$5 \cdot 10^3$		122	-0.9554
			10^4		212	-0.9553
Chebyshev		MC integration	$6 \cdot 10^5$		1277	-0.5894
			10^6		2117	-0.8450
			$2 \cdot 10^6$		4248	-0.9544
		MC simulation	10^3		27	-0.8933
			$5 \cdot 10^3$		109	-0.9554
			10^4		201	-0.9553

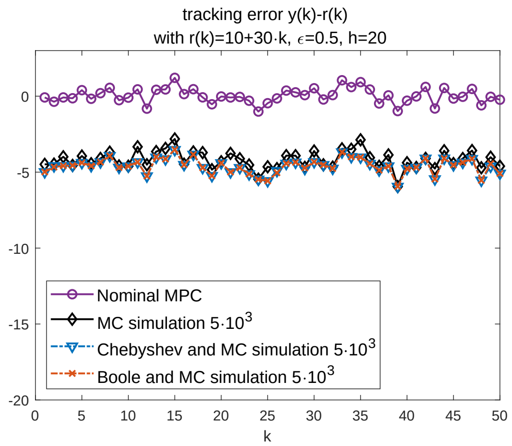


(a) Case (i)

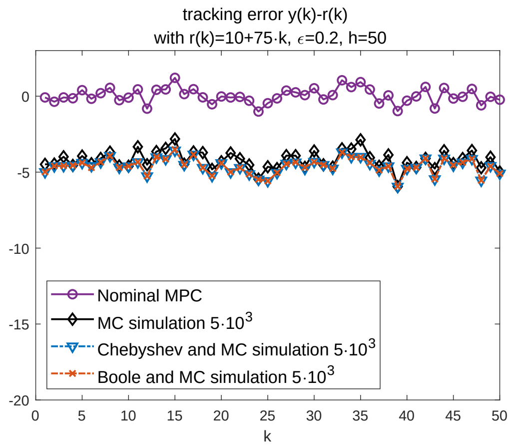


(b) Case (ii)

Figure 6.2: Tracking error resulting from using Boole and MC integration, Chebyshev and MC integration, MC simulation, and nominal MPC with (a) $r(k) = 10 + 30 \cdot k$, $h = 20$ and $\epsilon = 0.5$ (b) $r(k) = 10 + 75 \cdot k$, $h = 50$ and $\epsilon = 0.2$



(a) Case (i)



(b) Case (ii)

Figure 6.3: Tracking error resulting from using Boole and MC simulation, Chebyshev and MC simulation, MC simulation, and nominal MPC with (a) $r(k) = 10 + 30 \cdot k$, $h = 20$ and $\epsilon = 0.5$ (b) $r(k) = 10 + 75 \cdot k$, $h = 50$ and $\epsilon = 0.2$ (The settings of MC simulation and nominal MPC methods are the same as in Figure 6.2)

Chapter 7

Conclusions and recommendations

In this chapter we conclude the thesis by summarizing the main results of the previous chapters. Then we present some recommendations on topics that may be interesting for future research.

7.1 Conclusions of the thesis

In this thesis, we have provided efficient solutions to model predictive control (MPC) problems for max-plus linear (MPL) systems (a specific class of discrete-event systems (DES)), stochastic MPL systems, and continuous piecewise affine (PWA) systems (a specific class of hybrid systems). We have extended variants of optimistic optimization algorithm and optimistic planning algorithms to the control design problem for MPL systems and continuous PWA systems. Moreover, we have developed efficient approaches to solve the chance-constrained MPC problem for stochastic MPL systems. The main results presented in this thesis are summarized as follows:

- **MPC and optimal control for MPL systems**

  We have extended the deterministic optimistic optimization (DOO) algorithm to MPC for MPL systems with continuous control variables and bound constraints on the control variables. First the expressions of the objective function given in [47, 133] in MPC for MPL systems have been generalized. Then analytic expressions for the semi-metric required by DOO have been derived for each type of objective function. Based on the theoretical and numerical analysis, we found that the complexity of the proposed approach increases exponentially in the control horizon instead of in the prediction horizon. This is in contrast to the worst-case complexity of the mixed-integer linear programming (MILP) method for MPC problems, which is exponential in the prediction horizon. The examples have shown that the proposed approach based on DOO is more efficient than MILP when the prediction horizon is large and the control horizon is small.

  The infinite-horizon optimal control problem for MPL systems with discrete control variables has been solved by using the optimistic planning for deterministic systems (OPD) algorithm. The considered infinite-horizon objective function is a discounted sum of the tracking error between the output signal and a due date signal. Given a limited number of iterations, OPD returns at each step a control sequence resulting in a near-optimal value of the objective function. A bound on the difference between the optimal value of the objective function and the near-optimal value is provided. The results of a numerical example have shown that for the given MPL system the proposed approach yields a better tracking than a finite-horizon approach in which a subsequence of the returned control sequence is applied at every control step.

- **MPC for continuous PWA systems**

  We have extended DOO to MPC for discrete-time continuous PWA systems and MMPS systems, which in general leads to a nonlinear, nonconvex optimization problem. In particular, a 1-norm or ∞-norm objective function is considered subject to linear constraints on the states and the inputs. The feasible set is transformed into a hyperbox by considering the linear constraints as soft constraints and adding a penalty function to the objective function. We have developed a dedicated semi-metric and other parameters required by DOO for the proposed problem. A bound on the suboptimality of the returned solution with respect to a global optimum has been derived as a function of the number of iterations in the algorithm. A case study on adaptive cruise control has been implemented to illustrate the performance of the proposed approach.

- **Global optimization of PWA functions over a polytope**

  In [68], the common assumptions of optimistic optimization algorithms have been reformulated into a single assumption. Furthermore, a new definition of the measure of near-optimality analysis has been given. We have adapted this new setting for DOO and considered the global optimization of continuous nonconvex PWA functions over a polytope with the adapted DOO algorithm. The polytopic feasible set may be irregular with arbitrary shape for which the standard partitioning cannot be used. We have presented a partitioning approach based on Delaunay triangulation and edgewise subdivision. Based on this partitioning approach, analytic expressions for the core parameters of the adapted DOO algorithm have been derived. Numerical results have shown that the resulting DOO approach is more efficient than MILP when the considered PWA function has a large number of polyhedral subregions.

