Approximation Methods in Stochastic Max-Plus Systems

S. S. Farahani

Cover design: Maryam (Narjes) Shojai Kaveh
Cover illustrations: The robot is called Zebro and is designed at DCSC robotic laboratory. The other photos are downloaded from Wikipedia.

Approximation Methods in Stochastic Max-Plus Systems

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op woensdag 28 november 2012 om 12:30 uur
door Samira SAFAEI FARAHANI
Master of Science in Applied Mathematics,
Delft University of Technology,
geboren te Tehran, Iran.

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. B. De Schutter

Copromotor: Dr.ir. T. van den Boom

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof.dr.ir. B. De Schutter	Technische Universiteit Delft, promotor
Dr.ir. T. van den Boom	Technische Universiteit Delft, copromotor
Prof.dr. R. Babuška	Technische Universiteit Delft
Prof.dr. G.J. Olsder	Technische Universiteit Delft
Prof.dr. W. McEneaney	University of California, San Diego
Prof.dr. A. Stoorvogel	Universiteit Twente
Dr. B. Heidergott	Vrije Universiteit Amsterdam

This thesis 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: S. Safaei Farahani
E-mail: s.safaeifarahani@tudelft.nl
WWW: http://www.dcsc.tudelft.nl/~sfarahani/

ISBN 978-94-6203-219-4

Keywords: stochastic max-plus linear systems, model predictive control, approximation methods, moments of random variables, switching max-plus linear systems, max-min-plus-scaling systems.

Printed by Wöhrmaan Print Service, Zutphen, the Netherlands

“Fences are made for those who cannot fly.”

Elbert Hubbard

Contents

Preface

Time flies. This phrase has stuck to my mind in the last few months. It is still hard for me to believe that these four years have passed and I am now writing the very last piece of my dissertation. The best part is that I enjoyed every moment of my PhD study despite all the ups and downs.

This thesis is not only the result of my work, but it reflects all the help and support I have received from so many people during these four years.

My deepest gratitude goes to my advisor (promotor) Bart De Schutter without whom finishing this PhD project was impossible. His endless energy and enthusiasm were always encouraging for me and his valuable and precise comments as well as the extensive and useful discussions we had were constantly enlightening my way during these four years. He taught me to be patient and persevering not only in doing research but also in other aspects of life. I thank him for believing in me and my abilities and for supporting me in the hard moments. *Bart, jij bent diegene die ik het meeste te danken heb!*

I am undoubtedly indebted to my supervisor Ton van den Boom who was always there for me to answer my questions and to give priceless ideas for my research. His brightness and sharpness were definitely effective elements during our discussions and his sense of humor and openness always made me feel comfortable and trusted. His wise advices were always my savior when I couldn't find my way through. *Ton, bedankt voor alles!*

My wholehearted gratitude goes to William McEneaney for many illuminating discussions we had during my research visit to UCSD and for all his great tips and ideas during those discussions. I am also grateful to him for giving me the opportunity of meeting renowned scholars in my filed by inviting me to a workshop in Monterey.

I would like to thank all my committee members for taking time and for participating in my defense ceremony: Prof. Robert Babuška, Prof. Geert-Jan Olsder, Prof. Anton Stoorvogel, Prof. William McEneaney, and dr. Bernd Heidergott. Their constructive comments and suggestions helped me improve my dissertation in the final stage.

I am grateful to Peter Heuberger for all valuable discussions we had and for providing me with several useful references for my research. I would also like to thank Prof. Ioan Landau for his useful comments on identification of max-plus-linear systems. I am very grateful to dr. Bernd Heidergott for giving me helpful suggestions during my research and for answering my questions in great detail.

I enjoyed working with Stefan van Loenhout during his master project and I thank him for providing the material I needed for stochastic switching max-plus-linear systems. I am truly grateful to my friend and office mate Noortje for all her friendship and support during these four years. I especially thank her for translating my summary into Dutch. I am thankful to all my friends and colleagues at DCSC, especially to Ali, Pawel, Ilhan, Eric, Lakshmi, Amol, Dang, Andrea, Yashar, Mohammad, Jelmer, Ilya, Gabriel, and Sneszana, for letting me have unforgettable great memories during my PhD studies.

Without the supporting staff at DCSC, life would have been very hard. I thank them all especially Ellen van den Berg for arranging a new office for me, Kitty Dukker for being a *duizendpoot* in DCSC, Esther Booij for handling all paperwork, Olaf Voorwinde for taking care of all the finance related issues, and Will van Geest for helping me solve the problems I had with my PC.

I also thank my dearest friends Nikoo, Sanaz, Jairan, Ghazaleh, and Tina for being next to me all the time, for listening to me, and for their heartwarming presence in all moments. I am so proud to have you as my friends.

I am deeply grateful to my mom and dad for all their support and encouragement during these four years and through my whole life. Without them, I would not achieve what I have today. I also thank my sweet brother for being a good friend and a great listener.

The last sentences are for the most beloved person of my life, Behnam. First of all, I would like to thank you for not letting me quit the PhD during the first year; in a sense, I owe my PhD to you. I also thank you for all the serenity and joy you have brought into my life. Your presence was, and is, a relief to my mind and your constant support and love was helping me go through difficult moments during my PhD study.

Samira S. Farahani

Delft, November 2012

Chapter 1

Introduction

1.1 Motivation

Not only in the natural sciences (such as physics, biology, earth science, meteorology) and engineering disciplines (such as computer science, artificial intelligence), but also in the social sciences (such as economics, psychology, sociology and political science), mathematical models are used extensively in order to explain a system and to make predictions about the behavior of that system [9, 66, 71]. Based on the nature of the system, mathematical models can be classified in some of the following ways: linear vs. nonlinear, deterministic vs. stochastic, static vs. dynamic, and discrete vs. continuous. In this thesis, we consider models that belong to the class of discrete-event or hybrid systems.

In discrete-event systems, the state evolution depends entirely on the occurrence of discrete events over time. In this type of systems, the state components refer to the starting time of the activities and events refer to the start or the end of those activities. For example, in a manufacturing system, a state could be the time instant at which a particular machine starts working and an event is a machine breaking down or a part entering a buffer. Typical examples of discrete-event systems are telecommunication networks, manufacturing systems, parallel computing, traffic control systems, railway networks, etc.

Hybrid systems are characterized by the interaction of time-continuous models on the one hand, and logic rules and discrete-event models on the other hand. As a result, the evolution of a hybrid system may depend both on the progress of time and the occurrence of events. Typically when an event occurs, the system switches to a new operation mode. At each mode, the continuous time evolution of the system is governed by a different set of differential or difference equations. Typical examples of hybrid systems are manufacturing systems, computer networks, digital circuits, and logistic systems.

When selecting the most appropriate model for a discrete-event or a hybrid system, there is always a trade-off between the modeling power and the decision power; in other words, modeling frameworks that can describe large and general classes of discrete-event and hybrid systems are, in general, less amenable to (efficient) mathematical analysis. Some examples of modeling frameworks for discrete-event and hybrid systems are queuing theory, (extended) state machines, formal languages, automata, temporal logic models, generalized semi-Markov processes, Petri nets, and computer simulation models [19, 58, 91, 114, 115]. As a result, a special class of discrete-event models, namely the max-plus-linear (MPL) models, has been introduced that are amenable to mathematical analysis [4, 23, 57]. Such models apply to discrete-event systems with synchronization but no choice (e.g. an assembly line) and are described using max-plus-scaling functions, i.e., functions that are constructed using the operations maximization, addition, and multiplication by a scalar.

An extension of this class is the class of switching MPL systems [108], in which the state of the system can switch between different modes of operation, e.g. a production system that has different recipes for different products and hence, the system switches to a different mode to produce each product. At each mode, the system is described by an MPL model with particular system matrices for that mode. Other examples of switching MPL systems are a railway network or a telecommunication network. We study also the class of min-max-plus-scaling (MMPS) systems, which includes both hybrid and discrete-event systems and which are equivalent to a particular class of hybrid systems, namely continuous piecewise affine (PWA) systems [49, 55, 88]. In MMPS systems, the system dynamics are defined by MMPS expressions, i.e., expressions constructed using the operations maximization, minimization, addition, and multiplication by a scalar. Digital circuits are an example of MMPS systems. In all these systems, we can consider a stochastic structure imposed by the presence of model mismatch and/or disturbances [4, 57, 75].

To control stochastic systems belonging to the above-mentioned classes, one efficient control approach is model predictive control (MPC) [18, 44, 74]. MPC is an online model-based approach, in which at each event step (for discrete-event systems) or time step (for hybrid systems) an optimal control sequence is computed. This optimization is done over a finite series of events or finite period of time, and at each event or time step, only the first sample of the optimal control sequence will be applied to the system. In the next step, the horizon will be shifted forward and the new optimal control sequence will be computed. In the stochastic systems, the objective function defined in the MPC optimization problem consists of an expected value of stochastic max-plus-scaling functions and MMPS expressions. Hence, solving this optimization problem creates a considerable computational complexity due to the presence of the expected value. In general, the expected value is computed using either numerical integration or some available analytic approaches, which are all very time-consuming and complex.

1.2 Research Goals

The aim of this PhD research is to develop (approximation) methods, to compute the expected value of max-plus-scaling and MMPS functions with the focus on reducing the computational complexity and the computation time. Since the currently used methods, i.e., numerical or analytic integration, are computationally quite complex, MPC for stochastic (switching) MPL or MMPS systems has only been studied in the literature for systems with uniformly distributed or bounded noise, which causes limitations as in practice this distribution cannot always capture all the stochastic properties of the system.

Accordingly, we propose an approximation method to compute the expected value of max-plus-scaling and MMPS functions. This approximation method, which is the core of this thesis, is inspired by the relation between different types of vector norms, namely the p-norm and the ∞-norm. Using this approximation approach, we obtain an upper bound for the expected value of stochastic max-plus-scaling and MMPS functions that can be used as a replacement of the expected value itself when minimizing the objective function. This approach allows us to consider stochastic random variables with any probability distribution that has finite moments of which a closed form exists, such as the uniform distribution, the normal distribution, the beta distribution, etc.

After obtaining this approximation method, we apply it to address MPC and identification of stochastic MPL systems, MPC for stochastic switching MPL systems, and MPC for MMPS systems. The proposed approximation method simplifies the computations considerably. Moreover, we show in the case studies that by choosing the appropriate order of the moments, the approximation error can be made sufficiently small. In these examples, we obtain a much faster and more efficient way to address MPC and identification of different classes of max-plus systems with a performance that is comparable to the one using the existing methods to compute the expected value, such as numerical or analytic integration and Monte Carlo simulation.

1.3 Overview of the Thesis

This PhD thesis starts with a short overview on previous work done on MPC and identification of MPL and other classes of max-plus systems (Chapters 1-3) and then follows by presenting the main contribution of this thesis (Chapter 4), which is an approximation approach for computing the expected value of max-plus functions, as well as addressing MPC and identification of the mentioned classes using this new approach (Chapters 5-7). The relation between the chapters of this thesis is illustrated in Figure 1.1.

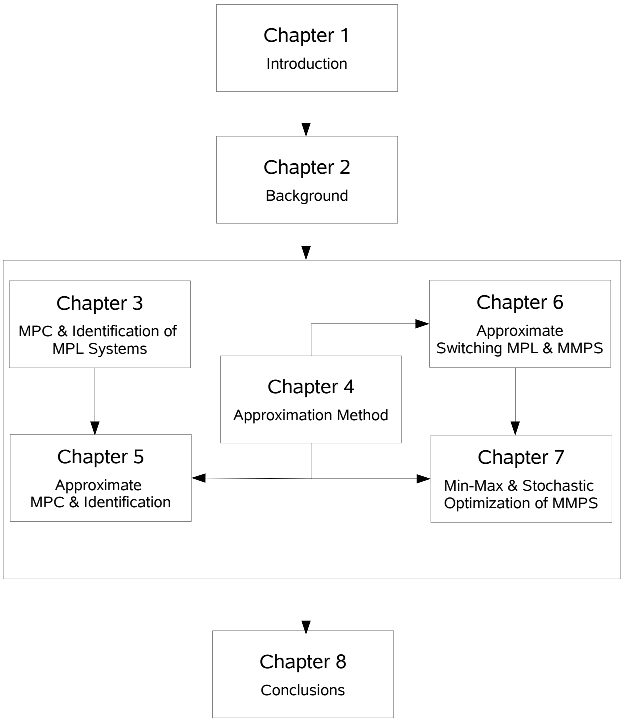

Chapter 1
Introduction
Chapter 2
Background
Chapter 3
MPC & Identification of MPL Systems
Chapter 4
Approximation Method
Chapter 5
Approximate MPC & Identification
Chapter 6
Approximate Switching MPL & MMPS
Chapter 7
Min-Max & Stochastic Optimization of MMPS
Chapter 8
Conclusions

Figure 1.1: Overview of the thesis

The thesis is organized as follows:

Chapter 2

This chapter provides background information on the max-plus algebra. Since we study different classes of MPL systems in this thesis, Chapter 2 gives a brief overview on each of these classes, namely the classes of stochastic MPL systems, stochastic switching MPL systems, and stochastic MMPS systems. For each of these systems, there is a simple example in order to make a clearer understanding of these systems.

Chapter 3

In Chapter 3, we give a concise account of MPC for stochastic MPL systems with an extension to switching MPL systems and MMPS systems. Moreover, identification of stochastic MPL systems is also discussed briefly. Both for MPC and identification, we explain the methodology and the computational procedure. We also present the currently applied methods to solve the MPC and identification optimization problem of these systems and discuss the computational complexities imposed by these methods.

Chapter 4

Chapter 4 introduces the new approximation approach based on moments of random variables and describes how it reduces the complexity of MPC and identification optimization problems. The main advantage of this approach is that it results in an analytic solution in the case that random variables have finite moments with a closed form and hence, it reduces the computation time of the optimization procedure as well. We also discuss the error of this approximation method and show that this error is bounded both from below and from above. Furthermore, the convexity of the approximation function obtained from this method is discussed. Having a convex approximation function, in many cases, results in a convex optimization problem, which can be solved efficiently.

Chapter 5

Chapter 5 is dedicated to the application of the approximation method to MPC and identification of stochastic MPL systems. In both cases, the approximate objective function will be defined, which is indeed an upper bound for the true objective function. Since this approximate function can be computed analytically, the computation time of the approximate optimization problem is very short and the computational complexity is much less than the one using numerical or analytic integration. We present examples with different types of distributions (with bounded and unbounded domain) for both MPC and identification of stochastic MPL systems to compare the results obtained using the approximation method with the results obtained using other methods, such as numerical or analytic integration and Monte Carlo simulation.

Chapter 6

In Chapter 6, we extend the approximation method further to two other classes of discrete-event systems, namely the stochastic switching MPL systems and the stochastic MMPS systems. Here again, we define an approximate objective function using the approximation method to obtain an upper bound for the objective function. For switching MPL systems, we consider both stochastic switching and stochastic system parameters at the same time and hence, we study the cases in which these two stochastic phenomena are independent of or dependent on each other. In case of MMPS systems, in order to be able to apply the approximation method, we first need to rewrite the MMPS objective function as a difference of two convex functions and next, apply the approximation method. For both switching MPL systems and MMPS systems, we present worked examples to compare the performance of the approximation method with other existing methods.

Chapter 7

Min-max optimization of MMPS systems appears in different problems defined for discrete-event and hybrid systems. Chapter 7 presents the min-max and stochastic optimization problem of MMPS systems. We study the min-max optimization over bounded control variables, the minimization of MMPS functions with stochastic variables, and the min-max optimization of MMPS functions with stochastic variables and provide a solution approach for each of them. We also present two applications, namely the filtering problem and the reference tracking problem, in which the min-max and stochastic optimization of MMPS functions appear.

Chapter 8

Chapter 8 concludes the thesis and provides some recommendations for future research. We present the main contributions of this thesis as well as suggesting further research on some related topics in order to improve the approximation method and some other interesting topics related to max-plus systems.

1.4 Publications

Most of the material presented in Chapters 4-6 of this PhD thesis has been published in conference proceedings or submitted to peer reviewed journals. The link between each chapter and the publications is as follows:

- Chapter 4 is based on the paper [37–39]
- Chapter 5 is based on the papers [35, 37–39]
- Chapter 6 is based on the papers [36, 113]

Chapter 2

Background

In this chapter, we present a brief overview of max-plus algebra, followed by a concise description of some special classes of discrete-event systems such as stochastic max-plus-linear (MPL) systems, stochastic switching MPL systems, and stochastic max-min-plus-scaling (MMPS) systems. Since the main focus of this thesis is on providing an approximation method in order to increase the computational efficiency of the control process of such systems, it is useful to first become acquainted with the general description of these systems.

2.1 Max-Plus Algebra

Define $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$ and $\varepsilon = -\infty$. The max-plus addition ($\oplus$) and multiplication ($\otimes$) are defined as follows:

$$\begin{aligned} x \oplus y &= \max(x, y) \\ x \otimes y &= x + y \end{aligned}$$

for $x, y \in \mathbb{R}_\varepsilon$. The zero element of the max-plus addition is ε, i.e., $x \oplus \varepsilon = x$, and the identity element of the max-plus multiplication is 0, i.e., $x \otimes 0 = x$. The set $\mathbb{R}_\varepsilon$ together with the operators $\oplus$ and $\otimes$ is called max-plus algebra and is denoted by $\mathcal{R}_\varepsilon = (\mathbb{R}_\varepsilon, \oplus, \otimes, \varepsilon, 0)$ [4]. Note that $\mathcal{R}_\varepsilon$ is a *semifield* since:

- the operation $\oplus$ is associative and commutative;
- the operation $\otimes$ is distributive with respect to $\oplus$ and its identity element 0 satisfies $\varepsilon \otimes 0 = 0 \otimes \varepsilon = \varepsilon$.

$\mathcal{R}_\varepsilon$ is also *idempotent* since the first operation is idempotent, i.e., $x \oplus x = x, \forall x \in \mathbb{R}_\varepsilon$, and it is commutative, i.e., $x \otimes y = y \otimes x$.

The corresponding max-plus matrix operations are defined [4] as

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})$$
$$(A \otimes C)_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kj} = \max_{k=1,\ldots,n} (a_{ik} + c_{kj})$$

for $A, B \in \mathbb{R}_{\varepsilon}^{m \times n}$ and $C \in \mathbb{R}_{\varepsilon}^{n \times p}$. We denote the i-th row of matrix A by $A_{i,\cdot}$ and the j-th column by $A_{\cdot,j}$. To avoid confusion in the sequel, we drop the multiplication sign in conventional algebra expressions while keeping the $\otimes$ sign in max-plus expressions.

Now let $\mathcal{S}_{\text{mps}}$ denote the set of max-plus-scaling functions, i.e., functions f of the form

$$f(z) = \max_{i=1,\ldots,m} (\tau_{i,1} z_1 + \cdots + \tau_{i,n} z_n + \xi_i) \tag{2.1}$$

with variable $z \in \mathbb{R}_{\varepsilon}^{n}$ and constant coefficients $\tau_{i,j} \in \mathbb{R}$ and $\xi_i \in \mathbb{R}$. In the sequel, we stress that f is a max-plus-scaling function of z by writing $f \in \mathcal{S}_{\text{mps},z}$.

Remark 2.1.1 Let $k \in \mathbb{N}/\{0\}$. Then for $x \in \mathbb{R}_{\varepsilon}$, the max-plus power is defined as $x^{\otimes^{k}} = k \cdot x$ and – by definition – $x^{\otimes^{0}} = 0$. Therefore, a max-plus-scaling function f can also be defined using max-plus notations as follows:

$$f(z) = \bigoplus_{i=1}^{m} z_1^{\otimes^{\tau_{i,1}}} \otimes \cdots \otimes z_n^{\otimes^{\tau_{i,n}}} \otimes \xi_i.$$

with variable $z \in \mathbb{R}_{\varepsilon}^{n}$ and constant coefficients $\tau_{i,j} \in \mathbb{R}$ and $\xi_i \in \mathbb{R}$. However, in this thesis, we use the expression in (2.1) for ease of accessibility and to emphasize linearity in terms of conventional algebra. □

As shown in [104], the set $\mathcal{S}_{\text{mps}}$ is closed under the operations $\oplus$, $\otimes$, and the scalar multiplication. The in-depth discussion on max-plus algebra can be found in [4, 23, 57].

2.2 Stochastic Max-Plus-Linear (MPL) Systems

Discrete-event systems form a large class of dynamic systems in which the evolution of the system is specified by the occurrence of certain discrete events, unlike continuous dynamic systems where the state of the system changes as time progresses. For such systems, there exist different modeling frameworks such as queuing theory, (extended) state machines, formal languages, automata, temporal logic models, generalized semi-Markov processes, Petri nets, and computer simulation models

[19, 58, 91]. Models of such systems are in general nonlinear in conventional algebra. However, there exists an important class of discrete event systems, namely the max-plus-linear (MPL) systems, for which the model is *linear* in the max-plus algebra.

The class of MPL systems consists of discrete-event systems with synchronization but no choice. Synchronization requires the availability of several resources at the same time, whereas choice appears, e.g., when some user must choose among several resources [4]. Typical examples of such systems are serial production lines, production systems with a fixed routing schedule, and railway networks. In stochastic systems, processing times and/or transportation times are assumed to be stochastic quantities, since in practice such stochastic fluctuations can, e.g. be caused by machine failure or depreciation [87]. Another type of error is modeling errors, which again leads to errors in the system matrices. Related topics on (stochastic) MPL systems including analysis, controller design, etc., can be found in [4, 6, 57, 75, 87, 92].

A stochastic MPL system can be modeled [4, 23] as follows:

$$x(k) = A(k) \otimes x(k-1) \oplus B(k) \otimes u(k) \tag{2.2}$$

$$y(k) = C(k) \otimes x(k) \tag{2.3}$$

where $A(k), B(k)$ and $C(k)$ are system matrices, which are perturbed by noise and/or modeling errors by assumption, $x(k)$ is the state of the system at event step k, and $u(k)$ and $y(k)$ are the input and output of the system. In fact, the vectors $x(k)$, $u(k)$, and $y(k)$ contain the time instants at which the internal, input, and output events occur for the k-th time, respectively. Modeling mismatch and disturbances perturb the system by introducing uncertainty in the system matrices. Sometimes it is difficult to distinguish between these two, but usually fast changes in the system matrices will be considered as noise and disturbances, whereas slow changes or permanent errors are considered as model mismatch. Following Van den Boom et al. [109], the uncertainties in the system are presented in a single framework, using one stochastic vector $e(k)$ with a certain probability distribution. Hence, the entries of the system matrices belong to $\mathcal{S}_{\text{mps}}$ (see [104]), i.e., $A(k) \in \mathcal{S}^{n_x \times n_x}_{\text{mps},e(k)}$, $B(k) \in \mathcal{S}^{n_x \times n_u}_{\text{mps},e(k)}$, $C(k) \in \mathcal{S}^{n_y \times n_x}_{\text{mps},e(k)}$, which is illustrated in Example 2.2.2 below.

Remark 2.2.1 In practice, the order of MPL systems can range from about 10 (e.g. for small production systems) to more than 200 (e.g., for a large scale railway network). The examples presented in this thesis consider systems with a small system order due to two reasons. The first reason is that this is a didactic choice, which allows more insight into the problem. The second reason is that in the following chapters, we will compare our proposed approximation methods with some other available methods from the literature and since some of these methods are often very slow and inefficient for systems of large order, our examples are chosen accordingly. □

Example 2.2.2 A simple production system with stochastic processing time [27]

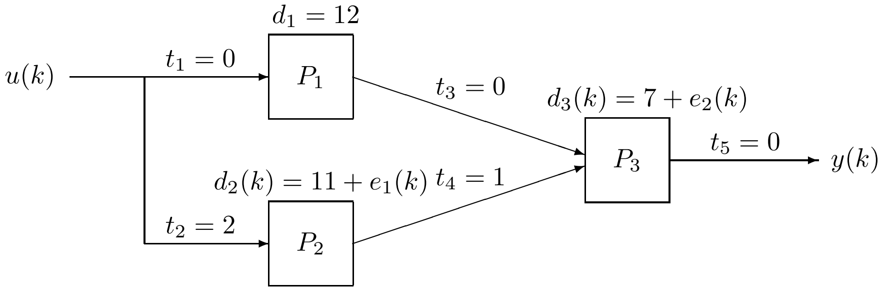


Figure 2.1: A simple production system.

Consider the system of Figure 2.1. This production system consists of three processing units: P_1, P_2, and P_3. Raw material is fed to P_1 and P_2 to be processed, and then it is sent to P_3 where assembly takes place. The processing times for P_1 is $d_1 = 12$ time units. We assume that the processing time for P_2 and P_3 are perturbed by noise, i.e., $d_2(k) = d_2 + e_1(k)$ and $d_3(k) = d_3 + e_2(k)$ where $d_2 = 11$, $d_3 = 7$, and $e(k) = [e_1(k)e_2(k)]^T$ is an stochastic vector with independent elements and with a given probability distribution (e.g. the uniform distribution or the normal distribution). We assume that it takes $t_2 = 2$ time units for the raw material to get from the input source to P_2 and that it takes $t_4 = 1$ time unit for the finished products of processing unit P_2 to reach P_3. The other transportation times (t_1, t_3, and t_5) are assumed to be negligible. We assume that at the input of the system and between the processing units there are buffers with a capacity that is large enough to ensure that no buffer overflow will occur. We consider the situation where initially all buffers are empty and none of the processing units contains raw material or intermediate products.