- **Chance-constrained MPC for stochastic MPL systems**

  MPC for stochastic MPL systems has been considered where linear constraints on the input and the outputs are written as chance constraints. We have developed two approaches to solve the resulting chance-constrained MPC optimization problem. Based on Boole's inequality, method 1 converts the chance constraint into several univariate constraints. Based on Chebyshev's inequality, method 2 transforms the chance constraint into linear constraints on the control inputs. The two approaches have been compared with the Monte Carlo simulation method and the nominal MPC method for a benchmark production system. The results have shown that although the nomical MPC method is faster than the other methods, it yields higher closed-loop costs. The Monte Carlo simulation method generally achieves the lowest closed-loop costs, but it requires a longer computation time. While achieving a similar performance, method 1 is 40% faster than the Monte Carlo simulation method and method 2 is 50% faster. Hence, method 2 is recommended if its required conditions are satisfied.

7.2 Recommendations for future work

Some suggestions and recommendations for future research are listed below.

- In Chapter 3 we have concluded that the complexity of DOO increases exponentially in the control horizon N_c instead of the prediction horizon N_p. This is in contrast to the worst-case complexity of the MILP method, which is exponential in the prediction horizon. However, if we bisect each dimension of the hyperbox feasible set, the number of branches of the tree established in DOO equals to $2^{N_c n_u}$. This implies that the complexity of DOO will become unacceptable with n_u increasing. One possible solution to solve this bottleneck is to use an input parameterization of the form $u(k) = f(\theta, x(k))$ where θ is a free parameter vector with a lower dimension than $u(k)$.

- The class of optimistic algorithms are applications of the so-called *optimism in the face of uncertainty principle* (i.e., the most promising area of the feasible set is searched first) to large-scale optimization problems [101]. For deterministic function optimization, the uncertainty in the optimistic principle comes from the fact that the feasible solution space of the objective function may be infinite while we are given a finite computational budget only. Moreover, the optimistic algorithms are also adept in dealing with stochastic situations where the uncertainty comes from the noisy estimate of the objective function evaluation. We have exploited the performance of deterministic variants of optimistic optimization algorithms and optimistic planning algorithms for MPL-MPC problems and PWA-MPC problems. The exploitation of stochastic variants of optimistic algorithms of MPC for MPL systems and PWA systems with disturbances and uncertainties are interesting topics for future work.

- MPL systems can model a subclass of DES with synchronization and no choice. This corresponds to the situation that a user or a product is assigned a fixed route when passing through the system and leads to a reduction of flexibility. The emergence of switching MPL systems [135] overcomes this shortcoming. Switching MPL systems are a class of DES that can switch between different modes of operation. In each mode the system is described by an MPL system with different system matrices for each mode. The switching may depend on the inputs and the states, or it may be a stochastic process. Switching MPL systems have been applied to gait generation for multilegged robots [91] and modeling railway networks [85]. The MPC optimization problem of switching MPL systems contains both continuous and discrete optimization variables. To solve this problem, it would be interesting to consider a mix of optimistic optimization algorithms and optimistic planning algorithms.

- Using MPC, an infinite-horizon optimal control problem is solved by repeatedly solving a finite-horizon optimal control problem in a receding horizon fashion. On the other hand, optimistic planning algorithms consider an infinite-horizon discounted objective function with discrete actions. Recently optimistic planning algorithms have been extended to be able to deal with continuous actions [21, 24]. It is interesting to compare the performance of considering an finite-horizon objective function and an infinite-horizon discounted objective function. A good start is to perform a simulation-based comparison; next theoretical analysis may be carried out to verify the simulation results.

- In Chapter 3, we have considered optimistic planning for MPL systems which is a subclass of discrete-event systems. However, OPD has a more general scope of application and works for optimal control problems of general discrete-time nonlinear systems. It would be interesting to develop optimistic planning for PWA systems.

- Due to the computational complexity of the optimization problem, it is difficult to apply MPC to large-scale systems in practice. Hence, we need to consider a more structural control design method, such as distributed MPC or multi-level MPC for large-scale DES and hybrid systems. Using distributed MPC, the considered large-scale DES or hybrid system is assumed to consist of a collection of interconnected subsystems and the control decisions are made by several control agents where each control agent manages one subsystem. Compared with the MPC optimization problem of the overall system, each agent then solves an optimization problem with a smaller size. The approaches proposed in this thesis such as optimistic optimization algorithms, optimistic planning algorithms, and the MILP method, can be employed by each agent to solve its problem. Considering multi-level MPC, at lower levels, control agents usually deal with local, fast dynamics and use more detailed models to optimize objectives with short horizons. The optimization problems at lower levels can be solved by existing mathematical programming methods, while at higher levels, control agents typically deal with large scales, slow dynamics and use less detailed models to optimize objectives with long horizons. The approaches developed in this thesis can be adapted for the optimization problems at higher levels. In addition, a possible direction to build a multi-level structure is based on design structure matrix (DSM) [16].

- In order to further improve the efficiency of current MPC approaches for MPL systems and PWA systems, one can consider more deeply the specific properties (such as nonexpansivity property of MPL systems) of the studied systems to make the optimization algorithms more efficient.

- Optimistic optimization algorithms and optimistic planning algorithms are both based on a hierarchical partitioning of the feasible space. The efficiency of the partitioning approach has an important influence on the performance of the optimistic algorithms. Looking for more effective partitioning approaches (such as adaptive mesh refinement) offers the possibility of improving the current optimistic algorithms.

- In this thesis we only consider numerical examples and simple applications for case studies. The performance of the implementation of the developed approaches in practical environments needs to be investigated.

Appendix A

Norms

This appendix is based on [62]. For any $x \in \mathbb{R}^n$, the p-norm of x is defined as:

$$\|x\|_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}, \qquad p \geq 1.$$

The 1-, 2-, and ∞-norms are the most important:

$$\begin{aligned}
\|x\|_1 &= |x_1| + \cdots + |x_n|, \\
\|x\|_2 &= (|x_1|^2 + \cdots + |x_n|^2)^{1/2}, \\
\|x\|_\infty &= \max(|x_1|, \ldots, |x_n|),
\end{aligned}$$

and it holds that

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \quad .$$

A classic result concerning p-norms is the *Hölder inequality*:

$$|x^T y| \leq \|x\|_p \| \|y\|_q \qquad \frac{1}{p} + \frac{1}{q} = 1.$$

Below are two special cases of this inequality:

$$\begin{aligned}
|x^T y| &\leq \|x\|_2 \|y\|_2, \\
|x^T y| &\leq \|x\|_1 \|y\|_\infty.
\end{aligned}$$

Bibliography

[1] B. Addad, S. Amari, and J.-J. Lesage. Networked conflicting timed event graphs representation in (Max, plus) algebra. *Discrete Event Dynamic Systems-Theory and Applications*, 22(4):429–449, December 2012.

[2] S. Amari, I. Demongodin, J.J. Loiseau, and C. Martinez. Max-plus control design for temporal constraints meeting in timed event graphs. *IEEE Transactions on Automatic Control*, 57(2):462–467, 2012.