A processing unit can only start working on a new product if it has finished processing the previous product. We assume that each processing unit starts working as soon as all parts are available. Define

- $u(k)$: time instant at which raw material is fed to the system for the k-th time,
- $x_i(k)$: time instant at which the i-th processing unit starts working for the k-th time,
- $y(k)$: time instant at which the k-th finished product leaves the system.

Let us now determine the time instant at which processing unit P_1 starts working for the k-th time. If we feed raw material to the system for the k-th time, then this raw material is available at the input of processing unit P_1 at time $t = u(k) + t_1$. However, P_1 can only start working on the new batch of raw material as soon as it has finished processing the previous, i.e., the $(k-1)$-th, batch. Since the processing time on P_1 is $d_1 = 12$ time units, the $(k-1)$-th intermediate product will leave P_1 at time $t = x_1(k-1) + d_1$. Since P_1 starts working on a batch of raw material as soon as the raw material is available and the current batch has left the processing unit, this implies that we have

$$x_1(k) = \max(x_1(k-1) + d_1,\ u(k) + t_1\,) \tag{2.4}$$

for $k = 1, 2, \ldots$ The condition that initially processing unit P_1 is empty and idle corresponds to the initial condition $x_1(0) = \varepsilon$ since then it follows from (2.4) that $x_1(1) = u(1)$, i.e., the first batch of raw material that is fed to the system will be processed immediately.

Using a similar reasoning, we find the following expressions for the time instants at which P_2 and P_3 start working for the k-th time and for the time instant at which the k-th finished product leaves the system:

$$\begin{aligned} x_2(k) &= \max(x_2(k-1) + d_2(k-1),\ u(k) + t_2\,) && (2.5)\\ x_3(k) &= \max(x_1(k) + d_1 + t_3, x_2(k) + d_2(k) + t_4,\ x_3(k-1) + d_3(k-1)\,)\\ &= \max(x_1(k-1) + 2d_1 + t_3,\ x_2(k-1) + d_2(k-1) + d_2(k) + t_4,\\ &\qquad x_3(k-1) + d_3(k-1), u(k) + d_1 + t_1 + t_3,\ u(k) + d_2(k) + t_2 + t_4\,) && (2.6)\\ y(k) &= x_3(k) + d_3(k) + t_5 && (2.7) \end{aligned}$$

for $k = 1, 2, \ldots$. The condition that initially all buffers are empty corresponds to the initial condition $x_1(0) = x_2(0) = x_3(0) = \varepsilon$.

Let us now rewrite the evolution equations of the production system in a max-plus format. It is easy to verify that (2.4) can be rewritten as

$$x_1(k) = d_1 \otimes x_1(k-1) \oplus t_1 \otimes u(k) \ .$$

Likewise, (2.5)–(2.7) can be written as follows:

$$\begin{aligned} x_2(k) &= d_2(k-1) \otimes x_2(k-1) \oplus t_2 \otimes u(k),\\ x_3(k) &= 2d_1 \otimes t_3 \otimes x_1(k-1) \oplus d_2(k-1) \otimes d_2(k) \otimes t_4 \otimes x_2(k-1) \oplus\\ &\quad d_3(k-1) \otimes x_3(k-1) \oplus d_1 \otimes t_1 \otimes t_3 \otimes u(k) \oplus d_2(k) \otimes t_2 \otimes t_4 \otimes u(k),\\ y(k) &= d_3(k) \otimes t_5 \otimes x_3(k). \end{aligned}$$

If we now rewrite the above equations in max-plus-algebraic matrix notation, we obtain

$$\begin{aligned} x(k) &= A(k)\otimes x(k{-}1)\oplus B(k)\otimes u(k) \\ y(k) &= C(k)\otimes x(k)\ . \end{aligned}$$

where $x(k) = [x_1(k) \quad x_2(k) \quad x_3(k)]^T$ and with the system matrices $A(k)$, $B(k)$ and $C(k)$ given as follows:

$$A(k) = \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\ \varepsilon & d_2(k-1) & \varepsilon \\ 2d_1+t_3 & d_2(k-1)+d_2(k)+t_4 & d_3(k-1) \end{bmatrix}$$

$$= \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\ \varepsilon & d_2+e_1(k-1) & \varepsilon \\ 2d_1+t_3 & 2d_2+e_1(k-1)+e_1(k)+t_4 & d_3+e_2(k-1) \end{bmatrix},$$

$$B(k) = \begin{bmatrix} t_1 \\ t_2 \\ \max(d_1+t_1+t_3, d_2(k)+t_2+t_4) \end{bmatrix}$$

$$= \begin{bmatrix} t_1 \\ t_2 \\ \max(d_1+t_1+t_3, d_2+e_1(k)+t_2+t_4) \end{bmatrix},$$

$$C(k) = \begin{bmatrix} \varepsilon & \varepsilon & d_3(k) \end{bmatrix}$$

$$= \begin{bmatrix} \varepsilon & \varepsilon & d_3+e_2(k) \end{bmatrix}.$$

Note that this is a model of the form (2.2)–(2.3). Here, $d_2(k) = d_2 + e_1(k) = d_2 \otimes e_1(k)$ and $d_3(k) = d_3 + e_2(k) = d_3 \otimes e_2(k)$, which implies that in MPL systems the influence of noise and disturbances is max-plus-multiplicative, which results in the perturbation of system matrices; this is in contrast to conventional linear systems in which the noise and disturbances are often considered to be additive and hence, are modeled by including an extra term in the system equations. □

2.3 Stochastic Switching MPL Systems[1]

Another class of discrete-event systems is the class of switching MPL systems, which consists of discrete-event systems that can switch between different modes

[1] The discussion on stochastic switching MPL systems is mainly based on the results appearing in [107, 112]

of operation such that each mode, itself, is an MPL system. The mode switching allows us to model a change in the structure of the system, such as breaking a synchronization or changing the order of events. In each mode $\ell \in \{1, \dots, n_L\}$, the system is described by an MPL state equation and an MPL output equation, with different system matrices for each mode, as follows:

$$x(k) = A^{(\ell(k))} \otimes x(k-1) \oplus B^{(\ell(k))} \otimes u(k) \tag{2.8}$$

$$y(k) = C^{(\ell(k))} \otimes x(k) \tag{2.9}$$

where $A^{(\ell(k))} \in \mathbb{R}^{n_x \times n_x}$, $B^{(\ell(k))} \in \mathbb{R}^{n_x \times n_u}$, and $C^{(\ell(k))} \in \mathbb{R}^{n_y \times n_x}$ are the system matrices for the ℓ-th mode. We assume that there are n_L possible modes.

The class of Switching MPL systems with deterministic and stochastic switching contains discrete-event systems with synchronization but no choice, in which the order of events may vary randomly and often cannot be determined a priori. This randomness may be due to e.g. (randomly) changing production recipes, varying customer demands or traffic demands, failures in production units, or faults in transmission links [5, 108]. A stochastic switching MPL system may be characterized by stochastic switching or stochastic system parameters, or both at the same time. For the case of stochastic switching with deterministic parameters, the probability of switching to mode ℓ at event step k may depend on the previous mode $\ell(k-1)$, the previous state $x(k-1)$, the input variable $u(k)$, a (additional) control variable $v(k)$, and the event step k, which can be denoted by [107]

$$P[L(k) = \ell(k) | \ell(k-1), x(k-1), u(k), v(k), k], \tag{2.10}$$

where $L(k)$ is a stochastic variable and $\ell(k)$ is its value[2]. Since (2.10) is a probability, it has the following properties:

$$\begin{aligned} &0 \leq P[L(k) = \ell(k) | \ell(k-1), x(k-1), u(k), v(k), k] \leq 1 \\ &\sum_{\ell(k)=1}^{n_L} P[L(k) = \ell(k) | \ell(k-1), x(k-1), u(k), v(k), k] = 1. \end{aligned}$$

In the case of having only perturbed system parameters with deterministic mode switching, at each mode $\ell \in \{1, \dots, n_L\}$ the system equations are of the form (2.2)-(2.3). In the last case in which both mode switching and systems parameters are stochastic variables, the system equations can be defined as follows [107]:

$$x(k) = A^{(\ell(k))}(e(k)) \otimes x(k-1) \oplus B^{(\ell(k))}(e(k)) \otimes u(k) \tag{2.11}$$

[2] As mentioned in [108], to obtain an optimal switching sequence in this case, we can use global random search algorithms such as genetic algorithms [24], tabu search [45], or a branch-and-bound method [22].

$$y(k) = C^{(\ell(k))}(e(k)) \otimes x(k) \tag{2.12}$$

where $A^{(\ell(k))}(e(k)), B^{(\ell(k))}(e(k))$, and $C^{(\ell(k))}(e(k))$ are the system matrices corresponding to mode $\ell(k)$ with a stochastic vector $e(k)$.

In the following example, we illustrate the first case in which the mode changing is stochastic and the system parameters are deterministic.

Example 2.3.1 A production system with stochastic switching between three different recipes [107]

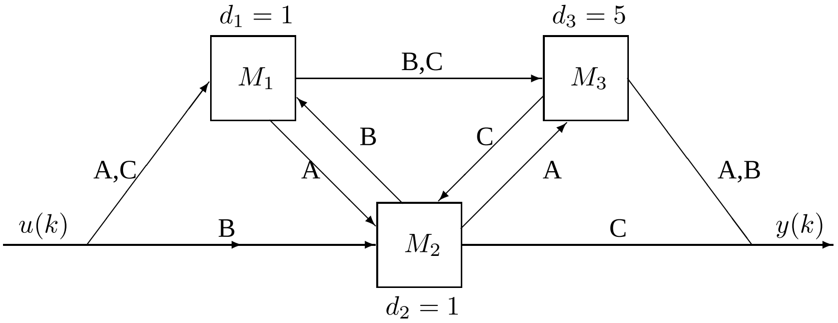


Figure 2.2: A production system with stochastic switching between different recipes.

Consider the production system of Figure 2.2. This system consists of three machines M_1, M_2, and M_3. Three products $(\mathrm{A}, \mathrm{B}, \mathrm{C})$ can be made with this system, each with its own recipe, meaning that the order in the production sequence is different for every product. For product A (using recipe $\ell(k) = 1$) the production order is M_1-M_2-M_3, which means that the raw material is fed to machine M_1 where it is processed. Next, the intermediate product is sent to machine M_2 for further processing, and finally the product A is finished in machine M_3. Similarly, for product B (using recipe $\ell(k) = 2$) the processing order is M_2-M_1-M_3, and for product C (using recipe $\ell(k) = 3$) the processing order is M_1-M_3-M_2. We assume that the type of the k-th product (A, B or C) is available at the start of the production, so that we do know $\ell(k)$ when computing $u(k)$.

Each machine starts working as soon as possible on each batch, i.e., as soon as the raw material or the required intermediate products are available, and as soon as the machine is idle (i.e., the previous batch has been finished and has left the machine). We define $u(k)$ as the time instant at which the system is fed with the raw material for the k-th product, $x_i(k)$ as the time instant at which machine i starts processing the k-th product, and $y(k)$ as time instant at which the k-th product leaves the system. We assume that all the internal buffers are large enough, and no overflow will occur.

We assume the transportation times between the machines to be negligible, and the processing times of the machines M_1, M_2 and M_3 to be given by $d_1 = 1$, $d_2 = 1$ and $d_3 = 5$, respectively. The system equations for recipe A are given by

$$\begin{aligned}
x_1(k) &= \max(x_1(k-1) + d_1, u(k)) \ , \\
x_2(k) &= \max(x_1(k) + d_1, x_2(k-1) + d_2) \\
&= \max(x_1(k-1) + 2d_1, x_2(k-1) + d_2, u(k) + d_1) \ , \\
x_3(k) &= \max(x_2(k) + d_2, x_3(k-1) + d_3) \\
&= \max(x_1(k-1) + 2d_1 + d_2, x_2(k-1) + 2d_2, \\
&\qquad\quad x_3(k-1) + d_3, u(k) + d_1 + d_2) \ , \\
y(k) &= x_3(k) + d_3 \ ,
\end{aligned}$$

leading to the following system matrices for recipe A:

$$A^{(1)} = \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\ 2d_1 & d_2 & \varepsilon \\ 2d_1 + d_2 & 2d_2 & d_3 \end{bmatrix} \quad , \quad B^{(1)} = \begin{bmatrix} 0 \\ d_1 \\ d_1 + d_2 \end{bmatrix} \quad , \quad C^{(1)} = \begin{bmatrix} \varepsilon & \varepsilon & d_3 \end{bmatrix} .$$

Similarly we derive for recipe B:

$$A^{(2)} = \begin{bmatrix} d_1 & 2d_2 & \varepsilon \\ \varepsilon & d_2 & \varepsilon \\ 2d_1 & d_1 + 2d_2 & d_3 \end{bmatrix} \quad , \quad B^{(2)} = \begin{bmatrix} d_2 \\ 0 \\ d_1 + d_2 \end{bmatrix} \quad , \quad C^{(2)} = \begin{bmatrix} \varepsilon & \varepsilon & d_3 \end{bmatrix} \ ,$$

and for recipe C:

$$A^{(3)} = \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\ 2d_1 + d_3 & d_2 & 2d_3 \\ 2d_1 & \varepsilon & d_3 \end{bmatrix} \quad , \quad B^{(3)} = \begin{bmatrix} 0 \\ d_1 + d_3 \\ d_1 \end{bmatrix} \quad , \quad C^{(3)} = \begin{bmatrix} \varepsilon & d_2 & \varepsilon \end{bmatrix} .$$

The switching probability from one recipe to the next one is assumed to be given by:

$$\begin{aligned}
P[L(k) = 1|1, x(k-1), u(k), k] = 0.64 \\
P[L(k) = 1|2, x(k-1), u(k), k] = 0.18 \\
P[L(k) = 1|3, x(k-1), u(k), k] = 0.18 \\
P[L(k) = 2|1, x(k-1), u(k), k] = 0.18 \\
P[L(k) = 2|2, x(k-1), u(k), k] = 0.64 \\
P[L(k) = 2|3, x(k-1), u(k), k] = 0.18
\end{aligned}$$

$$\begin{aligned}P[L(k) = 3|1, x(k-1), u(k), k] = 0.18\\P[L(k) = 3|2, x(k-1), u(k), k] = 0.18\\P[L(k) = 3|3, x(k-1), u(k), k] = 0.64\end{aligned}$$

which means that if we have a specific recipe for product k, then the probability of having the same recipe for product $k+1$ is 64%, and the probability of a switching to each other recipe is 18%. □

2.4 Stochastic Max-Min-Plus-Scaling (MMPS) Systems

A large class of discrete-event and hybrid systems can be described by a max-min-plus-scaling (MMPS) model. Hybrid systems [111] contain both continuous dynamics and switching. Typical examples are manufacturing systems, telecommunication and computer networks, traffic control systems, digital circuits, and logistic systems. In [81] it is shown that the class of MMPS systems encompasses several other classes of discrete-event systems such as continuous piecewise affine (PWA) systems.

Remark 2.4.1 Note that (stochastic) MMPS systems can be both event-driven and time-driven since both discrete-event and hybrid systems can be modeled using the class of (stochastic) MMPS systems. □

Definition 2.4.2 ([29]) $g : \mathbb{R}^n \to \mathbb{R}$ *is a scalar-valued MMPS function of the variables* $x_1, \ldots, x_n$ *if there exists scalar-valued MMPS functions* g_k *and* g_l *such that*

$$\begin{aligned}g(x_i) =& x_i|\alpha|\max(g_k(x), g_l(x))|\min(g_k(x), g_l(x))|\\&g_k(x) + g_l(x)|\beta g_k(x),\end{aligned}$$

where $|$ *stands for "or" and* $\alpha, \beta \in \mathbb{R}$ *for* $i = 1, \ldots, n$*.*

Accordingly, for a vector-valued MMPS function $g : \mathbb{R}^n \to \mathbb{R}^m$, each component of g is an MMPS function of the above form. Furthermore, MMPS functions are dense in the class of continuous functions since they are (cf. Chapter 6) equivalent to continuous PWA functions, which form a dense subspace of continuous functions [69, Section 2.7].

Remark 2.4.3 A natural extension of stochastic MPL systems is stochastic max-min-plus (MMP) systems [60, 86], such as a system describes discrete flows on networks or a traffic network. In this thesis, we take one step further and consider stochastic MMPS systems in which scaling is present as well. □

Accordingly, a state space representation of a deterministic MMPS system can be described as follows:

$$x(k) = \mathcal{M}_x(x(k-1), u(k)) \tag{2.13}$$

$$y(k) = \mathcal{M}_y(x(k), u(k)), \tag{2.14}$$

where $\mathcal{M}_x$, $\mathcal{M}_y$ are MMPS functions, $x(k)$ is the system state, $u(k)$ is the system input, and $y(k)$ is the system output at time or event step k. Similar to conventional linear systems, in stochastic MMPS systems in which noise and modeling errors are present, disturbances and modeling mismatches appear in the systems equations. Hence, the system (2.13)-(2.14) then turns into:

$$x(k) = \mathcal{M}_x(x(k-1), u(k), e(k)) \tag{2.15}$$

$$y(k) = \mathcal{M}_y(x(k), u(k), e(k)). \tag{2.16}$$

As mentioned in the previous sections, we consider both noise and modeling errors in a single framework and present it by the vector $e(k)$, which is a stochastic variable with a given probability density function.

Furthermore, it has been shown in [49, 55, 88] that MMPS systems are equivalent to a particular class of hybrid systems, namely continuous PWA systems.

Definition 2.4.4 ([42]) *A continuous piecewise affine function is defined by a finite number of hyperplanes that divide the whole space into a finite number of convex regions $\mathcal{C}_1 \ldots, \mathcal{C}_r$ surrounded by boundary hyperplanes, and by a set of constant Jacobian matrices $J^{(m)},\ m = 1, 2, \ldots, r$, each of which describes the linear behavior in each region as $J^{(m)}x + b^{(m)}$ where $b^{(m)}$ is a constant vector defined in a given region m, and r denotes the total number of regions, and the function is continuous across the boundaries of the regions .*

For more information on PWA functions and PWA systems we refer to [7, 20, 21, 61, 64, 69, 102] and the references therein.

Proposition 2.4.5 ([49, 88]) *Any MMPS function can be written as a continuous PWA function and vice versa.*

The relation between PWA and MMPS systems is useful for the investigation of structural properties of PWA systems such as observability and controllability but also in designing controller schemes like model predictive control (MPC) [8, 61].

The following example is a sample of a system with MMPS dynamics[3].

[3] Note that this system is in fact an MMP system which is a special case of an MMPS system with the scaling factors equal to 1. Later on in Chapter 6, examples of fully-fledged MMPS systems and functions will be considered explicitly.

Example 2.4.6 A production system with MMPS system dynamics [29]

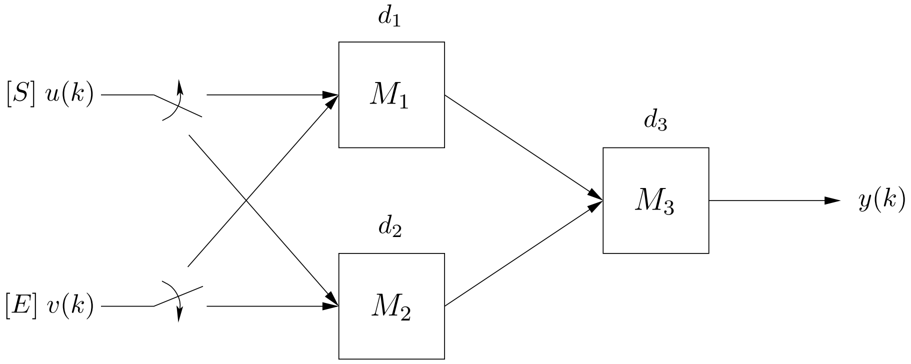


Figure 2.3: A simple manufacturing system with MMPS dynamics.

Consider the production system of Figure 2.3, which consists of three processing units M_1, M_2, and M_3 with processing times d_1, d_2, and d_3. Raw material is coming from two sources: from an external provider (denoted by E in Figure 2.3) over which we have no control, and from a source (denoted by S in Figure 2.3) for which we can completely control the release times (e.g., a storage unit with a large capacity so that its stock level never runs down to zero). The time instants at which the k-th batch of raw material from the controllable source and the external source arrives at the system are denoted by $u(k)$ and $v(k)$, respectively. The raw material from both sources can be processed by either M_1 or M_2, which perform similar tasks. However, M_2 is slower than M_1. Therefore, the part of the raw material that arrives first for the k-th product is processed on M_2 and the part that arrives last will be processed on M_1. So if $v(k) \leq u(k)$, then the k-th batch of raw material coming from the external (uncontrollable) source is processed on M_2 and the raw material coming from the controllable source on M_1; if $u(k) < v(k)$, the reverse holds. This implies that the k-th batch of raw material destined for M_1 arrives at the production unit at time instant $t = \max(u(k), v(k))$, and that the k-th batch destined for M_2 arrives at time instant $t = \min(u(k), v(k))$. The intermediate products generated by M_1 and M_2 are sent to M_3 where assembly takes places.

The processing time for M_2 and M_3 are $d_2 = 15$ and $d_3 = 1$ time units, respectively. We assume that the processing time for M_1 is perturbed by noise, i.e., $d_1(k) = 10 + e(k)$ where $e(k)$ is an stochastic variable with a given probability distribution. We assume that the transportation times in the manufacturing system are negligible, and that in between the production units there are storage buffer with a sufficiently large capacity, so that no buffer overflows occur. The time instant at which processing unit M_i starts processing the k-th batch is denoted by $x_i(k)$, and

$y(k)$ is the time instant at which the k-th finished product leaves the system. Assume that each production unit starts working for the k-th time as soon as the raw material is available and as soon as the production unit has finished processing the previous part. Hence,

$$\begin{aligned}
x_1(k) &= \max\bigl(x_1(k-1) + d_1(k), \max(u(k), v(k))\bigr)\\
&= \max\bigl(x_1(k-1) + d_1(k), u(k), v(k)\bigr)\\
x_2(k) &= \max\bigl(x_2(k-1) + d_2, \min(u(k), v(k))\bigr)\\
x_3(k) &= \max\bigl(x_3(k-1) + d_3, x_1(k) + d_1(k), x_2(k) + d_2\bigr)\\
&= \max\bigl(x_3(k-1) + d_3, x_1(k-1) + 2d_1(k), u(k) + d_1(k),\\
&\qquad v(k) + d_1(k), x_2(k-1) + 2d_2, \min(u(k), v(k)) + d_2\bigr)\\
y(k) &= x_3(k) + d_3.
\end{aligned}$$

Note that $x_i(k)$ and $y(k)$ are MMPS functions of $x(k-1), u(k), v(k)$, and $e(k)$. □

2.5 Summary

In this chapter we have briefly discussed max-plus algebra and three different classes of discrete-event systems, namely the class of stochastic MPL systems, the class of stochastic switching MPL systems, and the class of stochastic MMPS systems. For each of these classes, the state space representation of each system has been given and some related issues such as the mode switching probability description and the equivalence between MMPS and PWA functions have been addressed. Furthermore, for each class one example has been provided.

Chapter 3

Model Predictive Control and Identification of Stochastic Max-Plus-Linear Systems

In this chapter, state-of-the-art in control and identification of stochastic MPL systems is discussed. More specifically, we first discuss model predictive control (MPC) for stochastic MPL systems and next, we also consider MPC for two other classes of discrete-event systems, namely the class of stochastic switching MPL systems and the class of stochastic MMPS systems. For both MPC and identification of stochastic MPL systems, some solution approaches have been proposed in the literature, which are presented in this chapter. These approaches have some limitations, which will be discussed here as well.

3.1 Model Predictive Control (MPC) for Stochastic MPL Systems

MPC is an advanced control approach used in the process industry that relies on a dynamic model of the process and it has the following properties:

- MPC is a model-based controller design procedure that can easily handle processes with large time delays, multi-input multi-output processes, non-minimum phase processes, and unstable processes.

- It is an easy-to-tune method: in principle only three parameters have to be tuned.

- MPC can handle constraints on the inputs and the outputs of the process (due to, e.g., limited capacity of buffers, actuator saturation, output quality specifications, etc.) in a systematic way during the design and the implementation of the controller. Conventional control design techniques, such as LQG and H$_\infty$ control methods [3, 101], can often not be applied once additional constraints on inputs and outputs are included.

- MPC can handle structural changes, such as sensor or actuator failures, and changes in system parameters or the system structure, by adapting the model and by using a receding horizon approach, in which the model and the control strategy are regularly updated.

MPC is based on an iterative, finite horizon algorithm to obtain an optimal control sequence in order to minimize the objective function subject to the given constraints. At each iteration, the optimal control sequence is computed over a finite horizon, i.e., a finite period of time or a finite series of events. MPC uses the receding horizon principle, which means that after computation of the optimal control sequence, only the first sample will be implemented in the next iteration. Subsequently, the horizon will be shifted one sample, and the optimization will be restarted with new information of the measurements. The MPC methodology is explained in more details in the following section.

Conventional MPC uses linear or nonlinear discrete-time models [74]. However, MPC has also been extended to discrete-event systems due to its useful features mentioned above [27, 82]. One of the relevant topics that has attracted much attention recently, is the application of MPC for perturbed max-plus linear systems in which modeling errors, noise, and/or disturbances are present. In [104] such systems have been studied, which results in an MPC controller for perturbed MPL systems.

3.1.1 Problem Statement

In [27, 104] the MPC framework has been extended to MPL models (2.2)–(2.3) as follows. Following the conventional MPC methodology [44, 74], we define an objective function J that reflects the input and output objective functions from event step k to $k + N_\mathrm{p} - 1$, as

$$J(k) = J_\mathrm{out}(k) + \lambda J_\mathrm{in}(k) \tag{3.1}$$

where N_p is the prediction horizon and λ is a weighting factor. Typical choices of J_out and J_in for a stochastic system are:

$$\begin{aligned} J_{\mathrm{out},1}(k) &= \mathbb{E}[\|\tilde{y}(k) - \tilde{r}(k)\|_1], \quad J_{\mathrm{out},\infty}(k) = \mathbb{E}[\|\tilde{y}(k) - \tilde{r}(k)\|_\infty], \\ J_{\mathrm{out,t}}(k) &= \mathbb{E}[\max\{\tilde{y}(k) - \tilde{r}(k), 0\}] \\ J_{\mathrm{in},1}(k) &= -\|\tilde{u}(k)\|_1, \quad J_{\mathrm{in},\infty}(k) = -\|\tilde{u}(k)\|_\infty, \end{aligned} \tag{3.2}$$

$$J_{\text{in,t}}(k) = -\sum_{i=1}^{N_p \cdot n_u} \tilde{u}_i(k)$$

where $\mathbb{E}[\cdot]$ denotes the expected value operator and

$$\begin{aligned}\tilde{u}(k) &= [\ u^T(k)\ \ \cdots\ \ u^T(k+N_\text{p}-1)\ ]^T,\\ \tilde{y}(k) &= [\ y^T(k)\ \ \cdots\ \ y^T(k+N_\text{p}-1)\ ]^T,\\ \tilde{r}(k) &= [\ r^T(k)\ \ \cdots\ \ r^T(k+N_\text{p}-1)\ ]^T,\end{aligned} \tag{3.3}$$

with $r(k)$ denoting the vector of reference (due date) signals. To proceed further, as an example, let us consider[1] the following input and output objective functions:

$$J_\text{out}(k) = \sum_{j=0}^{N_\text{p}-1} \sum_{i=1}^{n_y} \mathbb{E}[\max(y_i(k+j) - r_i(k+j), 0)]\ , \tag{3.4}$$

$$J_\text{in}(k) = -\sum_{j=0}^{N_\text{p}-1} \sum_{l=1}^{n_u} u_l(k+j). \tag{3.5}$$

The physical interpretation for this choice of output and input objective functions is that $J_\text{out}(k)$ penalizes the late but not early deliveries for the i-th output at event step k and $J_\text{in}(k)$ promotes feeding as late as possible at each event step k. Hence, the objective function (3.1) can be defined as:

$$J(k) = \sum_{j=0}^{N_\text{p}-1} \sum_{i=1}^{n_y} \mathbb{E}[\max(y_i(k+j) - r_i(k+j), 0)] - \lambda \sum_{j=0}^{N_\text{p}-1} \sum_{l=1}^{n_u} u_l(k+j) \tag{3.6}$$

The aim is to compute an optimal input sequence $u(k), \ldots, u(k+N_\text{p}-1)$ that minimizes $J(k)$ in (3.6) subject to linear constraints on the inputs and the expected value of the outputs, i.e., the constraints of the form

$$A_\text{con}(k)\tilde{u}(k) + B_\text{con}(k)\mathbb{E}[\tilde{y}(k)] \leq c_\text{con}(k)$$

where $A_\text{con}(k) \in \mathbb{R}^{n_\text{con} \times N_\text{p} n_u}$, $B_\text{con}(k) \in \mathbb{R}^{n_\text{con} \times N_\text{p} n_y}$, and $c_\text{con} \in \mathbb{R}^{n_\text{con}}$ with n_con denoting the number of constraints.

Moreover, since the $u(k)$'s correspond to consecutive event occurrence times, we have the additional condition

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

[1] Note that the whole procedure presented here can also be applied to any other combination of the input and output objective functions in (3.2).

Furthermore, in order to reduce the number of decision variables and the corresponding computational complexity, we can introduce a control horizon N_{c} ($\leq N_{\mathrm{p}}$) and add the condition that the input rate should be constant from the point $k+N_{\mathrm{c}}-1$ on, i.e.,

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

or equivalently

$$\Delta^2 u(k+j) = \Delta u(k+j) - \Delta u(k+j-1) = 0 \quad \text{for} \;\; j = N_{\mathrm{c}}, \ldots, N_{\mathrm{p}}-1.$$

MPC uses a receding horizon principle, which means that after computation of the optimal control sequence $u(k), \ldots, u(k+N_{\mathrm{c}}-1)$, only the first control sample $u(k)$ will be implemented, subsequently the horizon is shifted one event step, and the optimization is restarted with new information of the measurements.

Now, by using successive substitution in (2.2)-(2.3), we obtain [104]

$$\tilde{y}(k) = \tilde{C}(k) \otimes x(k-1) \oplus \tilde{D}(k) \otimes \tilde{u}(k) \tag{3.7}$$

in which $\tilde{e}(k) = [\;e^T(k) \;\; \cdots \;\; e^T(k+N_{\mathrm{p}}-1)\;]^T$ and $\tilde{C}(k)$ and $\tilde{D}(k)$ are given by

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

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

where

$$\tilde{C}_m(k) = C(k+m-1) \otimes A(k+m-1) \otimes \ldots \otimes A(k)$$

$$\tilde{D}_{mn}(k) = \begin{cases} C(k+m-1) \otimes A(k+m-1) \otimes \ldots \otimes A(k+n) \otimes B(k+n-1) & m > n \\ C(k+m-1) \otimes B(k+m-1) & m = n \\ \varepsilon & m < n. \end{cases}$$

Note that since the entries of $A(k),\; B(k)$, and $C(k)$ belong to $\mathcal{S}_{\mathrm{mps}}$ (cf. Section 2.2), the entries of $\tilde{C}(k)$ and $\tilde{D}(k)$ belong to $\mathcal{S}_{\mathrm{mps}}$ as well.

Hence, we can rewrite (3.4)-(3.5), and accordingly (3.6), as

$$J_{\mathrm{out}}(k) = \sum_{i=1}^{N_{\mathrm{p}} \cdot n_y} \mathbb{E}\Big[\max\Big(\tilde{y}_i(k) - \tilde{r}_i(k),\; 0 \Big) \Big]$$

$$J_{\mathrm{in}}(k) = -\sum_{j=1}^{N_{\mathrm{p}}\cdot n_u} \tilde{u}_j(k)$$
$$J(k) = J_{\mathrm{out}}(k) + \lambda J_{\mathrm{in}}(k)$$

where $\tilde{y}_i(k)$ and $\tilde{r}_i(k)$ denote the i-th element of $\tilde{y}(k)$ and $\tilde{r}(k)$ respectively, and $\tilde{u}_j(k)$ denotes the j-th element of $\tilde{u}(k)$.

The stochastic MPL-MPC problem for event step k is then defined as follows [104]:

$$\begin{aligned} &\min_{\tilde{u}(k)} \quad J(k) \\ \text{s.t.} \quad &\tilde{y}(k) = \tilde{C}(\tilde{e}(k)) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}(k)) \otimes \tilde{u}(k) \\ &\Delta u(k+j) \geq 0 \quad \text{for } j = 0, \dots, N_{\mathrm{p}} - 1 \\ &\Delta^2 u(k+j) = 0 \quad \text{for } j = N_{\mathrm{c}}, \dots, N_{\mathrm{p}} - 1 \\ &A_{\mathrm{con}}(k)\tilde{u}(k) + B_{\mathrm{con}}(k)\mathbb{E}[\tilde{y}(k)] \leq c_{\mathrm{con}}(k). \end{aligned} \tag{3.8}$$

In the next section, we present a possible solution approach to solve (3.8).

3.1.2 Solution Approach

To solve the optimization problem (3.8), we need to compute the expected value of the signals $\max(\tilde{y}_j(k) - \tilde{r}_j(k), 0)$ and $\tilde{y}(k)$. As shown in [104, Lemma 2], both $\max(\tilde{y}_j(k) - \tilde{r}_j(k), 0)$ and $\tilde{y}(k)$ belong to $\mathcal{S}_{\mathrm{mps},[\tilde{r}^T(k),\, x^T(k-1),\, \tilde{u}^T(k),\, \tilde{e}^T(k)]^T}$.

We can rewrite both $\max(\tilde{y}_j(k) - \tilde{r}_j(k), 0)$ and $\tilde{y}(k)$ as a general max-plus-scaling function of $\tilde{u}(k)$ and $\tilde{e}(k)$ [104], which is denoted by $v(k)$, as follows:

$$v(k) = \max_{j=1,\dots,n_v} (\xi_j + \delta_j^T x(k-1) + \psi_j^T \tilde{r}(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}(k))$$

where n_v is the number of terms that appear in the maximization, $\xi_j \in \mathbb{R}_\varepsilon$, $\delta_j \in \mathbb{R}^{n_x}$, $\psi_j \in \mathbb{R}^{n_r}$, $\beta_j \in \mathbb{R}^{n_u}$, $\gamma_j \in \mathbb{R}^{n_{\tilde{e}}}$, and $\tilde{e}(k) \in \mathbb{R}^{n_{\tilde{e}}}$ is a stochastic variable with the probability density function $f(\tilde{e})$. For a shorter notation let $\alpha_j(k) = \xi_j + \delta_j^T x(k-1) + \psi_j^T \tilde{r}(k)$; hence,

$$v(k) = \max_{j=1,\dots,n_v} (\alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}(k)) \tag{3.9}$$

Accordingly, we need now to compute the expected value of $v(k)$, i.e., $\mathbb{E}[v(k)]$. By definition [89], $\mathbb{E}[v(k)]$ can be written as follows:

$$\begin{aligned} \mathbb{E}[v(k)] &= \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} v(k) f(\tilde{e}) d\tilde{e} \\ &= \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \max_{j=1,\dots,n_v} (\alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}(k)) f(\tilde{e}) d\tilde{e} \end{aligned}$$

$$= \sum_{j=1}^{n_v} \int_{\tilde{e} \in \Phi_j(\tilde{u}(k))} \cdots \int (\alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}) f(\tilde{e}) d\tilde{e} \tag{3.10}$$

where $d\tilde{e} = d\tilde{e}_1 d\tilde{e}_2 \ldots d\tilde{e}_{n_{\tilde{e}}}$ and the sets $\Phi_j(\tilde{u}(k))$ constitute a partition of $\mathbb{R}^{n_{\tilde{e}}}$ such that

$$\mathrm{int}(\Phi_\ell) \cap \mathrm{int}(\Phi_\nu) = \emptyset \;\text{ for }\; \ell \neq \nu$$

where $\mathrm{int}(\Phi_j)$ denotes the interior of Φ_j, and such that for $j = 1, \ldots, n_v$ we have

$$v(k) = \alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}(k) \quad \text{for all } \tilde{e} \in \Phi_j(\tilde{u}(k))$$

and $\bigcup_{j=1}^{n_v} \Phi_j(\tilde{u}(k)) = \mathbb{R}^{n_{\tilde{e}}}$, i.e., for all realizations of $\tilde{e}$, the j-th term in (3.9) gives the maximum, and the sets $\Phi_j(\tilde{u}(k))$ cover the whole space of $\mathbb{R}^{n_{\tilde{e}}}$ and only overlap at the boundaries of the regions.

Remark 3.1.1 Note that the sets $\Phi_j,\; j = 1, \ldots, n_v$ are polyhedra. This follows from the fact that Φ_j is described by a system of linear inequalities. Indeed, we have:

$$\Phi_j(\tilde{u}(k)) = \{\tilde{e} \mid \max_{\ell=1,\ldots,n_v} (\alpha_\ell(k) + \beta_\ell^T \tilde{u}(k) + \gamma_\ell^T \tilde{e}) = \alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}\},$$

or equivalently, for $\ell = 1, \ldots, n_v$,

$$\Phi_j(\tilde{u}(k)) = \{\tilde{e} \mid \alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e} \geq \alpha_\ell(k) + \beta_\ell^T \tilde{u}(k) + \gamma_\ell^T \tilde{e}\}.$$

Therefore, Φ_j is described by a system of linear inequalities and hence, is a polyhedron. □

Hence, $\mathbb{E}[\tilde{y}(k)]$ and $\mathbb{E}[\max(\tilde{y}_j(k) - \tilde{r}_j(k), 0)]$ can be computed using (3.10).

It is also shown in [104, Proposition 3], that the function $\mathbb{E}[v(k)]$ is convex in $\tilde{u}(k)$ and its subgradient $g_v(\tilde{u}(k))$ is given by

$$g_v(\tilde{u}(k)) = \sum_{j=1}^{n_v} \beta_j \int_{\tilde{e} \in \Phi_j(\tilde{u}(k))} \cdots \int f(\tilde{e}) d\tilde{e}. \tag{3.11}$$

Note that since the system matrices are perturbed by $\tilde{e}(k)$, $\tilde{y}(k)$ and $\max(\tilde{y}_j(k) - \tilde{r}_j(k), 0)$ both depend on $\tilde{e}(k)$. Furthermore, $\mathbb{E}[\tilde{y}(k)]$ and $\mathbb{E}[\max(\tilde{y}_j(k) - \tilde{r}_j(k), 0)]$ are convex in $\tilde{u}(k)$, due to [104, Lemma 3], which implies that $J_{\text{out}}(k)$ and accordingly, $J(k)$ are convex in $\tilde{u}(k)$.

Remark 3.1.2 It is assumed that the reference signal, $\tilde{r}(k)$, is fixed and known at event step k. The state, $x(k)$, depends on its previous value $x(k-1)$, which is assumed to be completely known at event step k (cf. Section 3.1.4), and on the optimal input $\tilde{u}(k)$; hence $x(k)$ changes due to the change of $\tilde{u}(k)$. Therefore, we only consider the variations of $\tilde{u}(k)$ and accordingly, the convexity of all functions in $\tilde{u}(k)$. □

Note that if the entries of $B_{\mathrm{con}}(k)$ in (3.8) are nonnegative, we will obtain monotonically nondecreasing constraints in $\mathbb{E}[\tilde{y}(k)]$. Hence, since $\mathbb{E}[\tilde{y}(k)]$ is convex in $\tilde{u}(k)$, all the constraints in (3.8) are then convex in $\tilde{u}(k)$. Therefore, only when all entries of $B_{\mathrm{con}}(k)$ are nonnegative, the MPL-MPC optimization problem turns out to be a convex problem in $\tilde{u}(k)$ [104, Property 4]. Such a problem can be solved using reliable and efficient convex optimization algorithms, such as interior point methods [83, 118].

3.1.3 Computational Aspects and Issues

One way of computing $\mathbb{E}[v(k)]$ in (3.10) is to use numerical integration. The common methods for numerical integration are (non)adaptive integration, (non)iterative integration, exponential quadrature, Monte Carlo integration, the Nyström method, the Quasi-Monte Carlo method, and the multi-step method [25]. However, numerical integration is in general both cumbersome and time-consuming, and it becomes even more complicated as the probability density function f becomes more and more complex.

In [104], an alternative method for computing $\mathbb{E}[v(k)]$ is proposed based on analytic integration. To that end, a piecewise polynomial probability density function defined on polyhedral sets is considered. Such a function can be obtained in two ways: either the stochastic vector already has a piecewise polynomial probability density function (such as the uniform distribution) or we approximate the real probability density function with a piecewise polynomial probability density function[2] (such as the normal distribution where its probability density function can be approximated by PWA functions).

Let $f(\tilde{e})$ be a piecewise polynomial function defined on polyhedral sets P_ℓ, $\ell = 1, \ldots, n_p$, such that

$$\bigcup_{\ell=1}^{n_p} P_\ell = \mathbb{R}^{n_e}$$

$$\mathrm{int}(P_i) \cap \mathrm{int}(P_j) = \emptyset \ \text{ for } \ i \neq j$$

[2] The approximate probability density function must be nonnegative and its integral over the domain of the real probability density function must be equal to 1. This can be assured by including these conditions as constraints in the parameter fitting.

where $\mathrm{int}(P_i)$ denotes the interior of P_i, and for $e \in P_\ell$ the probability density function is given by $f_\ell(e)$, where

$$f_\ell(e) = \sum_{i_1=0}^{M_1} \sum_{i_2=0}^{M_2} \cdots \sum_{i_{n_e}=0}^{M_{n_e}} \zeta_{i_1,i_2,\ldots,i_{n_e}} e_1^{i_1} e_2^{i_2} \cdots e_{n_e}^{i_{n_e}}$$

for some integers $M_1, \ldots, M_{n_e}$ and coefficients $\zeta_{i_1,i_2,\ldots,i_{n_e}} \in \mathbb{R}$.

Remark 3.1.3 Later in Chapter 6, Section 6.1.2, we explain that we can also approximate a piecewise polynomial probability density function by a multi-variable piecewise polynomial function, possibly multiplied by an exponential function. It is also discussed in that chapter that spline functions or phase-type distributions can also be approximated by a multi-variable piecewise polynomial function, possibly multiplied by an exponential function. □

Consider the signal $v(k) \in \mathcal{S}_{\mathrm{mps}}(z(k))$ and let $\tilde{u}(k)$ be its non-stochastic part. Let $\Psi_{j\ell}(\tilde{u}(k)) = \Phi_j(\tilde{u}(k)) \cap P_\ell$ for $j = 1, \ldots, n_v$, $\ell = 1, \ldots, n_p$. Then by Remark 3.1.1, $\Psi_{j\ell}(\tilde{u}(k))$ is a polyhedron, and $\mathbb{E}[v(k)]$ can be written as

$$\mathbb{E}[v(k)] = \sum_{\ell=1}^{n_p} \sum_{j=1}^{n_v} \int \cdots \int_{\tilde{e} \in \Psi_{j\ell}(\tilde{u}(k))} (\alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}) f_\ell(\tilde{e}) d\tilde{e}. \tag{3.12}$$

This is a sum of integrals of polynomial functions in $\tilde{e}$ and then can be solved analytically for each polyhedron $\Psi_{ij\ell}$ [17, 68]. Note that if a piecewise polynomial probability density function is used as an approximation of the "true" non-polynomial probability density function, the quality of the approximation can be improved by increasing the number of sets n_p.

Even if the integral in (3.12) can be computed analytically, the computational load is still quite heavy. This is due to the fact that this method contains two time-consuming steps: In the first step all polyhedra $\Psi_{j\ell}$ have to be specified, where the number of polyhedra Φ_j is equal to n_v and the number of polyhedra P_ℓ is n_p. Hence, in the worst case the number of polyhedra $\Psi_{j\ell}$ that has to be considered is $O(n_v n_p)$, which becomes more and more time-consuming as n_p and n_v become larger. In the second step, the integral over each of these regions has to be calculated, for which in the simplest case of having a uniform probability density function, we need to compute all the vertices of each polyhedron $\Psi_{j\ell}$. As explained in [76], we have the following upper bound for the number of the vertices of a polytope defined by n_v (non-redundant) inequality constraints in an $n_{\tilde{e}}$-dimensional space:

$$\mathcal{V}(n_v, n_{\tilde{e}}) = \binom{n_v - \lfloor \frac{n_{\tilde{e}}+1}{2} \rfloor}{n_v - n_{\tilde{e}}} + \binom{n_v - \lfloor \frac{n_{\tilde{e}}+2}{2} \rfloor}{n_v - n_{\tilde{e}}} \tag{3.13}$$

This means that in our case, where typically n_v is much larger than $n_{\tilde{e}}$, i.e., $n_v \gg n_{\tilde{e}} \gg 1$, the number of vertices for the worst case can be $O(n_v^{\lfloor \frac{n_{\tilde{e}}}{2} \rfloor})$, which is again time-consuming as n_v and $n_{\tilde{e}}$ increase. Accordingly for the case of a uniformly distributed noise, the complexity of the whole procedure in the worst case is of order $O(n_p n_v^{\lfloor \frac{n_{\tilde{e}}}{2} \rfloor + 1})$. In the case of other piecewise polynomial probability density functions, the order of complexity of the second step becomes even bigger since then, the integral computation is more complex than the one in the case of a uniform distribution. Therefore, the computational complexity of this method increases exponentially as $n_{\tilde{e}}$ increases and polynomially as n_v increases. It increases even more in the case of non-piecewise polynomial probability density functions, such as a normal probability density function, since these functions cannot be directly applied for this method and have to be approximated by piecewise polynomial probability density functions.

In [110] an effort is made to reduce the computational complexity of the above-mentioned method by approximating $\mathbb{E}[v(k)]$ in (3.10) using the method of variability expansion. Since variability expansion is an analytic method and does not resort to simulation, it is, in principle, possible to compute higher-order moments of performance characteristics of stochastic systems. As presented in [110, Section 4], it is assumed that the entries of $\tilde{e}(k)$ are independent and identically distributed (i.i.d) and an artificial parameter $\theta \in [0, 1]$ is introduced. The i-th entry of $\tilde{e}(k)$ is then replaced by its mean with probability $1 - \theta$ and the result is denoted by $\tilde{e}_\theta(k)$. The parameter θ allows controlling the level of randomness in the system, and letting θ go from 0 to 1 increases the level of stochasticity in the system. The main idea of variability expansion is as follows: considering $\mathbb{E}[v_\theta(k)]$ as a function of θ, it can be developed into a Taylor series in θ that converges to the true function on some subset $\mathcal{X} \subseteq \mathbb{R}$. In particular, if the value of $d^m/d\theta^m \mathbb{E}[v_\theta(k)]$ for $\theta = 0$ is denoted by $d^m/d\theta^m \mathbb{E}[v_0(k)]$, then $\mathbb{E}[v_1(k)]$, the “true” expected value of $v(k)$ is given by

$$\mathbb{E}[v(k)] = \mathbb{E}[v_1(k)] = \sum_{m=0}^{M} \frac{1}{m!} \frac{d^m}{d\theta^m} \mathbb{E}[v_0(k)] + R_M(k) \tag{3.14}$$

where for $M < n_{\tilde{e}}$

$$R_M(k) \leq \frac{1}{(M+1)!} \sup_{\theta \in [0,1]} \left| \frac{d^{M+1}}{d\theta^{M+1}} \mathbb{E}[v_0(k)] \right|$$

and $R_M = 0$ otherwise [56]. It is been also shown in [110] that a closed-form expression for the m-th order derivative $d^m/d\theta^m \mathbb{E}[v_0(k)]$ can be obtained.

The computational complexity of approximating $\mathbb{E}[v(k)]$ using the method of variability expansion has been discussed in [110, Section 5]. Based on this discussion, the overall complexity will at least be of order

$$O\Big(\binom{n_{\tilde{e}}}{M} n_v \mathcal{V}(2M + n_v - 1, M) \Big) = O\Big(\frac{n_{\tilde{e}}^M \; n_v^{\frac{M+1}{2}}}{M! \; (\frac{M-1}{2})!} \Big)$$

where $\mathcal{V}(\cdot,\cdot)$ is given in (3.13). The derivation of the above error order can be found in Appendix A. Clearly, the computational complexity increases polynomially if n_e and n_v increase and exponentially if M increases.

3.1.4 Timing

When solving the MPC problem for event step k, we assume that the state $x(k-1)$ is available (recall that $x(k-1)$ contains the time instants at which the internal activities or processes of the system start for the $(k-1)$-th cycle). However, MPL systems are different from conventional time-driven systems in the sense that the event counter k is not directly related to a specific time instant. Therefore, we will present a method to address the availability issue of the state at a certain time instant t [103, 108]. Since the components of $x(k-1)$ correspond to event times, they are in general easy to measure. So we consider the case of full state information. Also note that measurements of occurrence times of events are in general not as susceptible to noise and measurement errors as measurements of continuous-time signals involving variables such as temperature, speed, pressure, etc. Let t be the time instant when an MPC problem has to be solved. We can define the initial cycle k as follows:

$$k = \arg\max\Big\{ \ell \mid x_i(\ell - 1) \leq t, \; \forall i \in \{1, \ldots, n_x\} \Big\}$$

Hence, the state $x(k-1)$ is completely known at time t and thus $u(k-1)$ is also available (due to the fact that in practice, the entries of the system matrices are nonnegative or take the value ε and $[A(k) \;\; B(k)]$ will be row finite, i.e., has no row consisting of ε entries only). Note that at time t some components of the future[3] states and of the forthcoming inputs might be known (so $x_i(k+\ell) \leq t$ and $u_j(k+\ell) \leq t$ for some i, j and some $\ell > 0$). During the optimization at time instant t the known values of the inputs and states have to be fixed by equality constraints (see Remark 3.1.4 below), which fits perfectly in the framework of a linear programming problem. With these new equality constraints we can perform the MPC optimization at time t.

Remark 3.1.4 Consider a given time instant t and define[4]

$$\mathcal{K}_x = \{(i, \ell) | x_i(k + \ell) \text{ is known at time } t\},$$

[3] Future in the event counter sense.