[3] P.J. Antsaklis. A brief introduction to the theory and applications of hybrid systems. *Proceedings of the IEEE, Special Issue on Hybrid Systems: Theory and Applications*, 88 (7):879–887, July 2000.

[4] A. Atamtürk and M.W.P. Savelsbergh. Integer-programming software systems. *Annals of Operation Research*, 140(1):67–124, 2005.

[5] A. M. Atto, C. Martinez, and S. Amari. Control of discrete event systems with respect to strict duration: Supervision of an industrial manufacturing plant. *Computers & Industrial Engineering*, 61(4):1149–1159, 2011.

[6] S. Azuma, J. Imura, and T. Sugie. Lebesgue piecewise affine approximation of nonlinear systems. *Nonlinear Analysis-Hybrid Systems*, 4(1):92–102, 2010.

[7] F.L. Baccelli, G. Cohen, G.J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems.* John Wiley & Sons, New York, 1992.

[8] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

[9] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in Identification and Control*, volume 245, pages 207–226. Springer, London, 1999.

[10] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45(10): 1864–1876, 2000.

[11] L. Ben-Naoum, R. Boel, L. Bongaerts, B. De Schutter, Y. Peng, P. Valckenaers, J. Vandewalle, and V. Wertz. Methodologies for discrete event dynamic systems: A survey. *Journal A*, 36(4):3–14, 1995.

[12] M. Benmiloud, A. Benalia, M. Djemai, and M. Defoort. Hybrid control design for limit cycle stabilisation of planar switched systems. *International Journal of Control*, 91(7): 1720–1729, 2018.

[13] J. Bey. Simplicial grid refinement: On freudenthal's algorithm and the optimal number of congruence classes. *Numerische Mathematik*, 85(1):1–29, 2000.

[14] J.-L. Boimond and J.-L. Ferrier. Internal model control and max-algebra: Conroller design. *IEEE Transactions on Automatic Control*, 41(3):457–461, 1996.

[15] F. Borrelli. *Constrained optimal control of linear and hybrid systems*. Springer, Berlin, 2003.

[16] T.R. Browning. Design structure matrix extensions and innovations: A survey and new opportunities. *IEEE Transactions on Engineering Management*, 63(1):27–52, February 2016.

[17] S. Bubeck and R. Munos. Open loop optimistic planning. In *Proceedings 23rd Annual Conference on Learning Theory*, pages 27–29, Haifa, Israel, 2010.

[18] L. Buşoniu and I.-C. Morărescu. Consensus for black-box nonlinear agents using optimistic optimization. *Automatica*, 50(4):1201–1208, 2014.

[19] L. Buşoniu, R. Munos, B. De Schutter, and R. Babuška. Optimistic planning for sparsely stochastic systems. In *Proceedings 2011 IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 48–55, Paris, France, 2011.

[20] L. Buşoniu, R. Munos, and R. Babuska. A review of optimistic planning in Markov decision processes. In *Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control*, pages 494–516. Wiley, 2012.

[21] L. Buşoniu, A. Daniels, R. Munos, and R. Babuska. Optimistic planning for continuous-action deterministic systems. In *Proceedings of the 2013 Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 15–19, Singapore, 2013.

[22] L. Buşoniu, R. Postoyan, and J. Daafouz. Near-optimal strategies for nonlinear networked control systems using optimistic planning. In *Proceedings of the 2013 American Control Conference (ACC-13)*, pages 3020–3025, Washington DC, USA, June 2013.

[23] L. Buşoniu, M.-C. Bragagnolo, J. Daafouz, and C. Morarescu. Planning methods for the optimal control and performance certification of general nonlinear switched systems. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pages 3604–3609, Osaka, Japan, 2015.

[24] L. Buşoniu, E. Páll, and R. Munos. Discounted near-optimal control of general continuous-action nonlinear systems using optimistic planning. In *Proceedings of the 2016 American Control Conference*, Boston, USA, July 2016.

[25] G.C. Calafiore and L. Fagiano. Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, 58(1):219–224, 2013.

[26] E.F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry*. Springer-Verlag, London, 1995.

[27] R.M.F. Candido, L. Hardouin, M. Lhommeau, and R.S. Mendes. Conditional reachability of uncertain max plus linear systems. *Automatica*, 94:426–435, August 2018.

[28] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, New York, second edition, 2008.

[29] M.C. Cavichia and M.N. Arenales. Piecewise linear programming via interior points. *Computers & Operations Research*, 27(13):1303–1324, 2000.

[30] A. Charnes and W. W. Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.

[31] S.-W. Cheng, T.K. Dey, and J.R. Shewchuk. *Delaunay Mesh Generation*. CRC Press, 2012.

[32] L.O. Chua and A.C. Deng. Canonical piecewise-linear representation. *IEEE Transactions on Circuits and Systems*, 35(1):101–111, 1988.

[33] L.O. Chua and R.L.P. Ying. Canonical piecewise-linear analysis. *IEEE Transactions on Circuits and Systems*, 30(3):125–140, 1983.

[34] K.L. Chung. *A Course in Probability Theory*. Academic Press, San Diego, USA, third edition, 2001.

[35] G. Cohen, D. Dubois, J.-P. Quadrat, and M. Viot. A linear-system-theoretic view of discrete-event processes. In *Proceedings of the 22nd IEEE Conference on Decision and Control*, pages 1039–1044, San Antonio, Texas, USA, 1983.

[36] G. Cohen, D. Dubois, J.-P. Quadrat, and M. Viot. A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Transactions on Automatic Control*, 30(3):210–220, 1985.

[37] G. Cohen, P. Moller, J.-P. Quadrat, and M. Viot. Algebraic tools for the performance evaluation of discrete event systems. *Proceedings of the IEEE*, 77(1):39–85, 1989.

[38] D. Corona, I. Necoara, B. De Schutter, and T. van den Boom. Robust hybrid MPC applied to the design of an adaptive cruise controller for a road vehicle. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1721–1726, San Diego, California, December 2006.

[39] B. Cottenceau, L. Hardouin, J.L. Boimond, and J.L. Ferrier. Model reference control for timed event graphs in dioids. *Automatica*, 37(9):1451–1458, 2001.

[40] K.L. Croxton, B. Gendron, and T.L. Magnanti. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science*, 49(9):1268–1273, 2003.

[41] K.L. Croxton, B. Gendron, and T.L. Magnanti. Models and methods for merge-in-transit operations. *Transportation Science*, 37(1):1–22, 2003.

[42] R.A. Cuninghame-Green. Process synchronisation in a steelworks–a problem of feasibility. In *Proceedings of the 2nd International Conference on Operational Research*, pages 323–328, Aix-en-Provence, France, 1960.