[4] For the sake of simplicity of notation, we do not add t as an argument here.

$$\begin{aligned}
\mathcal{K}_u &= \{(j,\ell)|u_j(k+\ell) \text{ is known at time } t\},\\
\mathcal{U}_x &= \{(i,\ell)|x_i(k+\ell) \text{ is unknown at time } t\},\\
\mathcal{U}_u &= \{(j,\ell)|u_j(k+\ell) \text{ is unknown at time } t\}.
\end{aligned}$$

First, we require that $u_j(k+\ell)$ is equal to its known value for each $(j,\ell) \in \mathcal{K}_u$. Next, for every $(i,\ell) \in \mathcal{K}_x$, the state at time t can be written as follows:

$$\begin{aligned}
x_i(k+\ell) = \max\Big(&\max_{(i',\ell')\in\mathcal{K}_{x,i,\ell}} \big(x_{i'}(k+\ell') + a_{i'\ell'}\big),\\
&\max_{(i'',\ell'')\in\mathcal{U}_{x,i,\ell}} \big(x_{i''}(k+\ell'') + a_{i''\ell''}\big),\\
&\max_{(j',l')\in\mathcal{K}_{u,j,\ell}} \big(u_{j'}(k+l') + b_{j'l'}\big),\\
&\max_{(j'',l'')\in\mathcal{U}_{u,j,\ell}} \big(u_{j''}(k+l'') + b_{j''l''}\big)\Big)
\end{aligned}$$

where $a_{i'\ell',i''\ell''}, b_{j'l'}, b_{j''l''} \geq 0$ and $\mathcal{K}_{x,i,\ell}$ is a subset of pair $(i',\ell') \in \mathcal{K}_x$ that directly determines $x_i(k+\ell)$ (i.e., $a_{i'\ell'} \neq \varepsilon$ in the equation defines $x_i(k+\ell)$). The sets $\mathcal{U}_{x,i,\ell}, \mathcal{K}_{u,j,\ell}$, and $\mathcal{U}_{u,j,\ell}$ are defined in a similar way. Let us now show that in fact $\mathcal{U}_{x,i,\ell} = \mathcal{U}_{u,j,\ell} = \emptyset$. Indeed, at time t, we should have

$$x_i(k+\ell) \geq x_{i''}(k+\ell'') + a_{i''\ell''} \quad \forall (i'',\ell'') \in \mathcal{U}_{x,i,\ell}. \tag{3.15}$$

However, we have $x_{i''}(k+\ell'') > t$ since it is not known at time t, which yields a contradiction in the above inequality since $t \geq x_i(k+\ell)$ and $a_{i''\ell''} \geq 0$. Hence, $\mathcal{U}_{x,i,\ell} = \emptyset$. The same argument is valid for $\mathcal{U}_{u,j,\ell}$. However, if we write down an inequality of the form (3.15) for pairs in the sets $\mathcal{K}_{x,i,\ell}$ and $\mathcal{K}_{u,j,\ell}$, then we obtain valid inequalities with both known right-hand and left-hand sides. Therefore, for all $(i,\ell) \in \mathcal{K}_x$, no extra equation is needed due to causality. Hence, we only need to impose equality constraints on the known input and known state components in $\mathcal{K}_u$ and $\mathcal{K}_x$.

□

Another timing issue is related to the stochastic MPL systems. In these systems we again have the same timing problem as explained above as well as a timing issue related to the distribution of stochastic variables. If an event has not yet occurred at time t, the probability density function of stochastic variables involved in that event have to be updated, which may change the nature of the distribution. This may prevent the use of analytic expressions for moments, which are needed in the approximation methods used in Chapters $4-7$.

Therefore, in this thesis, the whole timing issue related to MPC for stochastic MPL systems is not considered due to the complexity it imposes to the problem definition specially when updated distributions has to be taken into account since, we cannot then apply the proposed approximation method of Chapter 4.

3.1.5 Extension to Stochastic Switching MPL and Stochastic MMPS systems

In [106–108], MPC has been applied and translated to switching MPL systems. In stochastic switching MPL systems with the system equations (2.11)-(2.12), for each mode $\ell \in \{1, \ldots, n_L\}$ and the vector $\tilde{\ell}(k) = [\ell(k), \ldots, \ell(k + N_\mathrm{p}1)]^T$, we can rewrite (3.7) as follows:

$$\tilde{y}(k) = \tilde{C}(\tilde{\ell}(k), \tilde{e}(k)) \otimes x(k-1) \oplus \tilde{D}(\tilde{\ell}(k), \tilde{e}(k)) \otimes \tilde{u}(k)$$

where $\tilde{C}(\tilde{\ell}(k), \tilde{e}(k))$ and $\tilde{D}(\tilde{\ell}(k), \tilde{e}(k))$ are defined as:

$$\tilde{C}(\tilde{\ell}(k), \tilde{e}(k)) = \begin{bmatrix} \tilde{C}_{(1)}(\tilde{\ell}(k), \tilde{e}(k)) \\ \vdots \\ \tilde{C}_{(N_\mathrm{p})}(\tilde{\ell}(k), \tilde{e}(k)) \end{bmatrix}$$

$$\tilde{D}(\tilde{\ell}(k), \tilde{e}(k)) = \begin{bmatrix} \tilde{D}_{(1,1)}(\tilde{\ell}(k), \tilde{e}(k)) & \cdots & \mathcal{E} \\ \vdots & \ddots & \vdots \\ \tilde{D}_{(N_\mathrm{p},1)}(\tilde{\ell}(k), \tilde{e}(k)) & \cdots & \tilde{D}_{(N_\mathrm{p},N_\mathrm{p})}(\tilde{\ell}(k), \tilde{e}(k)) \end{bmatrix}$$

with:

$$\begin{aligned} \tilde{C}_{(m)}(\tilde{\ell}(k), \tilde{e}(k)) &= C_{\ell(k+m-1)}(e(k+m-1)) \otimes A_{\ell(k+m-1)}(e(k+m-1)) \otimes \\ &\quad \ldots \otimes A_{\ell(k)}(e(k)) \\ \tilde{D}_{(m,n)}(\tilde{\ell}(k), \tilde{e}(k)) &= C_{\ell(k+j-1)}(e(k+m-1)) \otimes A_{\ell(k+m-1)}(e(k+j-1)) \otimes \\ &\quad \ldots \otimes A_{\ell(k+n)}(e(k+n)) \otimes B_{\ell(k+n-1)}(e(k+n-1)) \end{aligned}$$

Accordingly, the input and output objective functions can be defined as the ones in (3.4)-(3.5) and then a similar optimization problem as the one in (3.8) has to be solved.

In [105] a stabilizing model predictive controller has been derived for randomly switching MPL systems with deterministic parameters. The resulting optimization problem was solved using linear programming algorithms. The main drawback of the algorithm is that the number of linear constraints and the number of optimization variables in the linear programming problem increases fast with the prediction horizon and the number of modes in the system. In [106] an algorithm is used based on scenario generation in order to reduce the total number of mode switching sequences. The main idea of this algorithm is that only the most probable mode switching sequences will be investigated since the least probable ones have a negligible impact on the outcome of the objective function, and therefore can be neglected.

In the case of deterministic mode switching and stochastic parameters, for each mode, the system equations are of the form (2.2)-(2.3) and the same objective functions, i.e., (3.4)-(3.5) can be considered. Hence, to solve the MPL-MPC optimization problem (3.8) at each mode, the methods discussed in Section 3.1.3 can be applied.

MPC has been also applied to MMPS systems [29, 81]. In [29] a deterministic MMPS system with the system equations (2.13)-(2.14) is studied, which leads to a non-linear optimization problem that can be solved using standard algorithms such as multi-start sequential quadratic programming (SQP) [80]. In stochastic MMPS systems with the system equations (2.15)-(2.16), the input and output objective functions can be chosen among the ones in (3.2).

Remark 3.1.5 Any combination of the input and output objective functions in (3.2) can be applied to event-driven MMPS systems. Note that due to the minus sign in the input objective functions, these functions express just-in-time feeding. In the case of time-driven MMPS systems when usually the input energy is minimized, the input objective function in (3.2) will not have a minus sign. □

Note that since both $x(k)$ and $y(k)$ are MMPS functions of $x(k-1), u(k), \ldots, u(k+j), e(k), \ldots, e(k+j)$, we conclude that all objective functions in (3.2) are also MMPS functions of $\tilde{u}(k), x(k-1), \tilde{e}(k)$, and $\tilde{r}(k)$. Therefore, the stochastic MPC-MMPS problem can be defined as follows:

$$\begin{aligned} &\min_{\tilde{u}(k)} \quad \tilde{J}(k) \\ &\text{subject to:} \quad c(\tilde{u}(k), \tilde{y}(k), k) \leq 0. \end{aligned} \tag{3.16}$$

for some function $c(\cdot)$ with $\tilde{J}(k)$ an MMPS function of $\tilde{u}(k), x(k-1), \tilde{e}(k)$, and $\tilde{r}(k)$.

In [81] an uncertain MMPS system is considered in which the error vector is assumed to be in a bounded polyhedral set, and hence, it was possible to design a worst-case MMPS-MPC controller based on two different approaches, namely optimization over open-loop input sequences and optimization over disturbance feedback policies. It has also been shown in [81] that the resulting optimization problem, i.e., (3.16), can be solved efficiently using a two-step optimization approach that basically involves solving a sequence of linear programming problems. In the open-loop approach, the first step is solving a multi-parametric linear programming (mp-LP) problem off-line and next, obtaining the min-max canonical form of the worst-case performance criterion, which is an MMPS function. In the second step, only a sequence of linear programming problems has to be solved. In the disturbance feedback approach, the first step is to write the min-max canonical form of the worst-case performance criterion, and in the second step a nonlinear optimization problem with linear constraints has to be solved in which the inner optimization problem involves solving a sequence of linear programming problems.

In both stochastic switching MPL systems and stochastic MMPS systems, the computation of the objective function is quite complex and time-consuming using the available methods such as numerical or analytic integration.

3.2 Identification of Stochastic MPL Systems

Another interesting topic is the identification of the model parameters of a stochastic MPL system defined by a state space model. Most identification methods for MPL discrete-event systems use a transfer function approach [11, 43] while state space models have certain advantages: they explicitly take the initial state of the system into account, they can reveal “hidden” behavior such as unobservable, unstable modes, the extension from SISO to MIMO is more intuitive and elegant for state space models, and the analysis is often easier. Some examples of state space models for identification of *deterministic* MPL systems, using either the system’s Markov parameters or minimizing a prediction error based on input-output data and residuation methods, are presented in [26, 28, 96–98]. Since in a stochastic MPL system, the noise and disturbances result in a perturbation of system parameters, in the identification method, the stochastic properties of the systems have to be taken into account.

3.2.1 Problem Statement

For the identification of stochastic MPL systems, consider the following state space representation [4, 23]:

$$\begin{aligned} x(k+1) &= A(k) \otimes x(k) \oplus B(k) \otimes u(k) \\ &= \left[\begin{array}{cc} A(k) & B(k) \end{array}\right] \otimes \left[\begin{array}{c} x(k) \\ u(k) \end{array}\right] \qquad (3.17) \\ &= Q(k) \otimes \phi(k) \qquad (3.18) \end{aligned}$$

where

$$Q(k) = \left[\begin{array}{cc} A(k) & B(k) \end{array}\right] \in \mathbb{R}_{\varepsilon}^{n_x \times m}, \qquad \phi(k) = \left[\begin{array}{c} x(k) \\ u(k) \end{array}\right] \in \mathbb{R}_{\varepsilon}^{m}$$

with $m = n_x + n_u$ where n_x is the number of states and n_u is the number of inputs, $x(k)$ is the state of the system at event step k, and $u(k)$ is the input of the system at event step k. In fact $x(k)$ and $u(k)$ contain the time instants at which the internal and the input event occurs for the k-th time, respectively. We also assume that the entries of the system matrices belong to $\mathcal{S}_{\text{mps}}$ [104], i.e., $A(k) \in \mathcal{S}_{\text{mps},e(k)}^{n_x \times n_x}$, $B(k) \in \mathcal{S}_{\text{mps},e(k)}^{n_x \times n_u}$ where $e(k)$ is an stochastic vector with a given probability distribution and containing all the system uncertainties. In the sequel, we denote the uncertain system matrices with the matrix $Q(k)$ and the state and input vector with $\phi(k)$ (cf. (3.18)).

Remark 3.2.1 Note that the state space model (2.2)–(2.3) that we considered in the previous section relates $x(k)$ to $x(k-1)$, while in (3.18), $x(k+1)$ is defined using $x(k)$. This difference in modeling is due to the fact that the latter model is easier in notation for modeling the identification problems since in $\phi(k)$, both $x(k)$ and $u(k)$ are at the same event step, while the former model is easier for MPC purposes due to the similar definition of $\tilde{y}(k)$ and $\tilde{u}(k)$ in (3.3), in which both contains values from event step k up to $k + N_\mathrm{p} - 1$. □

In order to identify the unknown system parameters, we need to distinguish between the parameters that are known a priori, i.e., the parameters that are either constant or determined in advance such as the nominal transportation times in a production system, and the parameters that have to be identified. Therefore, the i-th row of the matrix $Q(k)$ can be written as:

$$Q_{i,\cdot}(k) = \Xi_{i,\cdot} + \theta^T \Delta^{(i)} + e^T(k) \Lambda S^{(i)} \tag{3.19}$$

where Ξ represents the parameters that are known a priori, θ is a vector of unknown parameters, $e(k) = [e_1(k), \ldots, e_{n_e}(k)]^T$ is a vector the elements of which are independent random variables, the diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{n_e})$ contains the amplitude of the noise, and $\Delta^{(i)}$ and $S^{(i)}$ are selection matrices for the i-th row with zeros and ones as entries. The role of the selection matrices is to determine which elements of the vectors θ and $e(k)$ will appear in the i-th row of $Q(k)$. For example, for the first row $i = 1$, let $\Delta^{(1)} = [1 \quad 0 \quad 1]^T$ and $S^{(1)} = [0 \quad 1 \quad 1]^T$; then $\theta^T \Delta^{(1)} = [\theta_1 \quad \theta_2 \quad \theta_3] \cdot [1 \quad 0 \quad 1]^T = \theta_1 + \theta_3$ and $e^T(k) \Lambda S^{(1)} = [e_1(k) \quad e_2(k) \quad e_3(k)] \cdot \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3) \cdot [0 \quad 1 \quad 1]^T = e_2(k)\lambda_2 + e_3(k)\lambda_3$.

We assume that the probability density function of $e(k)$, denoted by $f(e)$, and the matrices Ξ, Δ, and S are known a priori and that the only parameters that have to be identified are the components of θ and the diagonal elements of Λ, denoted by $\lambda = [\lambda_1, \ldots, \lambda_{n_e}]^T$. Therefore, in the identification procedure, we will derive estimates $\hat{\theta}$ and $\hat{\lambda}$ for θ and λ, respectively.

Here, the identification procedure is based on input-state data. Note that in MPL systems the state contains the time instants at which the state events occur. We assume that the state is observable and hence, these instants can be measured easily and so we usually have full state information. Consider the measured input-state sequence $\{(u_\mathrm{meas}(k), x_\mathrm{meas}(k)\}_{k=1}^N$ of a system of the form (3.18) and assume that the system parameters $\hat{\theta}$ and $\hat{\lambda}$ have to be identified using this sequence. Further, we assume that the input-state sequence is sufficiently rich[5] to capture all the relevant information about the system (see also [97, 98]). Now consider the following identification problem:

$$\begin{aligned} &\min_{(\hat{\theta},\hat{\lambda})} \quad J(\hat{\theta},\hat{\lambda}) \\ &\text{s.t.} \qquad\quad \hat{\lambda} > 0 \end{aligned} \tag{3.20}$$

with

$$J(\hat{\theta},\hat{\lambda}) = \sum_{k=1}^{N-1}\sum_{i=1}^{n_x}\left(\mathbb{E}[x_i(k+1|k)] - x_{\text{meas},i}(k+1)\right)^2 \tag{3.21}$$

where $\mathbb{E}[\cdot]$ denotes the expected value operator and $\mathbb{E}[x_i(k+1|k)]$ is the one-step-ahead prediction of x_i for event step $k+1$, using the knowledge from event step k. Considering (3.18) and (3.19), we can rewrite the one-step-ahead prediction as

$$\mathbb{E}[x_i(k+1|k)] = \mathbb{E}\big[\big(\Xi_{i,\cdot} + \theta^T\Delta^{(i)} + e^T(k)\Lambda S^{(i)}\big) \otimes \phi(k)\big]$$

and hence, the one-step-ahead prediction error will be given by

$$\begin{aligned} &\mathbb{E}[x_i(k+1|k)] - x_{\text{meas},i}(k+1) \\ &\quad = \mathbb{E}\big[\max_j\big(\xi_{ij} + \hat{\theta}^T\Delta^{(i)}_{\cdot,j} + e^T(k)\hat{\Lambda}S^{(i)}_{\cdot,j} + \phi_j(k) - x_{\text{meas},i}(k+1)\big)\big] \\ &\quad = \mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))] \end{aligned} \tag{3.22}$$

where

$$\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k)) = \max_j\big(\xi_{ij} + \hat{\theta}^T\Delta^{(i)}_{\cdot,j} + e^T(k)\hat{\Lambda}S^{(i)}_{\cdot,j} + \phi_j(k) - x_{\text{meas},i}(k+1)\big)$$

for a specific realization of the noise vector $e(k)$ and for $j = 1,\ldots,m$. To have a more compact notation, let $\alpha_{ij}(k) = \xi_{ij} + \phi_j(k) - x_{\text{meas},i}(k+1)$, $\Pi_{ij} = \Delta^{(i)}_{\cdot,j}$, and $\Gamma_{ij} = \text{diag}((S^{(i)})_{1,j},\ldots,(S^{(i)})_{n_e,j})$. Since $e^T(k)\hat{\Lambda}S^{(i)}_{\cdot,j}$ is a scalar and $\hat{\Lambda}$ is a diagonal matrix, we have:

$$e^T(k)\hat{\Lambda}S^{(i)}_{\cdot,j} = (S^{(i)}_{\cdot,j})^T\hat{\Lambda}e(k) = \hat{\lambda}^T\Gamma_{ij}e(k)$$

Therefore, we can rewrite $\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))$ as

$$\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k)) = \max_{j=1,\ldots,m}\left(\alpha_{ij}(k) + \Pi^T_{ij}\hat{\theta} + \hat{\lambda}^T\Gamma_{ij}e(k)\right) \tag{3.23}$$

which is now an MPL expression.

[5] Intuitively, this can be characterized as follows. Note that (3.18) and (3.19) imply that each component of $x(k+1)$ can be written as a max expression of terms in which the unknown parameters θ and λ appear. An input signal is then said to be sufficiently rich if it is such that each of these terms is the maximal one sufficiently often (this is also related to the idea of persistent excitation in conventional system identification [73]).

3.2.2 Solution Approach

In a similar way as Section 3.1.2, $\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]$ can be written as follows:

$$\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))] = \sum_{j=1}^{m} \int \cdots \int_{e\in\Omega_{ij}(\hat{\theta},\hat{\lambda},k)} (\alpha_{ij}(k) + \Pi_{ij}^T\hat{\theta} + \hat{\lambda}^T\Gamma_{ij}e)f(e)\mathrm{d}e \tag{3.24}$$

where $\mathrm{d}e = \mathrm{d}e_1,\ldots,\mathrm{d}e_{n_e}$ and $\Omega_{ij}(\hat{\theta},\hat{\lambda},k), i = 1,\ldots,n$, $j = 1,\ldots,m$ are polyhedral sets. Hence, the following proposition can be obtained.

Proposition 3.2.2 *([104, Proposition 3]) The function* $\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]$, *defined in* (3.22), *is convex in $\hat{\theta}$ and $\hat{\lambda}$, and its subgradients with respect to $\hat{\theta}$ and $\hat{\lambda}$ are*

$$g_{i,\hat{\theta}}(\hat{\theta},\hat{\lambda},k) = \sum_{j=1}^{m} \Big(\int \cdots \int_{e\in\Omega_{ij}(\hat{\theta},\hat{\lambda},k)} f(e)de\Big)\Pi_{ij} \tag{3.25}$$

$$g_{i,\hat{\lambda}}(\hat{\theta},\hat{\lambda},k) = \sum_{j=1}^{m} \Big(\int \cdots \int_{e\in\Omega_{ij}(\hat{\theta},\hat{\lambda},k)} e^T f(e)de\Big)\Gamma_{ij} \tag{3.26}$$

respectively.

Therefore, $J(\hat{\theta},\hat{\lambda})$ in the identification problem (3.20) can be written as

$$J(\hat{\theta},\hat{\lambda}) = \sum_{k=1}^{N}\sum_{i=1}^{n_x} \Big(\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\Big)^2 \tag{3.27}$$

with the gradients

$$\nabla_{\hat{\theta}} J(\hat{\theta},\hat{\lambda}) = \sum_{k=1}^{N}\sum_{i=1}^{n_x} 2\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\, g_{i,\hat{\theta}}(\hat{\theta},\hat{\lambda},k)$$

$$\nabla_{\hat{\lambda}} J(\hat{\theta},\hat{\lambda}) = \sum_{k=1}^{N}\sum_{i=1}^{n_x} 2\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\, g_{i,\hat{\lambda}}(\hat{\theta},\hat{\lambda},k).$$

Note that the identification problem (3.20) is a non-convex optimization problem due to the fact that $J(\hat{\theta},\hat{\lambda})$ in not convex for the following reasoning. By definition [93], a function $f(x) = (g(x))^p,\ p \geq 1$, is convex if g is convex and nonnegative. In our case $p = 2$ and $g(\hat{\theta},\hat{\lambda}) = \mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]$. Considering the definition of $\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]$ in (3.22), it is convex in $\hat{\theta}$ and $\hat{\lambda}$ but it is not nonnegative.

Therefore, $J(\hat{\theta}, \hat{\lambda})$ is not convex and hence, to solve (3.20), we need global non-convex optimization methods such as genetic algorithms [46], simulated annealing [34], etc. However, since the gradients of $J(\hat{\theta}, \hat{\lambda})$ are available, the optimal $\hat{\theta}$ and $\hat{\lambda}$ can also be found using multi-start gradient-based local optimization methods, such as a steepest descent method or a Quasi-Newton (DFP, BFGS) method [90].

3.2.3 Computational Aspects and Issues

Note that to compute the objective function (3.27), we first need to find the value of $\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]$, which leads to the solution of the integral in (3.24). Considering the methods explained in Section 3.1.3, $\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]$ can be computed using either numerical integration or analytic integration based on piecewise polynomial probability density functions.

Note that numerical integration is not an efficient way of computing the integral (3.24), since it is quite complex and time-inefficient. Furthermore, based on the complexity analysis of the analytic integration using piecewise polynomial probability density functions in Section 3.1.3, this method is also not computationally efficient and its complexity increases drastically when the number of random variables or the order of the system increase.

3.3 Summary

In this chapter, an overview of MPC and identification of stochastic MPL systems has been given. MPC is a model-based control approach that can be applied to both discrete-time and discrete-event systems. We have discussed how MPC can be used to control stochastic MPL systems. Moreover, MPC has also been applied to stochastic switching MPL systems and stochastic MMPS systems. Next, we have discussed the identification of the model parameters of a stochastic MPL system defined by a state space model based on input-state data.

Furthermore, we have discussed some existing solution approaches and computational aspects of MPC and identification of stochastic MPL systems. Since we deal with stochastic systems, the solution of these problems leads to the computation of an expected value. Some of the solution approaches and their computational aspects and complexities have been presented in this chapter. One approach is to use numerical integration, which is quite complex and time-consuming. Another approach is an analytic integration method that can be applied to distributions that have a piecewise affine polynomial probability density function, or when their probability density functions can be approximated by such functions. The complexity of this approach increases significantly as the number of stochastic variables or the order of system increase. An approximation approach base on variability expansion has been also discussed, which reduces the computational complexity of the analytic integration to some extent.

Since the above-mentioned computational approaches are complex and time-consuming, stochastic MPL systems have been mostly studied with the assumption that the noise and disturbances of the system are bounded without any further knowledge of their probability distribution. In the next chapter, we propose an alternative computation method that allows to use the knowledge about the distribution of the system noise. This method is less complex and more time-efficient compared with the two other methods discussed in this chapter (viz. the analytic integration based on piecewise polynomial probability density functions and the approximation method based on variability expansion).

Chapter 4

An Approximation Method for Computing the Expected Value of Max-Plus Functions

In this chapter, an approximation method based on the higher order moments of a random variable is proposed to compute the expected value of the maximum of several affine expressions. First, we explain in detail how this approximation method can be obtained and then, we compute an upper bound for the error introduced by this method. In the last section of this chapter, the convexity analysis of this approximation method is presented.

4.1 Problem Statement

Our aim is to compute the expected value of the maximum of several affine expressions (cf. Section 3.1.2, equation (3.10) and Section 3.2.2, equation (3.24)) efficiently. One solution approach is numerical integration, which imposes a huge computational burden, especially when the number of stochastic variables is high or the function to be integrated is non-linear. An analytic approach for the integral computation is proposed in [104] (cf. Section 3.1.3). However, this method is still very time-consuming and complex since at each event step many polyhedra have to be computed in order to compute the integral. To decrease the computational burden, an approximation method based on variability expansion is proposed in [110]. However, this method cannot reduce the complexity of the problem sufficiently and consequently the problem remains complex. Therefore, it is still desired to find an efficient method to either compute or approximate the expected value of the maximum of several affine expressions. To this end, we introduce an alternative approximation method that is based on the higher order moments of a random variable.