[43] R.A. Cuninghame-Green. Describing industrial processes with interference and approximating their steady-state behaviour. *Journal of the Operational Research Society*, 13(1):95–100, 1962.

[44] R.A. Cuninghame-Green. Projections in minimax algebra. *Mathematical Programming*, 10:111–123, 1976.

[45] R.A. Cuninghame-Green. *Minimax Algebra*. Springer-Verlag, Berlin, Germany, 1979.

[46] P.J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Dover Publications, Mineola, New York, second edition, 2007.

[47] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056, July 2001.

[48] B. De Schutter and T. van den Boom. MPC for continuous piecewise-affine systems. *Systems & Control Letters*, 52(3–4):179–192, July 2004.

[49] P. Declerck and M. K. Didi Alaoui. Optimal control synthesis of timed event graphs with interval model specifications. *IEEE Transactions on Automatic Control*, 55(2):518–523, 2010.

[50] M. di Bernardo, U. Montanaro, R. Ortega, and S. Santini. Extended hybrid model reference adaptive control of piecewise affine systems. *Nonlinear Analysis-Hybrid Systems*, 21:11–21, August 2016.

[51] H. Edelsbrunner and D.R. Grayson. Edgewise subdivision of a simplex. *Discrete & Computational Geometry*, 24(4):707–719, 2000.

[52] S.S. Farahani, T. van den Boom, H. van der Weide, and B. De Schutter. An approximation method for computing the expected value of max-affine expressions. *European Journal of Control*, 27:17–27, 2016.

[53] M. Farina and R. Scattolini. Model predictive control of linear systems with multiplicative unbounded uncertainty and chance constraints. *Automatica*, 70:258–265, 2016.

[54] M. Farina, L. Giulioni, and R. Scattolini. Stochastic linear model predictive control with chance constraints-a review. *Journal of Process Control*, 44:53–67, 2016.

[55] R. Fourer. A simplex algorithm for piecewise-linear programming I – Derivation and proof. *Mathematical Programming*, 33(2):204–233, 1985.

[56] T. Fujisawa and E.S. Kuh. Piecewise-linear theory of nonlinear networks. *SIAM Journal on Applied Mathematics*, 22(2):307–328, 1972.

[57] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335–348, 1989.

[58] M.J. Gazarik and E.W. Kamen. Reachability and observability of linear systems over max-plus. *Kybernetika*, 35(1):2–12, 1999.

[59] B. Giffler. Mathematical solution of production planning and scheduling problems. Technical report, IBM Advanced Systems Deelopment Division, 1961.

[60] B. Giffler. Scheduling general production systems using schedule algebra. *Naval Research Logistics Quarterly*, 10(3):237–255, 1963.

[61] B. Giffler. Schedule algebra: A progress report. *Naval Research Logistics Quarterly*, 15 (2):255–280, 1968.

[62] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, second edition, 1990.

[63] V.M. Goncalves, C.A. Maia, and L. Hardouin. On max-plus linear dynamical system theory: The regulation problem. *Automatica*, 75:202–209, January 2017.

[64] E.N. Gonçalves, R.M. Palhares, R.H.C. Takahashi, and R.C. Mesquita. New approach to robust $\mathscr{D}$-stability analysis of linear time-invariant systems with polytope-bounded uncertainty. *IEEE Transactions on Automatic Control*, 51(10):1709–1714, 2006.

[65] M. Gondran and M. Minoux. Eigenvalues and eigenvectors in semimodules and their interpretation in graph theory. In *Proceedings of the 9th International Mathematical Programming Symposium*, pages 333–348, Budapest, Hungary, 1976.

[66] M. Gondran and M. Minoux. Linear algebra in dioids: A survey of recent results. *North-Holland Mathematics Studies*, 95, 1984.

[67] V.V. Gorokhovik and O.I. Zorko. Piecewise affine functions and polyhedral sets. *Optimization*, 31(3):209–221, 1994.

[68] J.-B. Grill, M. Valko, and R. Munos. Black-box optimization of noisy functions with unknown smoothness. In *Advances in Neural Information Processing Systems 28*, pages 667–675. 2015.

[69] J. Haddad, B. De Schutter, D. Mahalel, I. Ioslovich, and P.-O. Gutman. Optimal steady-state control for isolated traffic intersections. *IEEE Transactions on Automatic Control*, 55(11):2612–2617, 2010.

[70] L. Hardouin, C.A. Maia, B. Cottenceau, and M. Lhommeau. Observer design for (max,plus) linear systems. *IEEE Transactions on Automatic Control*, 55(2):538–543, 2010.

[71] L. Hardouin, Y. Shang, C.A. Maia, and B. Cottenceau. Observer-based controllers for max-plus linear systems. *IEEE Transactions on Automatic Control*, 62(5):2153–2165, May 2017.

[72] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001.

[73] W.P.M.H. Heemels, B. De Schutter, J. Lunze, and M. Lazar. Stability analysis and controller synthesis for hybrid dynamical systems. *Philosophical Transactions of the Royal Society A*, 368(1930):4937–4960, 2010.

[74] B. Heidergott, G.J. Olsder, and J.W. van der Woude. *Max Plus at Work-Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications.* Princeton University Press, Princeton, New Jersey, 2006.

[75] Y.C. Ho. *Proceedings of the IEEE (Special issue on Dynamics of Discrete Event Systems)*, 77(1), 1989.

[76] Y.C. Ho and X.R. Cao. *Perturbation Analysis of Discrete Event Dynamic Systems.* Kluwer Academic Publishers, Boston, 1991.

[77] L. Houssin, S. Lahaye, and J.-L. Boimond. Control of (*max*,+)-linear systems minimizing delays. *Discrete Event Dynamic Systems*, 23(3):261–276, 2013.

[78] J.F. Hren and R. Munos. Optimistic planning of deterministic systems. In *Recent Advances in Reinforcement Learning, Proceedings of the 8th European Workshop, EWRL 2008, Villeneuve d'Ascq, France, June 30 - July 3, 2008, Revised and Selected Papers*, volume 5323 of *Lecture Notes in Computer Science*, pages 151–164. Springer, Berlin, 2008.

[79] B. Hruz and M.C. Zhou. *Modeling and Control of Discrete-event Dynamic Systems.* Springer, London, 1st edition, 2007.

[80] R. Iervolino, D. Tangredi, and F. Vasca. Lyapunov stability for piecewise affine systems via cone-copositivity. *Automatica*, 81:22–29, July 2017.

[81] M. Johansson. *Piecewise Linear Control Systems.* Springer, Berlin, Germany, 2003.

[82] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis.* Pearson Prentice Hall, Upper Saddle River, New Jersey, USA, sixth edition, 2007.

[83] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application*, 79(1):157–181, 1993.

[84] R.D. Katz. Max-plus (*A*,*B*)-invariant spaces and control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 52(2):229–241, 2007.

[85] B. Kersbergen, J. Rudan, T.J.J. van den Boom, and B. De Schutter. Towards railway traffic management using switching max-plus-linear systems. *Discrete Event Dynamic Systems-Theory and Applications*, 26(2):183–223, June 2016.

[86] S. Kersting and M. Buss. Direct and indirect model reference adaptive control for multivariable piecewise affine systems. *IEEE Transactions on Automatic Control*, 62 (11):5634–5649, November 2017.

[87] T.A.M. Kevenaar and D.M.W. Leenaerts. A comparison of piecewise-linear model descriptions. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 39(12):996–1004, 1992.

[88] L. Kocsis and Cs. Szepesvári. Bandit based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, pages 282–293, 2006.

[89] J. Komenda, S. Lahaye, J.-L. Boimond, and T. van den Boom. Max-plus algebra in the history of discrete event systems. *Annual Reviews in Control*, 45:240–249, 2018.

[90] D.M.W. Leenaerts and W.M.G. van Bokhoven. *Piecewise Linear Modeling and Analysis*. Kluwer Academic Publishers, Boston, USA, 1998.

[91] G.A.D. Lopes, B. Kersbergen, T.J.J. van den Boom, and B. De Schutter. Modeling and control of legged locomotion via switching max-plus models. *IEEE Transactions on Robotics*, 30(3):652–665, June 2014.

[92] D.G. Luenberger and Y.Y. Ye. *Linear and Nonlinear Programming*. Springer, New York, 2005.

[93] J. Lunze and F. Lamnabhi-Lagarrigue. *Handbook of Hybrid Systems Control: Theory, Tools, Applications*. Cambridge University Press, Cambridge, UK, 2009.

[94] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, England, 2002.

[95] C.A. Maia, C.R. Andrade, and L. Hardouin. On the control of max-plus linear system subject to state restriction. *Automatica*, 47(5):988–992, 2011.

[96] K. Máthé, L. Buşoniu, R. Munos, and B. De Schutter. Optimistic planning with a limited number of action switches for near-optimal nonlinear control. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 3518–3523, Los Angeles, California, USA, December 2014.

[97] D. Q. Mayne, M. M. Seron, and S. V. Rakovic. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[98] E. Menguy, J.-L. Boimond, L. Hardouin, and J.-L. Ferrier. Just-in-time control of timed event graphs: update of reference input, presence of uncontrollable input. *IEEE Transactions on Automatic Control*, 45(9):2155–2159, 2000.

[99] A. Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.

[100] R. Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *25th Annual Conference on Neural Information Processing Systems*, pages 783–791, Granada, Spain, December 2011.

[101] R. Munos. From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends in Machine Learning*, 7(1):1–130, 2014.

[102] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77 (4):541–580, 1989.

[103] Ion Necoara. *Model Predictive Control for Piecewise Affine and Max-Plus-Linear Systems.* PhD thesis, Delft University of Technology, 2006.

[104] F.J. Nourie and F. Guder. A restricted-entry method for a transportation problem with piecewise-linear concave costs. *Computers & Operations Research*, 21(7):723–733, 1994.

[105] G.J. Olsder. On the characteristic equation and minimal realizations for discrete-event dynamic systems. In A. Bensoussan and J.L. Lions, editors, *Analysis and Optimization of Systems*, volume 83 of *Lecture Notes in Control and Information Sciences*, pages 189–201. Springer, Berlin, Heidelberg, 1986.

[106] G.J. Olsder and C. Roos. Cramer and cayley-hamilton in the max algebra. *Linear Algebra and its Applications*, 101:87–108, 1988.

[107] G.J. Olsder, J.A.C. Resing, R.E. de Vries, M.S. Keane, and G. Hooghiemstra. Discrete event systems with stochastic processing times. *IEEE Transactions on Automatic Control*, 35(3):299–302, 1990.

[108] S. Ovchinnikov. Max-min representation of piecewise linear functions. *Beiträge zur Algebra und Geometrie/Contributions to Algebra and Geometry*, 43(1):297–302, 2002.

[109] R. Munos P. Preux and M. Valko. Bandits attack function optimization. pages 2245–2252, Beijing, China, 2014.

[110] A. Papoulis and S.U. Pillai. *Probability, random variables, and stochastic processes.* McGraw-Hill Education, New York, forth edition, 2002.

[111] P. Pejovic and D. Maksimovic. A new algorithm for simulation of power electronic systems using piecewise-linear device models. *IEEE Transactions on Power Electronics*, 10(3):340–348, 1995.

[112] G. Pola and M.D. Di Benedetto. Symbolic models and control of discrete-time piecewise affine systems: An approximate simulation approach. *IEEE Transactions on Automatic Control*, 59(1):175–180, January 2014.

[113] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[114] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design.* Nob Hill, Madison, Wisconsin, 2009.

[115] H. Razavi, K. Merat, H. Salarieh, A. Alasty, and A. Meghdari. Observer based minimum variance control of uncertain piecewise affine systems subject to additive noise. *Nonlinear Analysis-Hybrid Systems*, 19, February 2016.

[116] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, September 1978.

[117] N. Rizk, A. Martel, and A. Ramudhin. A Lagrangean relaxation algorithm for multi-item lot-sizing problems with joint piecewise linear resource costs. *International Journal of Production Economics*, 102(2):344–357, 2006.

[118] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Spinger, 2004.

[119] S. Ross. *A First Course in Probability*. Pearson Prentice Hall, Upper Saddle River, New Jersey, USA, eighth edition, 2010.

[120] V. Rostampour, D. Adzkiya, S. Soudjani, B. De Schutter, and T. Keviczky. Chance constrained model predictive controller synthesis for stochastic max-plus linear systems. In *IEEE International Conference on Systems, Man, and Cybernetics*, Budapest, Hungary, October 2016.

[121] M. Rubagotti, L. Zaccarian, and A. Bemporad. A Lyapunov method for stability analysis of piecewise-affine systems over non-invariant domains. *International Journal of Control*, 89(5):950–959, May 2016.

[122] S. Scholtes. *Introduction to Piecewise Differentiable Equations*. Springer, New York, 2012.

[123] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, England, 1986.

[124] G. Schullerus, V. Krebs, B. De Schutter, and T. van den Boom. Input signal design for identification of max-plus-linear systems. *Automatica*, 42(6):937–943, June 2006.