4.2 Description of the Approximation Method

This approximation approach is inspired by the relation between the ∞-norm and the p-norm of a vector.

Definition 4.2.1 ([47]) *For a vector $x \in \mathbb{R}^n$ and for $p \geq 1$, the p-norm and the ∞-norm of x are defined as:*

$$\begin{aligned} \|x\|_p &= \big(|x_1|^p + \cdots + |x_n|^p\big)^{1/p} \\ \|x\|_\infty &= \max(|x_1|, \ldots, |x_n|), \end{aligned} \tag{4.1}$$

respectively.

The relation between these norms is as follows [47]:

$$\|x\|_\infty \leq \|x\|_p \leq n^{1/p}\|x\|_\infty \tag{4.2}$$

Moreover, due to the monotonic and linear properties of the expected value, we have

$$\mathbb{E}\big[\|x\|_\infty\big] \leq \mathbb{E}\big[(|x_1|^p + \cdots + |x_n|^p)^{1/p}\big]. \tag{4.3}$$

Before we proceed further, consider the following theorem.

Theorem 4.2.2 (Jensen's Inequality [14]) *If x is a random variable such that $x \in \mathrm{dom}(\varphi)$ with probability one, and φ is a concave function, then $\mathbb{E}\left[\varphi(x)\right] \leq \varphi\left(\mathbb{E}\left[x\right]\right)$, provided the expectations exist. Likewise, if φ is a convex function, then $\varphi\left(\mathbb{E}\left[x\right]\right) \leq \mathbb{E}\left[\varphi(x)\right]$.*

Now, the following proposition shows how we can apply p-norms to find an upper bound for $\mathbb{E}\big[\max(x_1, \ldots, x_n)\big]$.

Proposition 4.2.3 *Consider random variables x_j for $j = 1, \ldots, n$ and let $p > 1$. Then*

$$\begin{aligned} \mathbb{E}\big[\max(x_1, \ldots, x_n)\big] &\overset{(i)}{\leq} \mathbb{E}\big[\max(|x_1|, \ldots, |x_n|)\big] \\ &\overset{(ii)}{\leq} \mathbb{E}\big[(|x_1|^p + \cdots + |x_n|^p)^{1/p}\big] \\ &\overset{(iii)}{\leq} \Bigg(\sum_{j=1}^{n} \mathbb{E}\big[|x_j|^p\big]\Bigg)^{1/p} \end{aligned} \tag{4.4}$$

Proof : Inequality (i) is straightforward. Inequality (ii) results from (4.3). Inequality (iii) results from Jensen's inequality for concave functions, i.e., Theorem 4.2.2. Note that we can apply Theorem 4.2.2 since $\varphi(x) = x^{1/p}$ is a concave function for $p > 1$ and $x > 0$, and in our case the argument x is $\sum_{i=1}^{n} |x_i|^p$ which is positive. □

Inequality (i) reduces to an equality if all variables x_j are nonnegative. Hence, in order to reduce the error in Inequality (i) for $j = 1, \ldots, n$, we define an offset L such that $x_j - L$ is almost always positive. Note that if x_j is drawn from a distribution with a finite domain (such as the uniform distribution), L can be defined such that $L \leq x_j$ for $j = 1, \ldots, n$ and hence, Inequality (i) turns into an equality. However, if x_j has an infinite domain (such as the normal distribution), Inequality (i) never reduces to an equality and we can only decrease the error by defining L such that it is less than or equal almost all $x_j,\ j = 1, \ldots, n$. For example if $x_j,\ j = 1, \ldots, n$ are normally distributed with mean μ_j and variance σ_j^2, then L can be defined as $L = \min_{j=1,\ldots,n}(\mu_j - 3\sigma_j)$. This choice of L is made based on the 3σ-rule, which states that 99.7% of the observations of a normally distributed random variable with mean μ and variance σ^2 fall within the interval $[\mu - 3\sigma, \mu + 3\sigma]$.

Remark 4.2.4 In the case that all $x_j,\ j = 1, \ldots, n$ are nonnegative, using the offset L is still useful. Indeed, in that case, L is not needed for Inequality (i) as it is an equality. However, for Inequality (ii), L, which is defined such that $L \leq x_j$ for $j = 1, \ldots, n$, will reduce the error. This is illustrated by the following example: Let $x_1 = 1000,\ x_2 = 1001$, then $\max(x_1, x_2) = 1001$. Now for $p = 2$, if $L = 0$ then $(x_1^p + x_2^p)^{1/p} = 1000\sqrt{2}$; however, for $L = \min(x_1, x_2) = 1000$, $((x_1 - L)^p + (x_2 - L)^p)^{1/p} + L = 1001$, which is equal to $\max(x_1, x_2)$. □

Accordingly, we can rewrite (4.4) as follows:

$$\begin{aligned}
\mathbb{E}\big[\max(x_1, \ldots, x_n)\big] &= \mathbb{E}\big[\max(x_1 - L, \ldots, x_n - L)\big] + L \\
&\leq \mathbb{E}\big[\max(|x_1 - L|, \ldots, |x_n - L|)\big] + L \\
&\leq \mathbb{E}\big[(|x_1 - L|^p + \cdots + |x_n - L|^p)^{1/p}\big] + L \\
&\leq \left(\sum_{j=1}^{n} \mathbb{E}\big[|x_j - L|^p\big]\right)^{1/p} + L. \qquad (4.5)
\end{aligned}$$

Remark 4.2.5 For a positive even integer $p = 2q,\ q \in \mathbb{N} \setminus \{0\}$, we have $\mathbb{E}[x^p] = \mathbb{E}[|x|^p]$. Hence, without loss of generality, we can use $\mathbb{E}[x^p]$ in (4.5). So from now on, p is an even integer larger than or equal to 2. □

Considering the above remark, we can approximate $\mathbb{E}[\max(x_1,\dots,x_n)]$ by its upper bound defined as:

$$\mathfrak{U}\Big(\mathbb{E}[\max(x_1,\dots,x_n)]\Big) = \Big(\sum_{j=1}^{n} \mathbb{E}\big[(x_j - L)^p\big]\Big)^{1/p} + L \tag{4.6}$$

for p a positive even integer and for independent random variables $x_j,\ j = 1,\dots,n$, and L is chosen such that $L \leq x_j$ for (almost) all $j \in \{1,\dots,n\}$.

Theorem 4.2.6 (Multinomial Theorem [54]) *For any positive integer m and any nonnegative integer n, we have:*

$$(x_1 + x_2 + \cdots + x_m)^n = \sum_{\substack{k_1+k_2+\cdots+k_m=n\\ k_1,k_2,\dots,k_m\in\mathbb{N}}} \binom{n}{k_1,k_2,\dots,k_m} \prod_{t=1}^{m} x_t^{k_t},$$

where

$$\binom{n}{k_1,k_2,\dots,k_m} = \frac{n!}{k_1!\,k_2!\cdots k_m!}$$

is a multinomial coefficient.

Recall that in both MPC and identification optimization problem of stochastic MPL systems (cf. Chapter 3), the expected value of max-plus-scaling functions appears in the definition of objective functions. This expected value is in general of the form $\mathbb{E}[\max_{j=1,\dots,n}(\beta_j + \gamma_j^T \tilde{e})]$ where $\beta_j \in \mathbb{R}_\varepsilon$, $\gamma_j \in \mathbb{R}^{n_{\tilde{e}}}$, and $\tilde{e}$ is a vector of independent random variables with a given probability distribution. Our aim in this chapter is to find an upper bound for $\mathbb{E}[\max_{j=1,\dots,n}(\beta_j + \gamma_j^T \tilde{e})]$. By adopting (4.5), we obtain:

$$\begin{aligned}
&\mathbb{E}[\max_{j=1,\dots,n}(\beta_j + \gamma_j^T \tilde{e})]\\
&\qquad \leq \Big(\sum_{j=1}^{n} \mathbb{E}\big[(\beta_j - L + \gamma_j^T \tilde{e})^p\big]\Big)^{1/p} + L\\
&\qquad \leq \Big(\sum_{j=1}^{n} \mathbb{E}\big[(\underbrace{\beta_j - L}_{z_{j,0}} + \underbrace{\gamma_{j,1}\tilde{e}_1}_{z_{j,1}} + \cdots + \underbrace{\gamma_{j,n_{\tilde{e}}}\tilde{e}_{n_{\tilde{e}}}}_{z_{j,n_{\tilde{e}}}})^p\big]\Big)^{1/p} + L\\
&\qquad \overset{(*)}{\leq} \Big(\sum_{j=1}^{n} \mathbb{E}\Big[\sum_{\substack{k_0+k_1+\cdots+k_{n_{\tilde{e}}}=p\\ k_0,k_1,\dots,k_{n_{\tilde{e}}}\in\mathbb{N}}} \binom{p}{k_0,k_2,\dots,k_{n_{\tilde{e}}}} \prod_{t=0}^{n_{\tilde{e}}} z_{j,t}^{k_t}\Big]\Big)^{1/p} + L\\
&\qquad \leq \Big(\sum_{j=1}^{n} \sum_{\substack{k_0+k_1+\cdots+k_{n_{\tilde{e}}}=p\\ k_0,k_1,\dots,k_{n_{\tilde{e}}}\in\mathbb{N}}} \frac{p!}{k_0!\,k_2!\cdots k_{n_{\tilde{e}}}!} \mathbb{E}\big[\prod_{t=0}^{n_{\tilde{e}}} z_{j,t}^{k_t}\big]\Big)^{1/p} + L
\end{aligned}$$

$$\overset{(**)}{\leq} \Bigg(\sum_{j=1}^{n} \sum_{\substack{k_0+k_1+\cdots+k_{n_{\tilde{e}}}=p \\ k_0,k_1,\ldots,k_{n_{\tilde{e}}}\in\mathbb{N}}} \frac{p!}{k_0!\,k_2!\cdots k_{n_{\tilde{e}}}!} \prod_{t=0}^{n_{\tilde{e}}} \mathbb{E}\big[z_{j,t}^{k_t}\big] \Bigg)^{1/p} + L \tag{4.7}$$

where $(*)$ is based on multinomial theorem (Theorem 4.2.6), and $(**)$ is due to the fact that the elements of the stochastic vector $\tilde{e}$, i.e., $\tilde{e}_1, \ldots, \tilde{e}_{n_{\tilde{e}}}$ are independent by assumption and for independent random variables $Z_1, \ldots, Z_{n_{\tilde{e}}}$, we have $\mathbb{E}[\prod_{t=1}^{n_{\tilde{e}}} Z_t] = \prod_{t=1}^{n_{\tilde{e}}} \mathbb{E}[Z_t]$.

Consequently, we can rewrite (4.6) as follows:

$$\begin{aligned} &\mathfrak{U}\Big(\mathbb{E}[\max_{j=1,\ldots,n}(\beta_j + \gamma_j^T \tilde{e})]\Big) \\ &\quad = \Bigg(\sum_{j=1}^{n} \sum_{\substack{k_0+k_1+\cdots+k_{n_{\tilde{e}}}=p \\ k_0,k_1,\ldots,k_{n_{\tilde{e}}}\in\mathbb{N}}} \frac{p!}{k_0!\,k_2!\cdots k_{n_{\tilde{e}}}!} (\beta_j - L)^{k_0} \prod_{t=1}^{n_{\tilde{e}}} \gamma_{j,t}^{k_t} \mathbb{E}\big[\tilde{e}_t^{k_t}\big] \Bigg)^{1/p} + L. \end{aligned} \tag{4.8}$$

Note that $\beta_j - L$ it is not a random variable since it does not include any elements of the stochastic vector $\tilde{e}$ and hence, we have $\mathbb{E}[(\beta_j - L)^{k_0}] = (\beta_j - L)^{k_0}$.

In the approximation function (4.8), we have to compute the k_t-th moment of each random variable $\tilde{e}_t,\ t = 0, \ldots, n_{\tilde{e}}$. The κ-th moment of a real-valued random variable e is defined as follows:

$$\mathbb{E}[e^{\kappa}] = \int_{-\infty}^{\infty} e^{\kappa} f(e) de \tag{4.9}$$

where $f(\cdot)$ is the probability density function of e.

In general, moments of a random variable can be finite or infinite. Hence, to be able to usefully apply $\mathfrak{U}\Big(\mathbb{E}[\max_{j=1,\ldots,n}(\beta_j + \gamma_j^T \tilde{e})]\Big)$ as an approximation of $\mathbb{E}[\max_{j=1,\ldots,n}(\beta_j + \gamma_j^T \tilde{e})]$, we need to consider random variables with finite moments for which a closed-form expression exists, such as variables with the uniform distribution, normal distribution, Beta distribution, etc. [10, 62, 89]. Note that if moments do not have a closed-form expression, one has to solve the integral (4.9) numerically. In that case, the approximation method will be less time-efficient than in the case the closed-form of the moments exists and depending on whether the integral (4.9) can be computed offline or not, applying numerical or analytic integration, presented in Section 3.1.2, directly to $\mathbb{E}[\max_{j=1,\ldots,n}(\beta_j + \gamma_j^T \tilde{e})]$ could even be better options. In the following, we present some examples of distributions that have finite moments with a closed-form expression: the uniform distribution, the Beta distribution, and the normal distribution. Some examples of other distributions that have finite moments with a closed-form expression can be found in [10, 62, 89].

For the case of a uniformly distributed random variable e on an interval $[a, b]$, i.e., $e \sim \mathcal{U}(a, b)$, the κ-th moment can be computed as [10]

$$\mathbb{E}[e^{\kappa}] = \frac{1}{\kappa+1} \sum_{l=0}^{\kappa} a^{l} b^{\kappa-l}. \tag{4.10}$$

For a random variable e that has a Beta distribution with parameters α and β, i.e., $e \sim \mathcal{B}(\alpha, \beta)$, the κ-th moment can be written in a recursive form as [10]

$$\mathbb{E}[e^{\kappa}] = \frac{\alpha+\kappa-1}{\alpha+\beta+\kappa-1} \mathbb{E}(e^{\kappa-1}). \tag{4.11}$$

In case of a normally distributed random variable e with mean μ and variance σ^2, i.e., $e \sim \mathcal{N}(\mu, \sigma^2)$, the κ-th moment has a closed-form that can be expressed as [117]:

$$\mathbb{E}\big[e^{\kappa}\big] = \sigma^{\kappa} i^{-\kappa} H_{\kappa}(i\mu/\sigma) \tag{4.12}$$

where

$$H_{\kappa}(e) \equiv (-1)^{\kappa} \exp(e^2/2) \frac{d^{\kappa}}{de^{\kappa}} \exp(-e^2/2)$$

is the κ-th Hermite polynomial. Note that the right-hand side of (4.12) is in fact real because $H_{\kappa}(e)$ contains only even powers of e if κ is even (note that here we assume that $\kappa = 2q,\ q \in \mathbb{N} \setminus \{0\}$). Considering equations (26.2.51) and (22.3.11) in [1] leads to

$$H_{\kappa}(e) = \kappa! \sum_{l=0}^{\kappa/2} \frac{(-1)^{l} e^{\kappa-2l}}{2^{l} l! (\kappa-2l)!} \tag{4.13}$$

where $\kappa/2 \in \mathbb{N} \setminus \{0\}$ since κ is an even integer in our case.

Remark 4.2.7 For the case of a normally distributed stochastic vector $\tilde{e}$, the random variable $x_j - L = \beta_j - L + \gamma_j^T \tilde{e}$ is also normally distributed with a certain mean and variance, using the property of the normal distribution that sum of the independent normally distributed random variables has also a normal distribution with a new mean and variance [31]. Hence, we can immediately compute the p-th moment in (4.6) and we do not need to use (4.8). In this way, our computation will be faster since we have less terms (compare (4.6) with (4.8)). In general, this remark is valid for all distributions that are preserved under the summation and for which a closed form of their higher-order moments exists, such as the Poisson and the Gamma distribution [89]. □

Note that the computational complexity of this approximation method is much less than the two other methods in Section 3.1.3. Considering (4.8), the total number of terms in the first sum is n and in the second sum, i.e., the multinomial sum, is $\binom{p+n_{\tilde{e}}-1}{p}$ [40], or equivalently,

$$\binom{p+n_{\tilde{e}}-1}{p} = \frac{(p+n_{\tilde{e}}-1)!}{p!(n_{\tilde{e}}-1)!} = \frac{\overbrace{(p+n_{\tilde{e}}-1)(p+n_{\tilde{e}}-2)\dots n_{\tilde{e}}}^{p \text{ terms}}}{p!}.$$

Hence, assuming that $n_{\tilde{e}} \gg p > 1$, the order of the error for the above sum is $O\Big(\frac{n_{\tilde{e}}^p}{p!}\Big)$. Also, the total number of the expected values that have to be computed is $pn_{\tilde{e}}$. Hence, the complexity of this approximation method is of the order

$$O\Big(n \cdot n_{\tilde{e}} \cdot p \cdot \frac{n_{\tilde{e}}^p}{p!}\Big) = O\Big(\frac{n \cdot n_{\tilde{e}}^{p+1}}{(p-1)!}\Big)$$

which increases polynomially as n and $n_{\tilde{e}}$ increase[1] and exponentially as p increases (at least for $n_{\tilde{e}} \gg p$). Note that the computation load depends also on the prediction horizon. By increasing the prediction horizon, the number of terms in the maximization n (or n_v in Chapter 3) may also increase significantly. This leads to an increase of the number of function evaluation in approximation method as well as in the other methods in Section 3.1.3.

4.3 On Error of the Approximation Method

In this section, we show that the error caused by approximating $\mathbb{E}[\max(x_1,\dots,x_n)]$ by its upper bound $\mathfrak{U}\Big(\mathbb{E}[\max(x_1,\dots,x_n)]\Big) = \big(\sum_{j=1}^n \mathbb{E}[(x_j-L)^p]\big)^{1/p} + L$ (cf. (4.6)) is bounded. Note that $\mathbb{E}[\max(x_1,\dots,x_n)]$ is bounded from above and from below. Indeed, its upper bound has been presented in (4.6) and its lower bound can be obtained using Jensen's inequality for convex functions, the max function in this case (cf. Theorem 4.2.2). Let $\mathfrak{L}\Big(\mathbb{E}[\max(x_1,\dots,x_n)]\Big) = \max(\mathbb{E}[x_1],\dots,\mathbb{E}[x_n])$ denote this lower bound. Hence,

$$\mathfrak{L}\Big(\mathbb{E}[\max(x_1,\dots,x_n)]\Big) \le \mathbb{E}[\max(x_1,\dots,x_n)] \le \mathfrak{U}\Big(\mathbb{E}[\max(x_1,\dots,x_n)]\Big). \tag{4.14}$$

Remark 4.3.1 In the trivial case that there is only one term in the $\max$ operator and also $p = 1$, the lower bound and the upper bound in (4.14) are equal. However, in

[1] In general for a given problem we have $n = n_v$ (cf. Section 3.1.3 of Chapter 3)

general, there are more than one term in the maximization and $p \to \infty$. We will also show later in Proposition 4.3.3 that in the case all $x_j \leq L, j = 1, \dots, n$, the difference between the upper bound and the exact value is always larger than 0 as $p \to \infty$. □

The error of approximating $\mathbb{E}[\max(x_1, \dots, x_n)]$ by its upper bound is then, always bounded as follows:

$$\begin{aligned} 0 &\leq \mathfrak{U}\Big(\mathbb{E}[\max(x_1, \dots, x_n)]\Big) - \mathbb{E}[\max(x_1, \dots, x_n)] \\ &\leq \mathfrak{U}\Big(\mathbb{E}[\max(x_1, \dots, x_n)]\Big) - \mathfrak{L}\Big(\mathbb{E}[\max(x_1, \dots, x_n)]\Big) \end{aligned} \tag{4.15}$$

and since in our case $x_j,\ j = 1, \dots, n$ are assumed to have finite moments, this upper bound is finite and the error of the approximation cannot be larger than this value.

As mentioned in the previous section, L is an off-set that is chosen such that $L \leq x_j$ for all $j = 1, \dots, n$ in the case that x_j has a distribution with a bounded domain and it is less than or equal to almost all $x_j,\ j = 1, \dots, n$ if the distribution of x_j has an unbounded domain. In this way, the error caused by Inequality (i) in (4.4) will be zero or decreased to a large extent. Note that even if all $x_j \geq 0,\ j = 1, \dots, n$ and hence, Inequality (i) turns into an equality, a positive L can still be applied in order to decrease the error caused by Inequality (ii) in (4.4) (cf. Remark 4.2.4).

Alternatively, in the case that each x_j has a distribution with a finite domain $\mathcal{X}_j$, we can introduce another upper bound for the error caused by applying the upper bound approximation function (4.6). To obtain the new upper bound, we consider the three inequalities in (4.4) and their corresponding error. The first error, due to (i), will be zero if $L = \min_{j=1,\dots,n} \min \mathcal{X}_j$. The second error due to (ii) approaches zero if $p \to \infty$, since by definition $\|x\|_\infty = \lim_{p\to\infty} \|x\|_p$. However, the third error, which is in fact the error of Jensen's inequality, needs more discussion. In [99, Theorem 2.1] and [100, Theorem 2.1] two upper bounds for Jensen's inequality are presented for the relative and the absolute error, respectively.

For a strictly positive, twice continuously differentiable, concave function f defined on an interval $[a, b]$, Jensen's inequality can be stated in the form

$$1 \leq \frac{f(\mathbb{E}[x])}{\mathbb{E}[f(x)]}$$

for which an upper bound can be formulated as follows [99]:

$$1 \leq \frac{f(\mathbb{E}[x])}{\mathbb{E}[f(x)]} \leq \max_{q\in[0,1]} \left[\frac{f(qa + (1-q)b)}{qf(a) + (1-q)f(b)}\right] =: S_f(a, b)$$

and it has been proved in [99] that there exists a unique $q_0 \in (0,1)$ for which $S_f(a,b)$ is maximal. Hence, the relative error can be defined as follows:

$$0 \leq \frac{f(\mathbb{E}[x]) - \mathbb{E}[f(x)]}{\mathbb{E}[f(x)]} \leq S_f(a,b) - 1 =: e_{\text{max,rel}}(a,b).$$

In a similar way, the absolute error can also be defined as follows [100]: For a differentiable, concave function f defined on an interval $[a,b]$ we have

$$\begin{aligned} 0 \leq f(\mathbb{E}[x]) - \mathbb{E}[f(x)] &\leq \max_{\omega \in [0,1]} [f(\omega a + (1-\omega)b) - \omega f(a) - (1-\omega) f(b)] \\ &=: e_{\text{max,abs}}(a,b) \end{aligned}$$

and again it has been shown that there exists a unique $\omega_0 \in (0,1)$ for which $e_{\text{abs}}(a,b)$ is maximal [100]. Note that since f is a strictly positive concave function, $a, b > 0$.