[125] Y. Shang. Towards geometric control of max-plus linear systems with applications to queueing networks. *International Journal of Systems and Science*, 44(1):34–49, 2013.

[126] Y. Shang, L. Hardouin, M. Lhommeau, and C.A. Maia. An integrated control strategy to solve the disturbance decoupling problem for max-plus linear systems with applications to a high throughput screening system. *Automatica*, 63:338–348, January 2016.

[127] E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, 1981.

[128] S. Sridhar, J. Linderoth, and J. Luedtke. Models and solution techniques for production planning problems with increasing byproducts. *Journal of Global Optimization*, 59(2–3):597–631, 2014.

[129] S.M. Tabatabaeipour, M. Gholami, and T. Bak. Configuration selection for reconfigurable control of piecewise affine systems. *International Journal of Control*, 88(6):1310–1323, June 2015.

[130] K. Tebani, S. Amari, and R. Kara. Closed-loop control of constrained discrete event systems: Application to a networked automation system. *International Journal of Advanced Manufacturing Technology*, 90(5-8):1295–1306, May 2017.

[131] M. Vaezi and A. Izadian. Piecewise affine system identification of a hydraulic wind power transfer system. *IEEE Transactions on Control Systems Technology*, 23(6):2077–2086, November 2015.

[132] M. Valko, A. Carpentier, and R. Munos. Stochastic simultaneous optimistic optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 19–27, Atlanta, Georgia, USA, 2013.

[133] T. van den Boom and B. De Schutter. Properties of MPC for max-plus-linear systems. *European Journal of Control*, 8(5):453–462, 2002.

[134] T. van den Boom and B. De Schutter. Model predictive control for perturbed max-plus-linear systems: A stochastic approach. *International Journal of Control*, 77(3):302–309, February 2004.

[135] T. van den Boom and B. De Schutter. Modeling and control of switching max-plus-linear systems with random and deterministic switching. *Discrete Event Dynamic Systems: Theory and Applications*, 22(3):293–332, September 2012.

[136] T. van den Boom and B. De Schutter. Model predictive control of manufacturing systems with max-plus algebra. In J. Campos, C. Seatzu, and X. Xie, editors, *Formal Methods in Manufacturing*, Industrial Information Technology, chapter 12, pages 343–380. CRC Press, February 2014.

[137] T. van den Boom, B. Heidergott, and B. De Schutter. Complexity reduction in MPC for stochastic max-plus-linear discrete event systems by variability expansion. *Automatica*, 43(6):1058–1063, June 2007.

[138] T.J.J. van den Boom and B. De Schutter. Analytic expressions in stochastic max-plus-linear algebra. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 1608–1613, Los Angeles, California, December 2014.

[139] A.J. van der Schaft and J.M. Schumacher. *An Introduction to Hybrid Dynamical Systems*, volume 251 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2000.

[140] V.M. Veliov and M.I. Krastanov. Controllability of piecewise linear systems. *Systems & Control Letters*, 7(5):335–341, 1986.

[141] J.P. Vielma, S. Ahmed, and G. Nemhauser. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research*, 58(2):303–315, 2010.

[142] M.P. Vitus, Z.Y. Zhou, and C.J. Tomlin. Stochastic control with uncertain parameters via chance constrained control. *IEEE Transactions on Automatic Control*, 61(10):2892–2905, 2016.

[143] C.L. Wang and Y.G. Tao. Interval strong solutions of interval systems of max-plus linear equations. *Linear Algebra and its Applications*, 537:148–159, January 2018.

[144] C.L. Wang, Y.G. Tao, P. Yang, and Z.J. Liu. Dimension reduction and feedback stabilization for max-plus linear systems and applications in VLSI array processors. *IEEE Transactions on Automatic Control*, 62(12):6353–6368, December 2017.

[145] J. Xie and J. Zhao. H-infinity model reference adaptive control for switched systems based on the switched closed-loop reference model. *Nonlinear Analysis-Hybrid Systems*, 27:92–106, February 2018.

[146] J. Xie, D. Yang, and J. Zhao. Multiple model adaptive control for switched linear systems: A two-layer switching strategy. *International Journal of Robust and Nonlinear Control*, 28(6):2276–2297, April 2018.

[147] J. Xu, B. De Schutter, and T.J.J. van den Boom. Model predictive control for max-plus-linear systems via optimistic optimization. In *12th IFAC/IEEE Workshop on Discrete Event Systems*, pages 111–116, Cachan, France, May 2014.

[148] J. Xu, L. Buşoniu, T.J.J. van den Boom, and B. De Schutter. Receding-horizon control for max-plus linear systems with discrete actions using optimistic planning. In *13th International Workshop on Discrete Event Systems*, pages 398–403, Xi'an, China, May–June 2016.

[149] J. Xu, T. van den Boom, and B. De Schutter. Optimistic optimization for model predictive control of max-plus linear systems. *Automatica*, 74:16–22, December 2016.

[150] J. Xu, T.J.J. van den Boom, L. Buşoniu, and B. De Schutter. Model predictive control for continuous piecewise affine systems using optimistic optimization. In *2016 American Control Conference*, pages 4482–4487, Boston, Massachusetts, July 2016.

[151] J. Xu, T. van den Boom, and B. De Schutter. Optimistic optimization for continuous nonconvex piecewise affine functions. *Automatica*, 2018. To appear.

[152] J. Xu, T. van den Boom, and B. De Schutter. Model predictive control for stochastic max-plus linear systems with chance constraints. *IEEE Transactions on Automatic Control*, 64(1):337–342, 2019.

[153] L.X. Zhang, Z.P. Ning, and W.X. Zheng. Observer-based control for piecewise-affine systems with both input and output quantization. *IEEE Transactions on Automatic Control*, 62(11):5858–5865, November 2017.

[154] M.C. Zhou and F. DiCesare. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publishers, Boston, 1991.

Summary

Optimization and Model-Based Control for Max-Plus Linear and Continuous Piecewise Affine Systems

This PhD thesis considers the development of optimization and model-based control techniques for max-plus linear (MPL) and continuous piecewise affine (PWA) systems. The three main topics investigated in this thesis are as follows:

1. Optimistic optimization and planning for model-based control of MPL systems

   Model predictive control (MPC) for MPL systems usually leads to a nonsmooth nonconvex optimization problem with real-valued variables, which may be hard to solve efficiently. An alternative approach is to transform the given problem into a mixed integer linear programming (MILP) problem. However, the computational complexity of current MILP algorithms increases in the worst case exponentially as a function of the prediction horizon. In this thesis, we adapt optimistic optimization for solving the MPC optimization problem for MPL systems. Optimistic optimization is a class of algorithms that can find an approximation of the global optimum for general nonlinear optimization problems. A key advantage of optimistic optimization is that one can characterize the level of near-optimality of the resulting solution with respect to the global optimum, given a finite computational budget. We consider MPC for MPL systems with boundary constraints on the increments of the control inputs. The objective function is a trade-off between the output cost (i.e., weighted tardiness-earliness penalty with respect to a due-date signal) and the input cost (i.e., feed as late as possible). A dedicated semi-metric is developed satisfying the necessary requirements for optimistic optimization. Based on the theoretical analysis, we prove that the complexity of optimistic optimization is exponential in the control horizon instead of the prediction horizon. Hence, using optimistic optimization is computationally more efficient when the control horizon is small and the prediction horizon is large.

   In addition, we address the infinite-horizon optimal control problem for MPL systems where the considered objective function is a sum of discounted stage costs over an infinite horizon. We consider the increments of the control inputs as control variables and the control space is discretized as a finite set. The resulting optimal control problem is equivalently transformed into an online planning problem that involves maximizing a reward function. We adapt an optimistic planning algorithm to solve this problem. Given a finite computational budget, a control sequence is returned and the first control action or a subsequence of the returned control sequence is applied to the system and then a receding-horizon scheme is adopted. The proposed

optimistic planning approach yields a characterization of the near-optimality of the resulting solution. The simulation results show that when a subsequence of the returned control sequence is applied, this approach results in a lower tracking error compared with a fintie-horizon approach.

2. Optimistic optimization for MPC of continuous PWA systems

   We further adapt optimistic optimization for solving the MPC optimization problem for continuous PWA systems. The considered 1-norm and ∞-norm objective functions are continuous PWA functions. The linear constraints on the states and the inputs are treated as soft constraints and replaced by adding a penalty function to the objective function. The proposed optimistic optimization approach is based on recursive partitioning of the resulting hyperbox feasible set. We derive expressions for the core parameters of optimistic optimization and discuss the near-optimality of the resulting solution by applying optimistic optimization. The performance of the proposed approach is illustrated with a case study on adaptive cruise control.

   From the first part of this topic, we can see that the optimization problem of a continuous nonconvex PWA function arises in the context of control of continuous PWA systems. In the literature, it has been shown that this type of optimization problem can be formulated as a MILP problem, the worst-case complexity of which grows exponentially with the number of polyhedral subregions in the domain of the PWA function. In the first part, we have applied optimistic optimization to solve the global optimization problem of a continuous nonconvex PWA function over a hyperbox control space. But the constraints on the states and the inputs were treated as soft constraints. In the second part, we extend optimistic optimization from a hyperbox feasible set to a polytopic feasible set. More specifically, we propose a partitioning framework of the polytopic feasible set satisfying the requirements of optimistic optimization by employing Delaunay triangulation and edgewise subdivision. For this partitioning approach, we derive analytic expressions for the core ingredients that are used for characterizing the near-optimality of the solution obtained by optimistic optimization. When applied to optimize PWA functions, the proposed optimistic optimization approach is computationally more efficient than MILP if the number of polyhedral subregions in the domain is much larger than the number of variables of the PWA function.

3. MPC for stochastic MPL systems with chance constraints

   The topic of the last part of this thesis is MPC for MPL systems with stochastic uncertainties the distribution of which is supposed to be known. We consider linear constraints on the inputs and the outputs. Due to the uncertainties, these linear constraints are formulated as probabilistic or chance constraints, i.e., the constraints are required to be satisfied with a predefined probability level. The proposed probabilistic constraints can be equivalently rewritten into a max-affine form (i.e., the maximum of affine terms) if the linear constraints are monotonically nondecreasing as a function of the outputs. Based on the resulting max-affine form, two methods are developed for solving the chance-constrained MPC problem for stochastic max-plus linear systems. Method 1 uses Boole's inequality to convert the multivariate chance constraints into univariate chance constraints for which the probability can be computed more efficiently. Furthermore, Method 2 employs the multidimensional

Chebyshev inequality and transforms the multivariate chance constraints into constraints that are linear in the inputs. With a production system example, the two proposed methods are compared to the numerical integration method and the nominal MPC method. From the point of view of computation time, both our methods are faster than numerical integration. From the point of view of tracking the due-date signal, Method 2 is generally better than the other three methods.

Samenvatting

Optimalisatie en modelgebaseerde regeling van max-plus-lineaire en continue stuksgewijs affiene systemen

Dit proefschrift gaat over de ontwikkeling van optimalisatie- en modelgebaseerde regelmethoden voor max-plus-lineaire (MPL) en continue stuksgewijs affiene (in het Engels: *piecewise affine*, PWA) systemen. Dit proefschrift onderzoekt de volgende drie hoofdthema's:

1. Optimistische optimalisatie en planning voor modelgebaseerde regeling van MPL-systemen

   Modelgebaseerde voorspellende regeling (in het Engels: *model predictive control*, MPC) voor MPL-systemen leidt meestal tot een niet-glad, niet-convex optimalisatieprobleem met reële variabelen. Omdat het moeilijk is om een dergelijk probleem efficiënt te oplossen, wordt een alternatieve benadering ontwikkeld om het gegeven probleem om te zetten in een gemengd integer lineair programmeringsprobleem (in het Engels: *mixed integer linear programming*, MILP). De rekencomplexiteit van de huidige MILP-algoritmen neemt echter in het slechtste geval exponentieel toe als een functie van de voorspellingshorizon. In dit proefschrift passen we optimistische optimalisatie aan voor het oplossen van het MPC-optimalisatieprobleem voor MPL-systemen. Optimistische optimalisatie is een klasse van algoritmen die een benadering van het globale optimum kunnen vinden van algemene niet-lineaire optimalisatieproblemen. Een belangrijk voordeel van optimistische optimalisatie is dat het niveau van bijna-optimaliteit van de resulterende oplossing in vergelijking met het globale optimum kan gekarakteriseerd worden, gegeven een eindig rekenbudget. We beschouwen MPC voor MPL-systemen met begrenzingen op de incrementen van de regelingangen. De doelfunctie is een afweging tussen de uitgangskosten (d.w.z. een gewogen traagheids-vroegheidstraf met betrekking tot een vervaldatumsignaal) en de ingangskosten (d.w.z. grondstoffen zo laat mogelijk voeden aan het systeem). We ontwikkelen een specifieke semi-metriek die voldoet aan de noodzakelijke vereisten voor optimistische optimalisatie. Gebaseerd op de theoretische analyse bewijzen we dat de complexiteit van optimistische optimalisatie exponentieel is in de regelhorizon in plaats van de voorspellingshorizon. Daarom is optimistische optimalisatie rekenkundig efficiënter wanneer de regelhorizon klein is en de voorspellingshorizon groot.