In our case the concave function is $f(x) = x^{1/p}$ and $f'(x) = \frac{1}{p}x^{\frac{1}{p}-1}$. Since we assume that p is a positive even integer greater than or equal to 2, the argument x has to be larger than or equal to zero, which is the case since $x = \sum_{j=1}^{n} x_j^p$. Let us first consider the case where x is strictly positive. The case where $x = 0$ will be considered later on (see Proposition 4.3.2). By substituting f in the above formulas, we can determine the optimal value of q and ω for each case as follows. Let

$$F(q) = \frac{(qa + (1-q)b)^{\frac{1}{p}}}{qa^{\frac{1}{p}} + (1-q)b^{\frac{1}{p}}}.$$

Then, $F'(q) = g(q)/(qa^{\frac{1}{p}} + (1-q)b^{\frac{1}{p}})^2$ with

$$\begin{aligned} g(q) = \frac{1}{p}\big(qa + (1-q)b\big)^{\frac{1}{p}-1}\big(qa^{\frac{1}{p}} + (1-q)b^{\frac{1}{p}}\big)(a-b) - \\ (a^{\frac{1}{p}} - b^{\frac{1}{p}})\big(qa + (1-q)b\big)^{\frac{1}{p}}. \end{aligned}$$

Since $a, b > 0$ and $q \in [0,1]$, $\frac{dF}{dq}(q) = 0$ is only possible if $g(q) = 0$. Therefore, we must have

$$(qa+(1-q)b)^{\frac{1}{p}-1}\left[\frac{1}{p}(qa^{\frac{1}{p}} + (1-q)b^{\frac{1}{p}})(a-b) - (a^{\frac{1}{p}} - b^{\frac{1}{p}})(qa+(1-q)b)\right] = 0$$

Since $(qa + (1-q)b)^{\frac{1}{p}-1} \neq 0$, we can conclude that

$$\frac{1}{p}(qa^{\frac{1}{p}} + (1-q)b^{\frac{1}{p}})(a-b) - (a^{\frac{1}{p}} - b^{\frac{1}{p}})(qa + (1-q)b) = 0$$

Hence[2],

$$\begin{aligned}
&\frac{1}{p}qa^{\frac{1}{p}+1}+\frac{1}{p}(1-q)ab^{\frac{1}{p}}-\frac{1}{p}qa^{\frac{1}{p}}b-\frac{1}{p}(1-q)b^{\frac{1}{p}+1}\\
&\quad -qa^{\frac{1}{p}+1}-(1-q)a^{\frac{1}{p}}b+qab^{\frac{1}{p}}+(1-q)b^{\frac{1}{p}+1}=0\\
&\Leftrightarrow q\big(\frac{1-p}{p}a^{\frac{1}{p}+1}-\frac{1}{p}a^{\frac{1}{p}}b+ab^{\frac{1}{p}}\big)+(1-q)\big(\frac{p-1}{p}b^{\frac{1}{p}+1}+\frac{1}{p}ab^{\frac{1}{p}}-a^{\frac{1}{p}}b\big)=0\\
&\Leftrightarrow q_0=\frac{\frac{1-p}{p}b^{\frac{1}{p}+1}-\frac{1}{p}ab^{\frac{1}{p}}+a^{\frac{1}{p}}b}{\frac{1-p}{p}(a^{\frac{1}{p}+1}+b^{\frac{1}{p}+1}-a^{\frac{1}{p}}b-ab^{\frac{1}{p}})}\\
&\quad 1-q_0=\frac{\frac{1-p}{p}a^{\frac{1}{p}+1}-\frac{1}{p}\;a^{\frac{1}{p}}b+ab^{\frac{1}{p}}}{\frac{1-p}{p}(a^{\frac{1}{p}+1}+b^{\frac{1}{p}+1}-a^{\frac{1}{p}}b-ab^{\frac{1}{p}})}
\end{aligned}$$

Similarly, we can obtain the optimal ω. To this end, let $G(\omega)=(\omega a+(1-\omega)b)^{\frac{1}{p}}-(\omega a^{\frac{1}{p}}+(1-\omega)b^{\frac{1}{p}})$, and hence,

$$\frac{dG}{d\omega}(\omega)=\frac{1}{p}(\omega a+(1-\omega)b)^{\frac{1}{p}-1}(a-b)-(a^{\frac{1}{p}}-b^{\frac{1}{p}}).$$

Now to find the maximum, we have:

$$\begin{aligned}
\frac{dG}{d\omega}(\omega)=0\Leftrightarrow &\quad (\omega a+(1-\omega)b)^{\frac{p-1}{p}}=\frac{a-b}{r(a^{\frac{1}{p}}-b^{\frac{1}{p}})}\\
\Leftrightarrow &\quad \omega(a-b)+b=\Big(\frac{a-b}{p(a^{\frac{1}{p}}-b^{\frac{1}{p}})}\Big)^{\frac{p}{p-1}}\\
\Leftrightarrow &\quad \omega_0=\frac{1}{a-b}\Big[\Big(\frac{a-b}{p(a^{\frac{1}{p}}-b^{\frac{1}{p}})}\Big)^{\frac{p}{p-1}}-b\Big]\\
1-\omega_0= &\ \frac{1}{a-b}\Big[a-\Big(\frac{a-b}{p(a^{\frac{1}{p}}-b^{\frac{1}{p}})}\Big)^{\frac{p}{p-1}}\Big]
\end{aligned}$$

Now, by replacing the optimal values q_0 and ω_0 in $F(q)$ and $G(\omega)$, respectively, the following expressions are obtained for $e_{\text{max,rel}}(a,b)$ and $e_{\text{max,abs}}(a,b)$:

$$\begin{aligned}
e_{\text{max,rel}}(a,b)=\Big[&\Big(\frac{\frac{1}{p}(ab^{\frac{1}{p}+1}-a^2b^{\frac{1}{p}}-a^{\frac{1}{p}}b^2+a^{\frac{1}{p}+1}b)}{-(\frac{p-1}{p})(a^{\frac{1}{p}+1}+b^{\frac{1}{p}+1}-a^{\frac{1}{p}}b-ab^{\frac{1}{p}})}\Big)^{\frac{1}{p}}\cdot\\
&\Big(\frac{-(\frac{p-1}{p})(a^{\frac{1}{p}+1}+b^{\frac{1}{p}+1}-a^{\frac{1}{p}}b-ab^{\frac{1}{p}})}{-a^{\frac{1}{p}}b^{\frac{1}{p}+1}-a^{\frac{1}{p}+1}b^{\frac{1}{p}}+a^{\frac{2}{p}}b+ab^{\frac{2}{p}}}\Big)\Big]-1
\end{aligned}\tag{4.16}$$

[2] If $q\alpha+(1-q)\beta=0\Rightarrow q=\frac{-\beta}{\alpha-\beta}$ and $1-q=\frac{\alpha}{\alpha-\beta}$.

$$e_{\max,\text{abs}}(a,b) = \Big(\frac{a-b}{p(a^{\frac{1}{p}} - b^{\frac{1}{p}})}\Big)^{\frac{1}{p-1}} \tag{4.17}$$
$$- \left(\frac{1}{a-b}\left[(a^{\frac{1}{p}} - b^{\frac{1}{p}})\left(\frac{a-b}{p(a^{\frac{1}{p}} - b^{\frac{1}{p}})}\right)^{\frac{p}{p-1}} - a^{\frac{1}{p}}b + ab^{\frac{1}{p}}\right]\right)$$

Hence, we derive the following proposition based on the above formulas.

Proposition 4.3.2 *Considering our assumptions that $a, b > 0$ and $p > 1$ is an even integer, we obtain the following result:*

$$\lim_{a\to 0^+} e_{\max,\text{rel}}(a,b) = \infty$$
$$\lim_{a\to 0^+} e_{\max,\text{abs}}(a,b) = b^{\frac{1}{p}}\left(\Big(\frac{1}{p}\Big)^{\frac{1}{p-1}} - \Big(\frac{1}{p}\Big)^{\frac{p}{p-1}}\right)$$

Proof : Let $b < \infty$ and $p < \infty$. Hence, the limit of $e_{\max,\text{rel}}(a,b)$ when $a \to 0^+$ is

$$\begin{aligned}
\lim_{a\to 0^+} e_{\max,\text{rel}}(a,b) &= \lim_{a\to 0^+}\left(\frac{-\frac{1}{p}a^{\frac{1}{p}}b^2}{-\frac{p-1}{p}b^{\frac{1}{p}+1}}\right)^{\frac{1}{p}} \cdot \frac{-\frac{p-1}{p}b^{\frac{1}{p}+1}}{-a^{\frac{1}{p}}b^{\frac{1}{p}+1}} - 1 \\
&= \lim_{a\to 0^+}\Big(\frac{1}{p-1}\Big)^{\frac{1}{p}} \cdot \frac{p-1}{p} \cdot \frac{b^{\frac{p-1}{p^2}}}{a^{\frac{p-1}{p^2}}} - 1 \\
&= \infty
\end{aligned}$$

and the limit of $e_{\max,\text{abs}}(a,b)$ when $a \to 0^+$ is obtained as

$$\begin{aligned}
\lim_{a\to 0^+} e_{\max,\text{abs}}(a,b) &= \Big(\frac{-b}{-pb^{\frac{1}{p}}}\Big)^{\frac{1}{p-1}} - \Big(\frac{-b^{\frac{1}{p}}}{-b}\Big(\frac{-b}{-pb^{\frac{1}{p}}}\Big)^{\frac{p}{p-1}}\Big) \\
&= \Big(\frac{1}{p}\Big)^{\frac{1}{p-1}}b^{\frac{1}{p}} - \Big(\frac{1}{p}\Big)^{\frac{p}{p-1}}b^{\frac{1}{p}} \\
&= b^{\frac{1}{p}}\left(\Big(\frac{1}{p}\Big)^{\frac{1}{p-1}} - \Big(\frac{1}{p}\Big)^{\frac{p}{p-1}}\right)
\end{aligned}$$

which is a finite value since both b and p are finite in this case. □

Since we have assumed that $y_j = \beta_j - L + \gamma_j^T\tilde{e}$ has a probability distribution with a finite domain, a and b can be easily obtained. Assume that each independent element of the stochastic vector $\tilde{e}$, i.e., $\tilde{e}_t,\ t = 1,\dots,n_{\tilde{e}}$, belongs to the interval $[\underline{c}_t, \bar{c}_t]$ where $\underline{c}_t, \bar{c}_t \in \mathbb{R}$. Since γ_{jt} can be positive or negative, we have

$$\min(\gamma_{jt}\underline{c}_t, \gamma_{jt}\bar{c}_t) \leq \gamma_{jt}\tilde{e}_t \leq \max(\gamma_{jt}\underline{c}_t, \gamma_{jt}\bar{c}_t).$$

Hence, we can show that each $y_j,\ j \in \{1,\dots,n\}$, also belongs to the interval $[a_j, b_j]$ where $0 \le a_j < b_j$ are defined as follows:

$$y_j = \beta_j - L + \gamma_j^T \tilde{e} = \beta_j - L + \gamma_{j1}\tilde{e}_1 + \cdots + \gamma_{jn_{\tilde{e}}}\tilde{e}_{n_{\tilde{e}}}$$

$$\Rightarrow \quad \underbrace{\beta_j - L + \sum_{t=1}^{n_{\tilde{e}}} \min(\gamma_{jt}\underline{c}_t, \gamma_{jt}\bar{c}_t)}_{a_j} \le y_j \le \underbrace{\beta_j - L + \sum_{t=1}^{n_{\tilde{e}}} \max(\gamma_{jt}\underline{c}_t, \gamma_{jt}\bar{c}_t)}_{b_j}$$

Recall that we choose L such that all $0 \le y_j,\ j = 1,\dots,n$. Hence, we can conclude that $0 \le a_j < b_j$ and hence, $a_j^p < b_j^p$. Therefore,

$$a_j^p \le y_j^p \le b_j^p \Rightarrow \underbrace{\sum_{j=1}^n a_j^p}_{a} \le \underbrace{\sum_{j=1}^n y_j^p}_{x} \le \underbrace{\sum_{j=1}^n b_j^p}_{b} \tag{4.18}$$

with $a < b$. Recall that the error caused by Inequality (ii) in (4.4) approaches 0 as $p \to \infty$. This suggests that in order to get a good approximation, p should be selected very large. However, since in our case both a and b depend on p (as shown in (4.18)), we need a more careful investigation to study the effect of $p \to \infty$ on $e_{\text{max,rel}}(a,b)$ and $e_{\text{max,abs}}(a,b)$. To this end, let $\alpha = \max_{j=1,\dots,n}(a_j)$ and $\beta = \max_{j=1,\dots,n} b_j$. Denote the number of $a_j,\ j = 1,\dots,n$ that are equal to α by A and the number of $b_j,\ j = 1,\dots,n$ that are equal to β by B. Since $b > a$, we conclude that $\beta > \alpha$. Now, for a large p, we can rewrite a and b as $a \approx A\alpha^p$ and $b \approx B\beta^p$. Using this notation, we obtain the following proposition.

Proposition 4.3.3 *Considering our assumption that p is a positive even integer and that $a \approx A\alpha^p,\ b \approx B\beta^p$ for a large p with A, B positive integers and $0 \le \alpha < \beta$, we have the following results:*

$$\textit{if } (\alpha = 0 \textit{ and } \beta > 0) : \begin{cases} \lim_{p\to\infty} e_{\text{max,rel}}(\alpha,\beta) = \infty \\ \lim_{p\to\infty} e_{\text{max,abs}}(\alpha,\beta) = \beta \end{cases}$$

$$\textit{if } (\alpha > 0 \textit{ and } \beta > 0) : \begin{cases} \lim_{p\to\infty} e_{\text{max,rel}}(\alpha,\beta) = \frac{\beta}{\alpha} - 1 \\ \lim_{p\to\infty} e_{\text{max,abs}}(\alpha,\beta) = \beta \end{cases}$$

Proof : First let $\alpha = 0$ and thus, $\beta > 0$. Now, by replacing a and b in (4.16) by $A\alpha^p$ and $B\beta^p$ respectively, we will be in the case of Preposition 4.3.2 and so,

$$\lim_{p\to\infty} e_{\text{max,rel}}(\alpha,\beta) = \infty.$$

However, for the absolute error, we obtain the following result after replacing a and b in (4.17) by $A\alpha^p$ and $B\beta^p$ respectively:

$$\lim_{p\to\infty} e_{\text{max,abs}}(\alpha,\beta) = \lim_{p\to\infty} \Big(B\beta^p\Big)^{\frac{1}{p}} \left[\Big(\frac{1}{p}\Big)^{\frac{1}{p-1}} - \Big(\frac{1}{p}\Big)^{\frac{p}{p-1}}\right]$$

$$\begin{aligned}&= \beta[1-0]\\ &= \beta\end{aligned}$$

in which the limit of $\left(\frac{1}{p}\right)^{\frac{1}{p-1}}$ is computed as follows. Since $\lim_{p\to\infty}\left(\dfrac{1}{p}\right)^{\frac{1}{p-1}}$ is of the form 0^0, it is an indeterminate limit and therefore, we use l'Hopital's rule. Let:

$$\begin{aligned}\lim_{p\to\infty}\left(\frac{1}{p}\right)^{\frac{1}{p-1}} = M \quad \Rightarrow \quad \ln M &= \lim_{p\to\infty}\frac{-1}{p-1}\ln p = \frac{\infty}{\infty}\\ &\overset{\text{l'Hopital}}{=} \lim_{p\to\infty}\frac{-\frac{1}{p}}{1} = 0\\ &\Rightarrow M = 1 \quad \Rightarrow \quad \lim_{p\to\infty}\left(\frac{1}{p}\right)^{\frac{1}{p-1}} = 1.\end{aligned}$$

Now let $\alpha > 0$ and $\beta > 0$. Note that since $\alpha < \beta$ if $p \to \infty$, we have $\alpha^p \ll \beta^p$. Moreover, in the nominator and denominator of both $e_{\text{max,rel}}(\alpha,\beta)$ in (4.16) and $e_{\text{max,abs}}(\alpha,\beta)$ in (4.17), the sums of the powers of α and β are equal in each expression. Therefore, to compute the limit of $e_{\text{max,rel}}(\alpha,\beta)$ and $e_{\text{max,abs}}(\alpha,\beta)$, we can conclude that term in which β has the largest power in each expression in the nominator or denominator is the most influential one and hence, will determine the limit value. Based on this argument, the limit of $e_{\text{max,rel}}(\alpha,\beta)$ when $p\to\infty$ can be obtained as follows.

$$\begin{aligned}\lim_{p\to\infty} e_{\text{max,rel}}(\alpha,\beta) &= \lim_{p\to\infty}\left[\left(\frac{-\frac{1}{p}(A\alpha^p)^{\frac{1}{p}}(B\beta^p)^2}{\frac{-(p-1)}{p}(B\beta^p)^{\frac{1}{p}+1}}\right)^{\frac{1}{p}} \cdot \frac{\frac{-(p-1)}{p}(B\beta^p)^{\frac{1}{p}+1}}{-(A\alpha^p)^{\frac{1}{p}}(B\beta^p)^{\frac{1}{p}+1}} - 1\right]\\ &= \lim_{p\to\infty}\left[\left(\frac{A^{\frac{1}{p}}\alpha B^2\beta^{p-1}}{(p-1)B^{\frac{1}{p}+1}}\right)^{\frac{1}{p}}\ \frac{p-1}{pA^{\frac{1}{p}}\alpha} - 1\right]\\ &= \lim_{p\to\infty}\left[\frac{(p-1)A^{\frac{1}{p^2}-\frac{1}{p}}B^{\frac{1}{p}-\frac{1}{p^2}}\beta^{1-\frac{1}{p}}}{p(p-1)^{\frac{1}{p}}\alpha} - 1\right]\\ &= \frac{\beta}{\alpha} - 1\end{aligned}$$

in which we used the fact that $\lim_{p\to\infty} x^{\frac{1}{p}} = 1$ and $\lim_{p\to\infty}(\frac{1}{p-1})^{\frac{1}{p}} = 1$ for any $x > 0$, as shown before.

The limit of $e_{\text{max,abs}}(\alpha,\beta)$ when $p\to\infty$ will be

$$\begin{aligned}\lim_{p\to\infty} e_{\text{max,abs}}(\alpha,\beta) &= \lim_{p\to\infty}\left[\left(\frac{1}{p}\right)^{\frac{1}{p-1}}\left(B\beta^p\right)^{\frac{1}{p}} - \left(\frac{1}{B\beta^p}\left(\frac{1}{p}\right)^{\frac{p}{p-1}}\left(B\beta^p\right)^{\frac{1+p}{p}}\right)\right]\\ &= \lim_{p\to\infty}\left[\left(\frac{1}{p}\right)^{\frac{1}{p-1}}B^{\frac{1}{p}}\beta - \left(\frac{1}{p}\right)^{\frac{p}{p-1}}B^{\frac{1}{p}}\beta\right]\end{aligned}$$

$$
\begin{aligned}
&= 1 \times \beta - 0 \\
&= \beta.
\end{aligned}
$$

using the fact that $\lim_{p\to\infty}\left(\frac{1}{p}\right)^{\frac{p}{p-1}} = 0$ since it is of the form 0^1. □

This proposition shows that if $p \to \infty$, then based on the values of β and α, the size of the relative and absolute errors can change. Moreover, experiments show that as p increases, both $e_{\text{max,rel}}(\alpha,\beta)$ and $e_{\text{max,abs}}(\alpha,\beta)$ increase *monotonically* until they converge to the values given in Proposition 4.3.3. This indicates that we cannot choose a very large p since the error will become too large. However, as mentioned before, the error caused by Inequality (ii) in (4.4) approaches 0 as $p \to \infty$. Consequently, there is a trade-off between having a small error in Inequality (iii) by choosing a relatively small p and having a small error in Inequality (ii) by choosing a very large p. So, the value of p had to be tuned accordingly.

4.4 Convexity of the Approximation

Recall that in the definition of the expected value of max-plus-scaling functions in Chapter 3, (3.10) is affine in the control variable $\tilde{u}(k)$ and (3.24) is affine in the control variables $\hat{\theta}$ and $\hat{\lambda}$. In this section, we prove that the approximation function (cf. (4.6))

$$
\mathfrak{U}\Big(\mathbb{E}[\max_{j=1,\ldots,n}(\beta_j - L + \gamma_j^T\tilde{e})]\Big) = \Big(\sum_{j=1}^{n}\mathbb{E}\big[(\beta_j - L + \gamma_j^T\tilde{e})^p\big]\Big)^{1/p} + L
$$

with $\beta_j = a_j + b_j^T w$, where w denotes a general control variable, is convex in w. To this end, let $\phi_j(w) = a_j + b_j^T w + \gamma_j^T\tilde{e} - L$, which is an affine and so a convex function in w. Hence, the p-th moment of $\phi_j(w)$, i.e., $\mathbb{E}[\big(\phi_j(w)\big)^p]$ can be defined as:

$$
\mathbb{E}[\big(\phi_j(w)\big)^p] = \int_{-\infty}^{+\infty}\ldots\int_{-\infty}^{+\infty}\big(\phi_j(w)\big)^p\, f(\tilde{e})d\tilde{e}
$$

where $f(\tilde{e})$ is the probability density function of $\tilde{e}$. Note that to the random variable $\tilde{e}$, we only assign distributions that have finite moments; hence, $\mathbb{E}[\big(\phi_j(w)\big)^p]$ is finite as well.

To prove the convexity of $\mathfrak{U}\Big(\mathbb{E}[\max_{j=1,\ldots,n}\phi_j(w)]\Big)$ in w, we need the following theorems [78, Chapter 5]:

Theorem 4.4.1 (Minkowski inequality for functions) *Let $h : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ be such that the functions $|h(x)|^\ell$ and $|g(x)|^\ell$, for an integer $\ell > 1$, are integrable*[3] *on $\mathbb{R}$. Then*

$$\Big\{\mathbb{E}\big[|h(x)+g(x)|^\ell\big]\Big\}^{1/\ell} \le \Big\{\mathbb{E}\big[|h(x)|^\ell\big]\Big\}^{1/\ell} + \Big\{\mathbb{E}\big[|g(x)|^\ell\big]\Big\}^{1/\ell}$$

Theorem 4.4.2 (Minkowski inequality for vectors) *Let $x = [x_1, \ldots, x_n]^T$ and $y = [y_1, \ldots, y_n]^T$ be two vectors in $\mathbb{R}^n$ and $\ell > 1$ be an integer. Then,*

$$\left(\sum_{j=1}^{n} |x_j + y_j|^\ell\right)^{1/\ell} \le \left(\sum_{j=1}^{n} |x_j|^\ell\right)^{1/\ell} + \left(\sum_{j=1}^{n} |y_j|^\ell\right)^{1/\ell}.$$

Recall that (cf. Remark 4.2.5) we assume $p = 2q,\ q \in \mathbb{Z}^+$; hence, $|x|^p = x^p$. Consequently, we drop the absolute value sign for the expressions with the power p in the rest of this section. First, we prove the following proposition:

Proposition 4.4.3 $\big(\mathbb{E}\big[\big(\phi_j(w)\big)^p\big]\big)^{1/p}$*, with ϕ_j an affine function in w, is a convex function of w.*

Proof : If we show that

$$\begin{aligned}&\Big\{\mathbb{E}\big[\big(\phi_j(\lambda w_1 + (1-\lambda)w_2)\big)^p\big]\Big\}^{1/p}\\ &\le \lambda\Big\{\mathbb{E}[\big(\phi_j(w_1)\big)^p]\Big\}^{1/p} + (1-\lambda)\Big\{\mathbb{E}[\big(\phi_j(w_2)\big)^p]\Big\}^{1/p}\end{aligned}$$

for any two points w_1 and w_2 in the domain of $\mathbb{E}\big[\big(\phi_j(w)\big)^p\big]$ and for any $0 \le \lambda \le 1$, then the proof is complete. Since ϕ_j is an affine function in w, we have

$$\phi_j(\lambda w_1 + (1-\lambda)w_2) = \lambda\phi_j(w_1) + (1-\lambda)\phi_j(w_2)$$

Therefore, from the Minkowski inequality for functions (cf. Theorem 4.4.1) and keeping in mind that p is an even integer we obtain:

$$\begin{aligned}&\Big(\mathbb{E}\big[\big(\phi_j(\lambda w_1 + (1-\lambda)w_2)\big)^p\big]\Big)^{1/p}\\ &= \Big(\mathbb{E}\big[\big(\underbrace{\lambda\phi_j(w_1)}_{h(x)} + \underbrace{(1-\lambda)\phi_j(w_2)}_{g(x)}\big)^p\big]\Big)^{1/p}\\ &\le \lambda\Big(\mathbb{E}[\big(\phi_j(w_1)\big)^p]\Big)^{1/p} + (1-\lambda)\Big(\mathbb{E}[\big(\phi_j(w_2)\big)^p]\Big)^{1/p} \qquad (4.19)\end{aligned}$$

So the inequality holds true and consequently, $\big(\mathbb{E}\big[\big(\phi_j(w)\big)^p\big]\big)^{1/p}$ is a convex function in w. □

[3] If for a domain D the integral $\int_D f(x)\,dx$ exists, then the function f is called integrable on D.

Now for a shorter notation, let

$$\mathbb{F}(w) = \mathfrak{U}\Big(\mathbb{E}[\max_{j=1,\ldots,n} \phi_j(w)]\Big) = \left(\sum_{j=1}^{n} \mathbb{E}[\big(\phi_j(w)\big)^p]\right)^{1/p} + L. \tag{4.20}$$

Remark 4.4.4 Let $x = [x_1, \ldots, x_n]^T$ and $y = [y_1, \ldots, y_n]^T$ be two vectors in $\mathbb{R}^n$ and $p \geq 2$ an even integer. If $|x_j| < |y_j|$ for $j = 1, \ldots, n$, then it is easy to verify that

$$\left(\sum_{j=1}^{n} |x_j|^p\right)^{1/p} < \left(\sum_{j=1}^{n} |y_j|^p\right)^{1/p}.$$

□

Considering (4.20), we prove the convexity of $\mathbb{F}(w)$ in w in the following proposition.

Proposition 4.4.5 $\mathbb{F}(w)$ *is a convex function of* w.