   Daarnaast pakken we het oneindige-horizon optimale-regelingprobleem voor MPL-systemen aan waarbij de beschouwde doelfunctie een som is van gedisconteerde kosten per fase over een oneindige horizon. We beschouwen de incrementen van de regelingangen als regelvariabelen en we discretiseren de ruimte van regelingangen als een eindige verzameling. Het resulterende optimale-regelingprobleem wordt op gelijkaardige wijze omgezet in een online planningsprobleem dat gericht is op het maximaliseren van een beloningsfunctie. We passen het optimistisch planningsalgoritme aan om dit probleem op te lossen. Voor een gegeven eindig rekenbudget wordt een regelsequentie berekend en de eerste regelactie of een deelrij van de berekende regelsequentie wordt toegepast op het systeem. Vervolgens wordt een schema gebruikt met een glijdende horizon. De voorgestelde optimistische planningsbenadering levert een karakterisering op van de bijna-optimaliteit van de resulterende oplossing. De simulatieresultaten tonen aan dat wanneer een subsequentie van de berekende regelsequentie wordt toegepast, deze benadering resulteert in een lagere volgfout (in het Engels: *tracking error* vergeleken met een eindige-horizonbenadering.

2. Optimistische optimalisatie voor MPC van continue PWA-systemen

   We passen optimistische optimalisatie verder aan voor het oplossen van het MPC-optimalisatieprobleem voor continue PWA-systemen. De beschouwde doelfuncties op basis van een 1-norm of een ∞-norm zijn continue PWA-functies. De lineaire beperkingen op de toestanden en de ingangen worden behandeld als zachte beperkingen en vervangen door het toevoegen van een strafterm aan de doelfunctie. De voorgestelde optimistische optimalisatiebenadering is gebaseerd op recursieve partitionering van de resulterende hyperbox verzameling van toegelaten waarden. We leiden uitdrukkingen af voor de kernparameters van optimistische optimalisatie en we bespreken de bijna-optimaliteit van de resulterende oplossing. De prestatie van de voorgestelde aanpak wordt geïllustreerd met een *case study* over adaptieve snelheidsregeling.

   Uit de resultaten van het eerste deel van dit onderwerp kunnen we zien dat het optimalisatieprobleem met een continue niet-convexe PWA-functie ontstaat in de context van regeling van continue PWA-systemen. In de literatuur is aangetoond dat dit type optimalisatieprobleem kan worden geformuleerd als een MILP-probleem, waarvan de complexiteit in het slechtste geval exponentieel groeit met het aantal polyhedrale deelgebieden in het domein van de PWA-functie. In het eerste deel passen we optimistische optimalisatie toe om globale optimalisatieprobleem van een continue niet-convexe PWA-functie op te lossen met een hyperbox als verzameling van toegelaten waarden. Hierbij werden de beperkingen op de toestanden en ingangen behandeld als zachte beperkingen. In het tweede deel breiden we optimistische optimalisatie uit van een hyperbox als verzameling van toegelaten waarden naar een polytopische verzameling van toegelaten waarden. In het bijzonder stellen we een partitioneringsraamwerk voor van de polytopische verzameling van toegelaten waarden die voldoet aan de vereisten van optimistische optimalisatie door gebruik te maken van Delaunay triangulatie en onderverdeling van de randen. Voor deze partitioneringsaanpak leiden we analytische uitdrukkingen af voor de basisingrediënten die worden gebruikt voor het karakteriseren van de bijna-optimaliteit van de oplossing verkregen door optimistische optimalisatie. Voor het optimaliseren van PWA-functies is de voorgestelde optimistische optimalisatiemethode rekenkundig efficiënter dan MILP als het aantal polyhedrale deelgebieden in het domein veel groter is dan het aantal variabelen van de PWA-functie.

3. MPC voor stochastische MPL-systemen met kansbeperkingen

   Het onderwerp van het laatste deel van dit proefschrift is MPC voor MPL-systemen met stochastische onzekerheden waarvan de kansverdeling verondersteld wordt bekend te zijn. We beschouwen lineaire beperkingen op de ingangen en de uitgangen. Vanwege de onzekerheden worden deze lineaire beperkingen geformuleerd als probabilistische of toevallige beperkingen, d.w.z. de beperkingen moeten voldaan zijn met een vooraf bepaald waarschijnlijkheidsniveau. De voorgestelde probabilistische beperkingen kunnen op equivalente wijze worden herschreven in een max-affiene vorm (d.w.z. het maximum van affiene termen) als de lineaire beperkingen monotoon niet-dalend zijn als een functie van de uitgangen. Gebaseerd op de resulterende max-affiene vorm worden twee methoden ontwikkeld voor het oplossen van het MPC-probleem voor stochastische MPL systemen met kansbeperkingen: methode 1 gebruikt de ongelijkheid van Boole om de multi-variable kans om te zetten in kansbeperkingen in 1 variable en waarvoor de waarschijnlijkheid efficiënter kan worden berekend. Methode 2 maakt gebruik van de multidimensionale Chebyshev-ongelijkheid en transformeert de multi-variable kansbeperkingen in beperkingen die lineair zijn in de ingangen. Door middel van een voorbeeld van een productiesysteem worden de twee voorgestelde methoden vergeleken met de numerieke integratiemethode en de nominale MPC-methode. Vanuit het oogpunt van rekentijd zijn de nieuw ontwikkelde methoden sneller dan numerieke integratie. Vanuit het oogpunt van het volgen van het vervaldatumsignaal is methode 2 in het algemeen beter.

Curriculum Vitae

Jia Xu was born on July 10, 1987 in Dezhou, Shandong Province, China. In July 2009, she obtained the BSc degree in Statistics from the School of Mathematics and Statistics, Shandong University at Weihai, China. In September 2009, she became a master student in the School of Mathematical Sciences, Tongji University, Shanghai, China. In March 2011, she was recommended as a PhD student in the Department of Control Science and Engineering, College of Electronics and Information Engineering, Tongji University.

In September 2012, she was sponsored by the Chinese Scholarship Council to become a PhD student at the Delft Center for Systems and Control, Delft University of Technology, The Netherlands. In her PhD project, she worked on optimization and model-based control for max-plus linear and continuous piecewise affine systems, under the supervision of Prof.dr.ir Bart De Schutter and Dr.ir. Ton van den Boom. At the end of her first year, she obtained the certificate of the Dutch Institute of Systems and Control (DISC). In 2015, she had the opportunity to visit the Automation Department of the Technical University of Cluj-Napoca and collaborate with Prof.dr. Lucian Buşoniu. Her research interests are model predictive control, optimization, discrete-event systems, and hybrid systems.