Proof : Note that in this proof the constant value L in $\mathbb{F}(w)$ is omitted since this term does not influence the convexity. Now, we prove that for any $w_1,\ w_2$ in the domain of $\mathbb{F}(w)$ and for any $\lambda \in [0, 1]$ the following inequality holds:

$$\mathbb{F}(\lambda w_1 + (1-\lambda)w_2) \leq \lambda \mathbb{F}(w_1) + (1-\lambda)\mathbb{F}(w_2)$$

and therefore, it is a convex function of w. We have:

$$\begin{aligned}
&\mathbb{F}(\lambda w_1 + (1-\lambda)w_2) \\
&= \left(\sum_{j=1}^{n} \mathbb{E}\big[\big(\phi_j(\lambda w_1 + (1-\lambda)w_2)\big)^p\big]\right)^{1/p} \\
&= \left(\sum_{j=1}^{n} \mathbb{E}\big[\big(\lambda\phi_j(w_1) + (1-\lambda)\phi_j(w_2)\big)^p\big]\right)^{1/p} \qquad \text{since } \phi_j \text{ is affine in } w \\
&= \left(\sum_{j=1}^{n} \Big(\mathbb{E}\big[\big(\lambda\phi_j(w_1) + (1-\lambda)\phi_j(w_2)\big)^p\big]^{1/p}\Big)^p\right)^{1/p} \\
&\overset{(i)}{\leq} \left(\sum_{j=1}^{n} \Big(\underbrace{\lambda\left(\mathbb{E}[\big(\phi_j(w_1)\big)^p]\right)^{1/p} + (1-\lambda)\left(\mathbb{E}[\big(\phi_j(w_2)\big)^p]\right)^{1/p}}_{x_j+y_j}\Big)^p\right)^{1/p}
\end{aligned}$$

$$\overset{(ii)}{\leq} \left(\sum_{j=1}^{n} \Big(\underbrace{\lambda \left(\mathbb{E}[(\phi_j(w_1))^p] \right)^{1/p}}_{x_j} \Big)^p \right)^{1/p} + \left(\sum_{j=1}^{n} \Big(\underbrace{(1-\lambda) \left(\mathbb{E}[(\phi_j(w_2))^p] \right)^{1/p}}_{y_j} \Big)^p \right)^{1/p}$$

$$\leq \lambda \left(\sum_{j=1}^{n} \mathbb{E}[(\phi_j(w_1))^p] \right)^{1/p} + (1-\lambda) \left(\sum_{j=1}^{n} \mathbb{E}[(\phi_j(w_2))^p] \right)^{1/p}$$

$$\leq \lambda \mathbb{F}(w_1) + (1-\lambda)\mathbb{F}(w_2)$$

where (i) is due to (4.19) in combination with Remark 4.4.4, and (ii) is due to Theorem 4.4.2. □

Since $\mathbb{F}(w)$ is a convex function of w, we can also compute its subgradient with respect to w as follows:

$$\begin{aligned}
\frac{\partial}{\partial w}\mathbb{F}(w) &= \frac{\partial}{\partial w} \left(\sum_{j=1}^{n} \mathbb{E}\big[(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^p \big] \right)^{1/p} \\
&= \frac{1}{p} \left(\sum_{j=1}^{n} \mathbb{E}\big[(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^p \big] \right)^{1/p-1} \\
&\quad \cdot \frac{\partial}{\partial w} \left(\sum_{j=1}^{n} \mathbb{E}\big[(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^p \big] \right) \\
&= \frac{1}{p} \left(\sum_{j=1}^{n} \mathbb{E}\big[(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^p \big] \right)^{1/p-1} \\
&\quad \cdot \sum_{j=1}^{n} \frac{\partial}{\partial w} \mathbb{E}\big[(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^p \big] \\
&= \left(\sum_{j=1}^{n} \mathbb{E}\big[(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^p \big] \right)^{1/p-1} \qquad (4.21) \\
&\quad \cdot \sum_{j=1}^{n} b_j \mathbb{E}\big[(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^{p-1} \big]
\end{aligned}$$

where in the last step, the derivative of the expected value is computed as follows,

$$\frac{\partial}{\partial w} \mathbb{E}\big[(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^p \big]$$

$$
\begin{aligned}
&= \frac{\partial}{\partial w}\Big[\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^p f(\tilde{e}) d\tilde{e}\Big]\\
&\overset{(*)}{=} \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\frac{\partial}{\partial w}(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^p f(\tilde{e}) d\tilde{e}\\
&= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} p(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^{p-1} b_j f(\tilde{e}) d\tilde{e}\\
&= pb_j \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(a_j + b_j^T w + \gamma_j^T \tilde{e} - L)^{p-1} f(\tilde{e}) d\tilde{e}\\
&= pb_j \mathbb{E}\big[a_j + b_j^T w + \gamma_j^T \tilde{e} - L\big]^{p-1}
\end{aligned}
$$

Note that in $(*)$ due to the continuity of the integrand and its derivatives, the Leibniz rule is used for differentiating the integral. For more details the interested reader is referred to [41].

4.5 Summary

In this chapter, we have introduced an approximation method in order to compute the expected value of a max-plus-scaling function, which is defined as the maximum of several affine expressions. This method is based on higher-order moments of a random variable and we have applied it under the assumption of having independent random variables. This method is applicable to any distribution with finite moments for which a closed-form exists. As such, it involves no analytic or numerical integration. Accordingly, using closed-form moments, we obtain an analytic solution the complexity of which is much less than numerical integration, analytic integration, or even the approximation method based on variability expansion discussed in Section 3.1.3.

We have also shown that the error caused by using this approximation method is bounded from above and hence, cannot exceed a certain value. Moreover, we have discussed the behavior of the upper bound of the error in the cases in which each of its parameters takes different values, such as 0 or ∞.

The last section of this chapter has been dedicated to the convexity analysis of the approximation function. We have proved that the approximation function is convex and hence, its subgradient can be obtained accordingly. The convexity of the approximation function is useful in the computation of the objective function in the MPC and identification optimization problem for MPL systems.

Chapter 5

Approximation Approach for Model Predictive Control and Identification of Stochastic Max-Plus-Linear Systems

In the previous chapter, we have proposed an approximation method to compute the expected value of a stochastic max-plus-scaling function that results in an analytic solution if we choose probability distributions that have finite moments for which a closed-form expression exists. Now, this method will be applied to MPC and identification of stochastic MPL systems. Using this approximation method for the mentioned problems will decrease the computational complexity and the computation time.

5.1 Approximate Stochastic MPL-MPC

In Section 3.1 of Chapter 3, MPC has been proposed as a control approach for stochastic MPL systems. Based on the MPC procedure, an optimization problem has to be solved in order to obtain an optimal input sequence that minimizes the objective function. Since we deal with stochastic MPL systems, this optimization problem is characterized by a significant computational complexity, which we try to decrease by applying the approximation method of the previous chapter.

5.1.1 Problem Statement

In the stochastic MPL-MPC optimization problem (3.8), the objective function is defined as (cf. (3.1))

$$J(k) = J_{\text{out}}(k) + \lambda J_{\text{in}}(k) \tag{5.1}$$

where the output objective function consists of an expected value of stochastic max-plus-scaling functions (cf. (3.4)). Therefore, in order to solve (3.8), this expected value needs to be computed. Recall from Section 3.1.2 that the stochastic max-plus-scaling function appearing in the MPL-MPC problem has the following general form:

$$v(k) = \max_{j=1,\dots,n_v} (\alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}(k)) \tag{5.2}$$

where $\tilde{e}$ is a vector of independent random variables with the given probability density function $f(\cdot)$. Hence, we need to compute

$$\mathbb{E}[v(k)] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \max_{j=1,\dots,n_v} (\alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}(k)) f(\tilde{e}) d\tilde{e}. \tag{5.3}$$

To this end, several solution approaches have been proposed (cf. Section 3.1.3); however, each of these methods has a high level of computational complexity. Therefore, we apply the approximation method of Chapter 4 in order to simplify the problem and to increase the computational efficiency.

5.1.2 Approximation Approach

Considering (4.8) of Section 4.2, we can approximate $\mathbb{E}[v(k)],\ k = 1, \dots, N_{\text{p}} - 1$ by

$$\begin{aligned}\mathfrak{U}\Big(\mathbb{E}[v(k)]\Big)\\ &= \Bigg(\sum_{j=1}^{n_v} \sum_{k_0+k_1+\dots+k_{n_{\tilde{e}}}=p} \frac{p!}{k_0!\, k_2! \cdots k_{n_{\tilde{e}}}!} \gamma_{j,0}^{k_0} \prod_{t=1}^{n_{\tilde{e}}} \gamma_{j,t}^{k_t} \mathbb{E}\big[\tilde{e}_t^{k_t}(k)\big]\Bigg)^{1/p} + L\end{aligned} \tag{5.4}$$

where for each $j \in \{1, \dots, n_v\}$ we have $\gamma_{j,0} = \alpha_j(k) + \beta_j^T \tilde{u}(k) - L$ and $\gamma_{j,t}$ are scalars for $t = 1, \dots, n_{\tilde{e}}$. Note that (5.4) can be used for any probability distribution assigned to $\tilde{e}_1, \dots, \tilde{e}_{n_{\tilde{e}}}$, and it is only useful to be applied when the probability distribution has finite moments and a closed form of these moments exists. For the case that we have probability distributions that are preserved under summation, we can use the following expression in order to approximate $\mathbb{E}[v(k)]$ (cf. (4.6)):

$$\mathfrak{U}\Big(\mathbb{E}[v(k)]\Big) = \Bigg(\sum_{j=1}^{n_v} \mathbb{E}\big[(\alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}(k) - L)^p\big]\Bigg)^{1/p} + L \tag{5.5}$$

Using this expression directly is computationally much faster than using (5.4). The reason is that since we already know the distribution of each random variable $\alpha_j(k) + \beta_j^T \tilde{u}(k) + \gamma_j^T \tilde{e}(k) - L$, we can directly compute its p-th moment without first having to expand the p-th power expression, as is done in (5.4).

As shown in Section 4.4, both (5.4) and (5.5) are convex in $\tilde{u}$, and their subgradient with respect to $\tilde{u}$ can be obtained as follows (cf. (4.21)). For (5.4), the subgradient is

$$\begin{aligned}\nabla_{\tilde{u}}\mathfrak{U}\Big(\mathbb{E}[v(k)]\Big) = \Big(\sum_{j=1}^{n} \sum_{k_0+k_1+\cdots+k_{n_{\tilde{e}}}=p} \frac{p!}{k_0!\,k_2!\cdots k_{n_{\tilde{e}}}!}\gamma_{j,0}^{k_0}\prod_{t=1}^{n_{\tilde{e}}}\gamma_{j,t}^{k_t}\mathbb{E}\big[\tilde{e}_t^{k_t}(k)\big]\Big)^{1/p-1} \cdot \\ \Big(\sum_{j=1}^{n_v}\beta_j \sum_{k_0+k_1+\cdots+k_{n_e}=p-1} \frac{(p-1)!}{k_0!\,k_2!\cdots k_{n_e}!} k_0 \gamma_{j,0}^{k_0-1}\prod_{t=1}^{n_e}\gamma_{j,t}^{k_t}\mathbb{E}\big[\tilde{e}_t^{k_t}(k)\big]\Big)\end{aligned} \tag{5.6}$$

and for (5.5), it is

$$\begin{aligned}\nabla_{\tilde{u}}\mathfrak{U}\Big(\mathbb{E}[v(k)]\Big) &= \left(\sum_{j=1}^{n_v}\mathbb{E}\big[(\alpha_j + \beta_j^T\tilde{u}(k) + \gamma_j^T\tilde{e}(k) - L)^p\big]\right)^{1/p-1} \cdot \\ &\quad \left(\sum_{j=1}^{n_v}\beta_j\mathbb{E}\big[(\alpha_j + \beta_j^T\tilde{u}(k) + \gamma_j^T\tilde{e}(k) - L)^{p-1}\big]\right)\end{aligned} \tag{5.7}$$

Now that we can approximate the expected value of a stochastic max-plus-scaling function $v(k)$, we can also compute the objective function (5.1) approximately as follows. Recall that $J_{\text{out}}(k)$ is defined as a sum over $\mathbb{E}[\max(\tilde{y}(k) - \tilde{r}(k), 0)]$. Hence, we obtain the following approximate objective function:

$$\begin{aligned}J_{\text{app}}(k) &= J_{\text{out,app}}(k) + \lambda J_{\text{in}}(k) \\ &= \sum_{j=1}^{N_{\text{p}}\cdot n_y}\Big(\mathfrak{U}\Big(\mathbb{E}[\max(\tilde{y}(k) - \tilde{r}(k), 0)]\Big)\Big)_j - \lambda\sum_{l=1}^{N_{\text{p}}\cdot n_u}\tilde{u}_l(k)\end{aligned} \tag{5.8}$$

and hence, instead of solving the optimization problem (3.8), we can solve the following approximate optimization problem:

$$\begin{aligned} &\min_{\tilde{u}(k)} \quad J_{\text{app}}(k) \\ \text{s.t.}\quad &\Delta u(k+j) \geq 0 \quad \text{for } j = 0, \ldots, N_{\text{p}} - 1 \\ &\Delta^2 u(k+j) = 0 \quad \text{for } j = N_{\text{c}}, \ldots, N_{\text{p}} - 1 \\ &A_{\text{con}}(k)\tilde{u}(k) + B_{\text{con}}(k)\mathfrak{U}\Big(\mathbb{E}[\tilde{y}(k)]\Big) \leq c_{\text{con}}(k)\end{aligned} \tag{5.9}$$

Recall that $\mathfrak{U}\Big(\mathbb{E}[\tilde{y}(k)]\Big)$ is an upper bound for $\mathbb{E}[\tilde{y}(k)]$. Hence, if the entries of $B_{\text{con}}(k)$ are nonnegative, then $A_{\text{con}}(k)\tilde{u}(k) + B_{\text{con}}(k)\mathfrak{U}\Big(\mathbb{E}[\tilde{y}(k)]\Big) \leq c_{\text{con}}(k)$ also implies $A_{\text{con}}(k)\tilde{u}(k) + B_{\text{con}}(k)\mathbb{E}[\tilde{y}(k)] \leq c_{\text{con}}(k)$. Moreover, the minimization of $J_{\text{app}}(k)$ (in general) also results in minimization of $J(k)$ since $J_{\text{app}}(k)$ is an upper bound for $J(k)$.

Note that convexity of the approximation function $\mathfrak{U}\Big(\mathbb{E}[v(k)]\Big)$ in $\tilde{u}$ implies that the last inequality constraint in (5.9) is also convex in $\tilde{u}$. Hence, if we assume that all the entries of B_{con} are nonnegative, in that case, the whole approximate MPL-MPC optimization problem (5.9) turns out to be convex in $\tilde{u}$, and having a closed form expression for subgradients (cf. (5.6) and (5.7)), we can solve this optimization problem efficiently using gradient-based convex optimization algorithms such as the cutting-plane method or the ellipsoid algorithm [15].

5.1.3 Example

We consider an example similar to the one of [104] and we study it for the cases in which the noise vector is uniformly distributed and normally distributed. For each case, we apply different methods to compute the objective function (3.8), namely, the analytic integration of Section 3.1.3, numerical integration[1], Monte Carlo simulation, and the approximation method of Section 4.2. Afterwards, we compare the performance of the MPC controller using these methods.

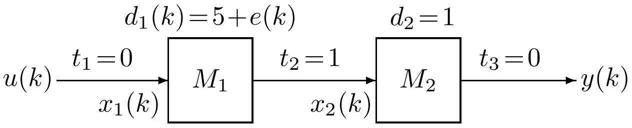


Figure 5.1: A production system.

Consider the simple manufacturing system of Figure 5.1. This system consists of two machines, M_1 and M_2, and operates in batches. The raw material is fed to machine M_1 where preprocessing is done. Afterwards the intermediate product is fed to machine M_2 and finally the complete product leaves the system. We assume that each machine starts working as soon as possible on each batch, i.e., as soon as the raw material or the required intermediate product is available, and as soon as the machine is idle (i.e., the previous batch of products has been processed and has left the machine). Define:

$u(k)$: time instant at which the raw material is fed to the system for the k-th time

[1] The numerical integration is based on Monte Carlo integration [25]

$y(k)$: time instant at which the k-th product leaves the system
$x_i(k)$: time instant at which machine i starts for the k-th time
$t_j(k)$: transportation time on link j for the k-th batch
$d_i(k)$: processing time on machine i for the k-th batch

The system equations are given by

$$\begin{aligned} x_1(k) &= \max(x_1(k-1) + d_1(k-1), u(k) + t_1(k)) \\ x_2(k) &= \max(x_1(k) + d_1(k) + t_2(k), x_2(k-1) + d_2(k-1)) \\ &= \max(x_1(k-1) + d_1(k-1) + d_1(k) + t_2(k), \\ &\qquad u(k) + d_1(k) + t_1(k) + t_2(k), x_2(k-1) + d_2(k-1)) \\ y(k) &= x_2(k) + d_2(k) + t_3(k) \end{aligned}$$

and in matrix notation this becomes

$$\begin{aligned} x(k) &= A(k) \otimes x(k-1) \oplus B(k) \otimes u(k) \\ y(k) &= C(k) \otimes x(k) \ . \end{aligned}$$

where the system matrices A, B and C are given as follows:

$$A(k) = \begin{bmatrix} d_1(k-1) & \varepsilon \\ d_1(k-1) + d_1(k) + 1 & 1 \end{bmatrix} , \ B(k) = \begin{bmatrix} 0 \\ d_1(k) + 1 \end{bmatrix} ,$$

$$C(k) = \begin{bmatrix} \varepsilon & 1 \end{bmatrix} .$$

The objective functions (3.4)-(3.5) will be optimized for $N_\mathrm{p} = 3$, $N_\mathrm{c} = 2$, and $\lambda = 0.05$. It is assumed that the transportation times are constant: $t_1(k) = 0, t_2(k) = 1, t_3(k) = 0$; the production time of M_2 is constant: $d_2(k) = 1$; the due date (reference) signal is $r(k) = 4+6\cdot k$; and $x(0) = [0 \ \ 7]^T$. The signal d_1 is assumed to be corrupted by noise: $d_1(k+\ell) = 5+e(k+\ell)$ where $e(k+\ell), \ \ell = -1, \ldots, N_\mathrm{p}-1$ are random variables with a given probability distribution. Hence, the vector $\tilde{e}(k) = [\ d_1(k-1) \ \ \ldots \ \ d_1(k+N_\mathrm{p}-1) \ ]^T$ consists of independent stochastic variables. Now, we can rewrite $\max(y(k+\ell) - r(k+\ell), 0), \ \ell = 1, \ldots, N_\mathrm{p}-1$ such that it is divided into deterministic and stochastic parts. This yields expressions of the following form[2]:

$$\begin{aligned} \max(y(k) - r(k), 0) &= \max(\eta_1 + \tilde{e}_1 + \tilde{e}_2, \eta_2 + \tilde{e}_2, \eta_3, 0) \\ \max(y(k+1) - r(k+1), 0) &= \max(\eta_4 + \tilde{e}_1 + \tilde{e}_2 + \tilde{e}_3, \eta_5 + \tilde{e}_2 + \tilde{e}_3, \\ &\qquad \eta_6 + \tilde{e}_3, \eta_7 + \tilde{e}_1 + \tilde{e}_2, \eta_8 + \tilde{e}_2, \eta_9, 0) \end{aligned} \tag{5.10}$$

[2] We have omitted the argument k for brevity.

$$\begin{aligned}\max(y(k+2)-r(k+2),0) = \max(&\eta_{10}+\tilde{e}_1+\tilde{e}_2+\tilde{e}_3+\tilde{e}_4,\\ &\eta_{11}+\tilde{e}_2+\tilde{e}_3+\tilde{e}_4,\eta_{12}+\tilde{e}_3+\tilde{e}_4,\eta_{13}+\tilde{e}_4,\\ &\eta_{14}+\tilde{e}_1+\tilde{e}_2+\tilde{e}_3,\eta_{15}+\tilde{e}_2+\tilde{e}_3,\eta_{16}+\tilde{e}_3,\\ &\eta_{17}+\tilde{e}_1+\tilde{e}_2,\eta_{18}+\tilde{e}_2,\eta_{19},0)\end{aligned}$$

where $\eta_1,\ldots,\eta_{19}$ are sums of deterministic values and $\tilde{e}_1,\ldots,\tilde{e}_4$ are the entries of $\tilde{e}$.

Now, we study separately the two cases where the entries of $\tilde{e}$ are uniformly and normally distributed, respectively.

Uniform distribution

In this case, we assume that $\tilde{e}_1,\ldots,\tilde{e}_4$ are uniformly distributed, i.e., $\tilde{e}_i \sim \mathcal{U}(-1,1)$, $i=1,\ldots,4$. As mentioned in Section 4.2, we define the scalars L_1, L_2, and L_3 for the three expressions in (5.10), as follows:

$$\begin{aligned}L_1 &= \min(\eta_1-2,\eta_2-1,\eta_3,0)\\ L_2 &= \min(\eta_4-3,\eta_5-2,\eta_6-1,\eta_7-2,\eta_8-1,\eta_9,0)\\ L_3 &= \min(\eta_{10}-4,\eta_{11}-3,\eta_{12}-2,\eta_{13}-1,\eta_{14}-3,\\ &\qquad \eta_{15}-2,\eta_{16}-1,\eta_{17}-2,\eta_{18}-1,\eta_{19},0)\end{aligned} \tag{5.11}$$

where the numbers $-1,-2,-3$, and -4 are the lower bounds of $\gamma_j^T\tilde{e}$ where γ_j is a vector of 0s and 1s that indicates which elements of $\tilde{e}$ appear in the maximization.

Now, we compute the optimal input sequence by solving the MPC problem in closed-loop for $k=1,\ldots,40$. Figure 5.2 shows the difference between the resulting output signal y and the due date signal r. The optimization has been done using *fmincon* optimizer in Matlab with one initial value since the objective functions (both the exact and the approximate one) are convex and hence, the global minimum can be reached. In this figure, the “Exact solution” is obtained by using analytic integration of Section 3.1.3 to compute the expected value appearing in the objective function. The “Nominal MPC” is indeed obtained by ignoring the effect of noise while computing the optimal input sequence. As a result, the due dates will be most of the time violated in this case and we have late deliveries. This is due to the fact that we compute the optimal input in the absence of noise and hence we cannot minimize its effect. Finally, the “Approximation” is obtained by using $J_{\text{app}}(k)$ defined in (5.8). We have chosen different values of p to find out which one gives the closest result to the exact solution. As can be seen, for $p=20,30$, and 40 the result of the closed-loop controlled system using the approximation method is quite close to the one using the exact solution. Note that we have also computed the expected value in the objective function using numerical integration and Monte Carlo simulation. However, since the plot of the difference between the output signal y and the due date signal r using these two methods matches exactly the one using analytic integration, we have not plotted them in Figure 5.2.

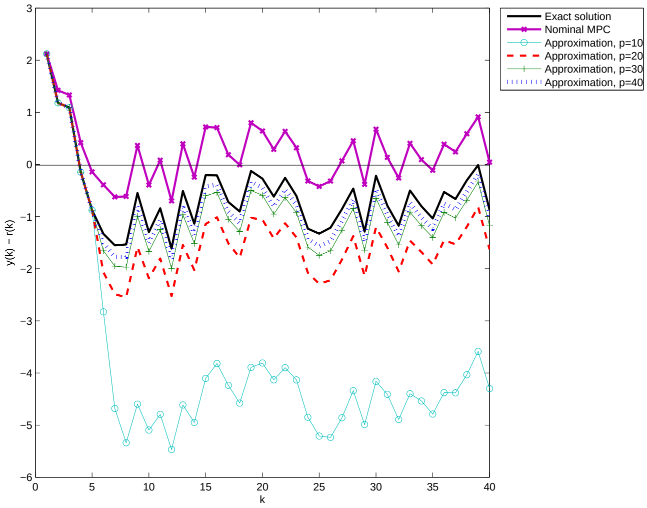


Figure 5.2: Due date error $y(k) - r(k)$ for the closed-loop system using nominal MPC, analytic integration to compute the expected value in the objective function, and the approximate objective function $J_{\text{app}}(k)$ in (5.8).

Table 5.1 reports the total optimization time[3] for the closed-loop simulation over 40 event steps using analytic integration (i.e., the exact solution), nominal MPC, numerical integration, Monte Carlo simulation, and the approximation method for different values of p. Moreover, for each method, the objective function over the entire simulation period, i.e., $J_{\text{tot}} = \sum_{k=1}^{40} (\max(y(k) - r(k), 0) - \lambda u(k))$, has been computed using 20 different noise realizations and then, its mean value is reported in the table. In addition, the relative error[4] of the mean value of J_{tot} using the analytic integration versus using the other methods are presented in this table.

Despite the fact that nominal MPC is extremely fast, the mean value of J_{tot} is considerably larger than the one using the analytic integration method to compute the objective function. This is due to ignoring the effect of the noise while computing the optimal input sequence. Hence, just applying nominal MPC is not

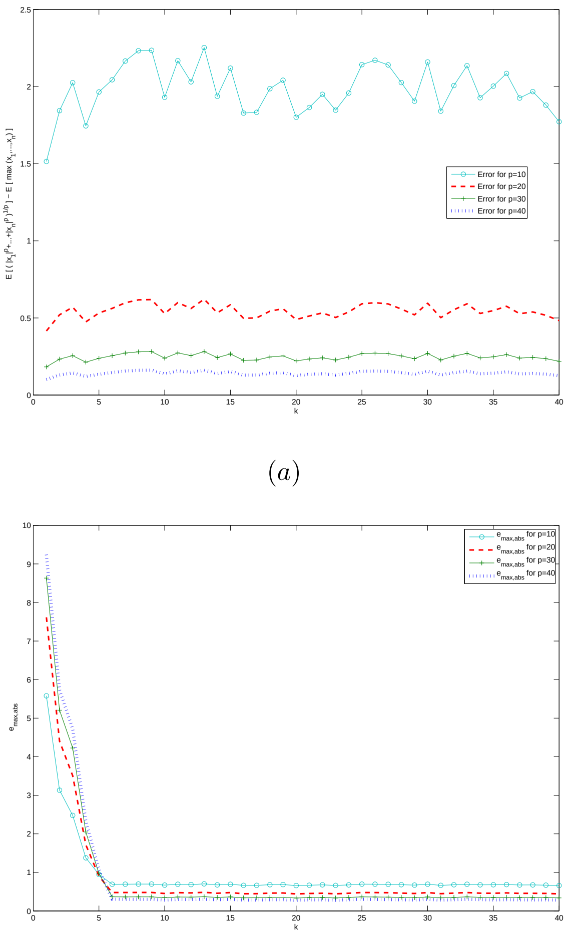
[3] These times are obtained running Matlab 7.11.0 (R2010b) on a 2.33 GHz Intel Core Duo E655 processor.

[4] The relative error is defined here as $\frac{|J_{\text{tot,oth}} - J_{\text{tot,ai}}|}{|J_{\text{tot,ai}}|}$ where $J_{\text{tot,ai}}$ is obtained using the analytic integration approach and $J_{\text{tot,oth}}$ is obtained using other methods.

Methods for computing the objective function	Computation time[s]	Mean value (MV) of J_{tot}	Relative error of MV of J_{tot}
Analytic integration	209	-235.4937	–
Nominal MPC	1.07	-228.0679	3.15%
Numerical integration	563	-235.5079	0.006%
Monte Carlo	2648	-235.5035	0.004%
Approximation method:			
$p = 10$	91	-228.3628	3.03%
$p = 20$	4009	-233.7849	0.73%
$p = 30$	25310	-234.8377	0.28%
$p = 40$	89451	-235.1228	0.16%

Table 5.1: The computation time, the mean value of J_{tot}, and the relative error of the mean value of J_{tot} using different methods to compute the expected value in the objective function.

a good choice. As reported in Table 5.1, for 40 event steps, the total computation times of the optimization procedure using the analytic integration is about a factor 2.5 lower than the one using numerical integration. Note that the small difference between the objective function values of these two methods is due to the error of numerical integration, which can be improved by increasing the number of samples in the numerical integration. For this experiment, we have chosen 10^5 samples. If we increase the number of samples to 10^7, the value of the relative error of the objective function will be 0.0009%; however, the computation time will be about a factor 250 larger than the one using the analytic integration, and for 10^{10} samples it is not tractable anymore. Based on a trade-off between the CPU time and the objective function value, it has been decided to do the experiments with 10^5 samples. The total computation time of the optimization procedure using Monte Carlo simulation, reported in Table 5.3, is also for 10^5 samples. For 10^7 samples, the relative error of objective function will be 0.0005%, but again the computation time will be a factor 100 larger than the one listed in Table 5.3. Hence, due to the same trade-off as before, we also chose 10^5 samples for the Monte Carlo simulations. For the approximation method, we observe that on the one hand, the mean value of J_{tot} for a large p, i.e., $p = 40$, becomes closer to the one of the analytic solution and accordingly, the relative error becomes smaller. On the other hand, the computation time increases drastically by increasing p. Note that since we use the approximation function (5.4), for $p = 10$ we deal with at least 66 and at most 1001 terms while computing the p-th moments of the uniformly distributed random variables using the multivariate theorem (cf. Theorem 4.2.6). This number is even worse if $p = 40$, i.e., we have at least 861 and at most 135751 terms. Hence, considering both the computation time and the relative error of the objective function, the results of the approximation method for $p = 20$ and 30 can be used in place of the exact solution.



(a)

(b)

Figure 5.3: (a) The error caused by Inequality (ii) in (4.4) *of Proposition 4.2.3; (b) The upper bound* $e_{\text{max,abs}}$ *(cf.* (4.16)*) for the error caused by Inequality (iii) (Jensen's inequality) in* (4.4) *of Proposition 4.2.3.*

In Figure 5.3 the errors occurring in (4.4) in Proposition 4.2.3 are illustrated. In fact, we have computed the difference between $J^*(k)$ and $J^*_{\text{app}}(k)$, where $J^*(k)$

is obtained by solving the optimization problem (3.8) in closed-loop using the analytic integration and $J^*_{\mathrm{app}}(k)$ is computed using (5.8) with the same optimal state and input sequence obtained from the optimization problem (3.8), and then the errors that occur in each inequality in (4.4) are presented here. Note that in the case of the uniform distribution, since the domain of random variables is bounded, we can specify the value of L in (4.5) explicitly, and hence, all variables will be nonnegative. Therefore, the error caused by Inequality (i) in (4.4) is zero. The error caused by Inequality (ii), i.e., the error between the p-norm and the ∞-norm, is shown in Figure 5.3(a), which, by choosing a larger p, becomes smaller and eventually it converges to zero. Figure 5.3(b) shows the upper bound, i.e., $e_{\mathrm{abs}}(a,b)$ in (4.16), for the error of Jensen's inequality due to (iii).

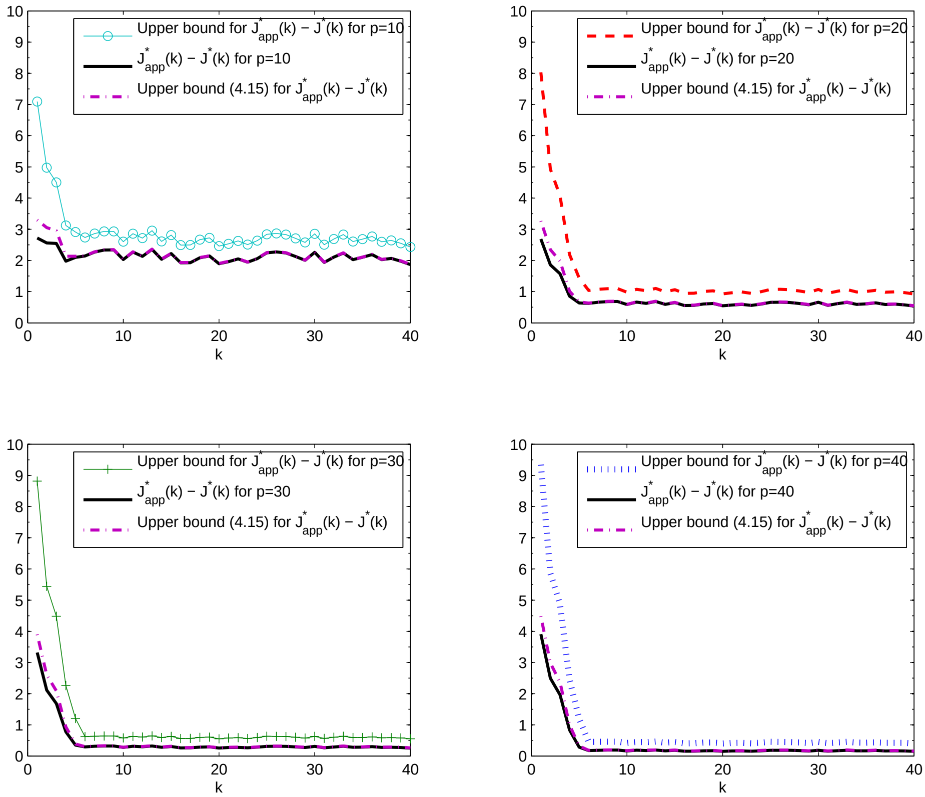


Figure 5.4: The upper bounds for and the exact value of $J^*_{\mathrm{app}}(k) - J^*(k)$.

Figure 5.4 shows the exact difference between $J^*_{\mathrm{app}}(k)$ and $J^*(k)$, i.e., $J^*_{\mathrm{app}}(k)-$

$J^*(k)$, the upper bound for this difference obtained by adding the results of the first and the second plot of Figure 5.3 for different values of p, and the upper bound (4.15) for $J^*_{\rm app}(k) - J^*(k)$. As illustrated in this figure, by increasing p, both upper bounds and the difference between $J^*_{\rm app}(k)$ and $J^*(k)$ become smaller and after a few step, they all remain below 0.7 for $p = 30$ and 40. It also shows that, after few steps, the upper bound (4.15) for $J^*_{\rm app}(k) - J^*(k)$ is very close to the exact difference, and the other upper bound also becomes quite close to the exact difference, which are indications for the good performance of the approximation method in this specific example.

As a result, by comparing the CPU times, the mean value of $J_{\rm tot}$, and the relative errors, we can conclude that in the case of the uniform distribution, the methods based on analytic and numerical integration are preferred to the approximation method. Note that in any case, if the approximation method is used, one should make a trade-off between the computation time and the closeness of the results to the solution obtained using analytic integration (as shown in Figure 5.2) and hence, tune the value of p accordingly, which was done here by means of experiments.

Normal distribution

In the second case, we assume that the signal d_1 is corrupted by standard normally distributed noise, i.e., the elements of $\tilde{e}$ are $\tilde{e}_i \sim \mathcal{N}(0,1),\ \ i = 1,\dots,4$. In this case, we can define the scalar L as follows:

$$\begin{aligned}
L_1 &= \min(\eta_1 - 3\sqrt{2}, \eta_2 - 3(1), \eta_3, 0)\\
L_2 &= \min(\eta_4 - 3\sqrt{3}, \eta_5 - 3\sqrt{2}, \eta_6 - 3(1), \eta_7 - 3\sqrt{2}, \eta_8 - 3(1), \eta_9, 0) \qquad (5.12)\\
L_3 &= \min(\eta_{10} - 3(2), \eta_{11} - 3\sqrt{3}, \eta_{12} - 3\sqrt{2}, \eta_{13} - 3(1), \eta_{14} - 3\sqrt{3},\\
&\qquad\quad \eta_{15} - 3\sqrt{2}, \eta_{16} - 3(1), \eta_{17} - 3\sqrt{2}, \eta_{18} - 3(1), \eta_{19}, 0)
\end{aligned}$$

Recall that this choice is based on the 3σ-rule.

Figure 5.5 shows the difference between the output signal y and the due date signal r for closed loop simulation. As in the previous case, the optimal input sequence is computed for the closed-loop system for $k = 1,\dots,40$. Note that since we have normally distributed random variables, their probability density function is not piecewise polynomial. Hence, to be able to use the analytic integration of Section 3.1.3, first it has to be approximated by a piecewise polynomial function and since this approximation would cause significant complexity besides increasing the computation time, we did not consider this approach for this example. Therefore, in this figure, the "Exact solution" is obtained by using numerical integration to compute the expected value in the objective function, the "Nominal MPC" is obtained by ignoring the noise in the computation of the optimal input sequence, and the "Approximation" is obtained by using the approximate objective function $J_{\rm app}(k)$

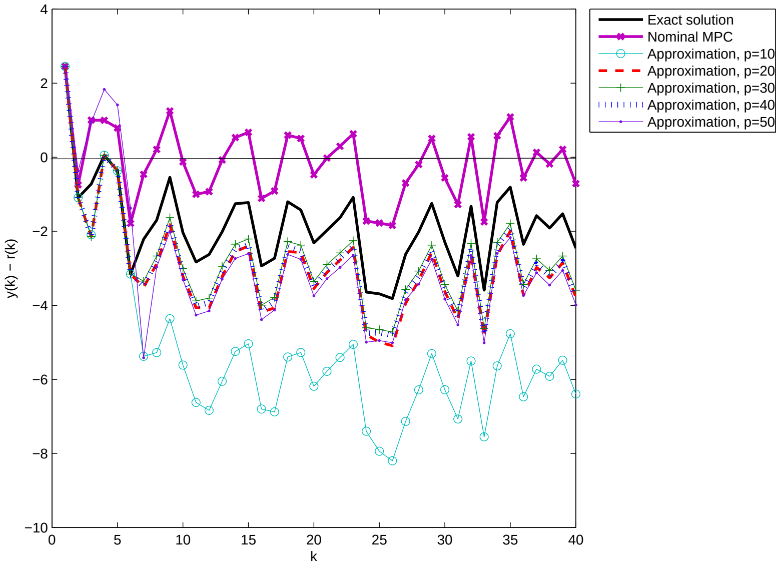


Figure 5.5: Due date error $y(k) - r(k)$ for the closed-loop system using nominal MPC, analytic integration to compute the expected value in the objective function, and the approximate objective function $J_{\mathrm{app}}(k)$ in (5.8).

in (5.8). Here also, we have used Monte Carlo simulation to compute the expected value in the objective function, but it is not plotted since it exactly covers the plot of the exact solution. Similar to the previous case, the nominal MPC results in the violation of due dates. The approximation method gives results that are close to the "Exact solution" for $p = 20,\ 30$, and 40. Note that for $p = 50$, the results obtained using the approximation method is not comparable to the "Exact solution" anymore. This observation shows clearly that a larger p does not always give a better result.

In Table 5.2, the total optimization time for closed-loop simulation over 40 event steps using numerical integration, nominal MPC, Monte Carlo simulation, and the approximation method for different values of p are reported. Similar to the previous case, for each method, the mean value of the objective function J_{tot} over the entire simulation period using 20 different noise realizations and the relative error of the mean value of J_{tot} using the analytic integration versus using the other methods is presented.

Here again, compared to numerical integration, nominal MPC results in a very large objective function value. Hence, despite the very short computation time, this method is not reliable to be applied. The computation time of the optimization procedure using the approximation method is on average about a factor 200 smaller

Methods for computing the objective function	Computation time[s]	Mean value (MV) of J_{tot}	Relative error of MV of J_{tot}
Numerical integration	744	-229.7778	–
Nominal MPC	1.08	-217.1595	5.49%
Monte Carlo	2574	-229.8415	0.03%
Approximation method:			
$p = 10$	8.19	-224.3753	2.35%
$p = 20$	9.37	-227.5341	0.98%
$p = 30$	12.19	-226.9461	1.23%
$p = 40$	14.89	-226.1622	1.57%
$p = 50$	19.01	-225.5101	1.86%

Table 5.2: The computation time, the mean value of J_{tot}, and the relative error of the mean value of J_{tot} using different methods to compute the expected value in the objective function.

than the one using Monte Carlo simulation and it is about a factor 60 smaller than the one using numerical integration. Here also the number of samples of both Monte Carlo simulation and numerical integration is 10^5 since, as explained in the previous case, a larger number of samples increases the computation time significantly. Note also that since we use the approximation function (5.5), increasing the value of p has only a linear effect on the computation time and even for $p = 50$, the computation time is very low compared to Monte Carlo simulation or numerical integration. As a result, by comparing the CPU time of these three methods, we can conclude that the approximation method is considerably faster than numerical integration and Monte Carlo simulation. Moreover, this table shows again that – considering the relative error for different values of p – a larger p does not always make the approximation better. Consequently, one needs to find the appropriate value of p, which has been done here by means of experiments, that gives the best approximation result.

Figure 5.6 illustrates $J^*_{\mathrm{app}}(k) - J^*(k)$, where, similar to the previous example, $J^*(k)$ is obtained by solving the optimization problem (3.8) in closed-loop using numerical integration and $J^*_{\mathrm{app}}(k)$ is computed using (5.8) with the same optimal state and input sequence obtained from the optimization problem (3.8). This figure also shows the upper bound for $J^*_{\mathrm{app}}(k) - J^*(k)$, which is indeed obtained using (4.15). Note that unlike the previous example, since here we have a normally distributed error vector, which has an unbounded domain, we cannot compute the upper bound for Jensen's inequality, i.e., $e_{\mathrm{max,abs}}$ (cf. Section 4.3). As shown in Figure 5.6, for $p = 20$ the upper bound for $J^*_{\mathrm{app}}(k) - J^*(k)$ is below 1 after few steps and

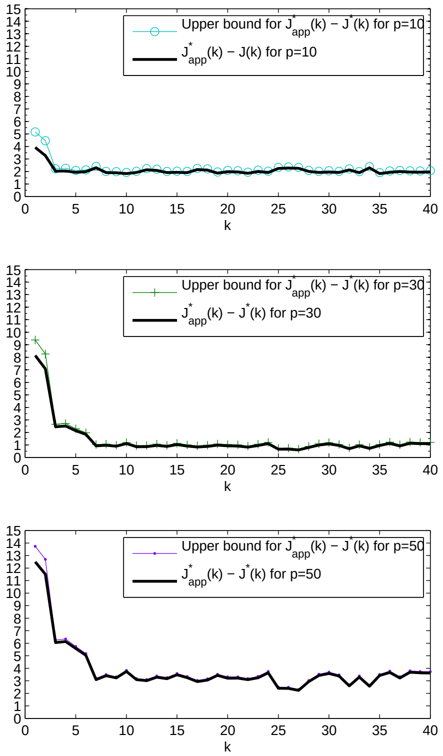


Figure 5.6: The upper bound (4.15) *for and the exact value of* $J^*_{\mathrm{app}}(k) - J^*(k)$.

for $p = 30$ it is almost below 1. However, for $p = 40$ both the upper bound for $J^*_{\mathrm{app}}(k) - J^*(k)$ and the difference itself are between 1 and 2 and for $p = 50$ they both become larger than 2. This observation supports our previous conclusion that a larger p is not always a better option and hence, for choosing p there must be a trade-off between having a good approximation and the size of the error caused by this approximation method.

Consequently, based on the computation time, the mean value of J_{tot}, and the relative and the approximation errors, we can conclude that the approximation method is a reliable and time-efficient method to solve the stochastic MPL-MPC optimization problem in this case. Indeed, for $p = 20,\ 30$, and 40 the result of the closed-loop optimization using the approximation method is quite close to the one using the numerical integration and, in this specific example, the approximation-based approach is about 60 times faster than the approach using numerical integration.

5.2 Approximate Identification of Stochastic MPL Systems

Similar to conventional systems, model parameters of a stochastic MPL system can also be identified. We consider a state space model and we take into account the stochastic properties of the system in the identification process. Just as the MPC problem discussed in the previous section, in identification of stochastic MPL systems, we face with the computational difficulties as well, due to the presence of the expected value in the objective function. Our aim in this section is to apply the approximation method of Chapter 4 to decrease the computational complexity of identification of stochastic MPL systems.

5.2.1 Problem Statement

We aim to solve the following identification problem:

$$\begin{aligned}\min_{(\hat{\theta},\hat{\lambda})} \quad & J(\hat{\theta},\hat{\lambda}) \\ \text{s.t.} \quad & \hat{\lambda} > 0\end{aligned} \tag{5.13}$$

with

$$\begin{aligned}J(\hat{\theta},\hat{\lambda}) &= \sum_{k=1}^{N-1}\sum_{i=1}^{n} \left(\mathbb{E}[x_i(k+1|k)] - x_{\text{meas},i}(k+1)\right)^2 \\ &= \sum_{k=1}^{N-1}\sum_{i=1}^{n} \Big(\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\Big)^2,\end{aligned} \tag{5.14}$$

where, considering (3.18) and (3.19),

$$\begin{aligned}&\mathbb{E}[x_i(k+1|k)] - x_{\text{meas},i}(k+1) \\ &\qquad = \mathbb{E}\big[\max_{j=1,\ldots,m}\big(\xi_{ij}+\hat{\theta}^T\Delta^{(i)}_{\cdot,j}+e^T(k)\hat{\Lambda}S^{(i)}_{\cdot,j}+\phi_j(k)-x_{\text{meas},i}(k+1)\big)\big] \\ &\qquad = \mathbb{E}\big[\max_{j=1,\ldots,m}(\alpha_{ij}(k)+\Pi_{ij}^T\hat{\theta}+\hat{\lambda}^T\Gamma_{ij}e(k))\big] \\ &\qquad = \mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\end{aligned} \tag{5.15}$$

for appropriately defined matrices and vectors α_{ij}, Π_{ij}, and Γ_{ij} (cf. Section 3.2.1). The goal is to identify the two parameters $\hat{\theta}$ and $\hat{\lambda}$ by solving (5.13). To this end, $\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]$ has to be computed in an efficient way. Numerical integration or the analytic integration using piecewise polynomial probability density functions are again possible computational methods. However, considering the complexity and computation load of these methods, it is desired to find alternative solution approaches.

5.2.2 Approximation Approach

As an alternative approach to increase the computational efficiency, we will apply the approximation method to solve the identification problem (5.13). Consequently, we can approximate the function $\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]$ in (5.15) by $\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\Big)$ where

$$\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\Big) \\ = \Bigg(\sum_{j=1}^{m}\sum_{k_0+k_1+\cdots+k_{n_e}=p}\frac{p!}{k_0!\,k_2!\cdots k_{n_e}!}\gamma_{ij,0}^{k_0}\prod_{t=1}^{n_{\tilde{e}}}\gamma_{ij,t}^{k_t}\mathbb{E}\big[\tilde{e}_t^{k_t}(k)\big]\Bigg)^{1/p}+L \quad (5.16)$$

with $\gamma_{ij,0}=\alpha_{ij}+\Pi_{ij}^T\hat{\theta}-L$ and $\gamma_{ij,t}=(\hat{\lambda}^T\Gamma_{ij})_t e_t$ for $j=1,\dots,m$ and $t=1,\dots,n_e$ and with the stochastic vector $e=[e_1,\dots,e_{n_e}]^T$ and Γ_{ij} being a diagonal matrix. Furthermore, we can obtain subgradients of $\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\Big)$ with respect to $\hat{\theta}$ and $\hat{\lambda}$. Note that only $\gamma_{ij,0}$ depends on $\hat{\theta}$, and the rest of $\gamma_{ij,t},\ t=1,\dots,n_e$ depend only on $\hat{\lambda}$. Accordingly, by applying the chain rule, we obtain the following subgradients for (5.16):

$$\nabla_{\hat{\theta}}\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\Big) \\ = \Bigg(\sum_{j=1}^{m}\sum_{k_0+k_1+\cdots+k_{n_e}=p}\frac{p!}{k_0!\,k_2!\cdots k_{n_e}!}\gamma_{ij,0}^{k_0}\prod_{t=1}^{n_{\tilde{e}}}\gamma_{ij,t}^{k_t}\mathbb{E}\big[\tilde{e}_t^{k_t}(k)\big]\Bigg)^{1/p-1}\cdot \\ \Bigg(\sum_{j=1}^{m}\Pi_{ij}\sum_{k_0+k_1+\cdots+k_{n_e}=p-1}\frac{(p-1)!}{k_0!\,k_2!\cdots k_{n_e}!}k_0\gamma_{ij,0}^{k_0-1}\prod_{t=1}^{n_{\tilde{e}}}\gamma_{ij,t}^{k_t}\mathbb{E}\big[\tilde{e}_t^{k_t}(k)\big]\Bigg)$$

and

$$\nabla_{\hat{\lambda}}\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\Big) \\ = \Bigg(\sum_{j=1}^{m}\sum_{k_0+k_1+\cdots+k_{n_e}=p}\frac{p!}{k_0!\,k_2!\cdots k_{n_e}!}\gamma_{ij,0}^{k_0}\prod_{t=1}^{n_{\tilde{e}}}\gamma_{ij,t}^{k_t}\mathbb{E}\big[\tilde{e}_t^{k_t}(k)\big]\Bigg)^{1/p-1}\cdot \\ \Bigg(\sum_{j=1}^{m}\sum_{k_0+k_1+\cdots+k_{n_e}=p}\frac{(p-1)!}{k_0!\,k_2!\cdots k_{n_e}!}\gamma_{ij,0}^{k_0}\sum_{\ell=1}^{n_e}k_\ell(\Gamma_{ij})_{\ell\ell}\gamma_{ij,\ell}^{k_\ell-1}\mathbb{E}[e_\ell^{k_\ell}]\prod_{\substack{t=1\\t\neq\ell}}^{n_e}\gamma_{ij,t}^{k_t}\mathbb{E}\big[\tilde{e}_t^{k_t}(k)\big]\Bigg)$$

In the case that the elements of the stochastic vector $e(k)$ have a distribution that is preserved under summation, we can equivalently approximate $\mathbb{E}[\eta_i(k+$

$1, \hat{\theta}, \hat{\lambda}, e(k))]$ by

$$\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]\Big) = \left(\sum_{j=1}^{m} \mathbb{E}\big[\big(\alpha_{ij} + \Pi_{ij}^T \hat{\theta} + \hat{\lambda}^T \Gamma_{ij} e(k) - L\big)^p\big] \right)^{1/p} + L \tag{5.17}$$

with the following subgradients

$$\begin{aligned} &\nabla_{\hat{\theta}} \mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]\Big) \\ &\qquad = \left(\sum_{j=1}^{m} \mathbb{E}\big[\big(\alpha_{ij} + \Pi_{ij}^T \hat{\theta} + \hat{\lambda}^T \Gamma_{ij} e(k) - L\big)^p\big] \right)^{1/p-1} \cdot \\ &\qquad\quad \sum_{j=1}^{m} \Pi_{ij} \mathbb{E}\big[\big(\alpha_{ij} + \Pi_{ij}^T \hat{\theta} + \hat{\lambda}^T \Gamma_{ij} e(k) - L\big)^{p-1}\big] \end{aligned}$$

and

$$\begin{aligned} &\nabla_{\hat{\lambda}} \mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]\Big) \\ &\qquad = \left(\sum_{j=1}^{m} \mathbb{E}\big[\big(\alpha_{ij} + \Pi_{ij}^T \hat{\theta} + \hat{\lambda}^T \Gamma_{ij} e(k) - L\big)^p\big] \right)^{1/p-1} \cdot \\ &\qquad\quad \sum_{j=1}^{m} \Gamma_{ij} \mathbb{E}\big[e(k)\ \big(\alpha_{ij} + \Pi_{ij}^T \hat{\theta} + \hat{\lambda}^T \Gamma_{ij} e(k) - L\big)^{p-1}\big] \end{aligned}$$

As a result, the objective function $J(\hat{\theta}, \hat{\lambda})$ in (5.14) can be approximated by replacing $\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]$ with $\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]\Big)$ as follows:

$$J_{\text{app}}(\hat{\theta}, \hat{\lambda}) = \sum_{k=1}^{N-1} \sum_{i=1}^{n} \Big(\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]\Big) \Big)^2 \tag{5.18}$$

with the gradients

$$\begin{aligned} &\nabla_{\hat{\theta}} J_{\text{app}}(\hat{\theta}, \hat{\lambda}) \\ &\quad = \sum_{k=1}^{N-1} \sum_{i=1}^{n} 2\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]\Big)\ \nabla_{\hat{\theta}} \mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1, \hat{\theta}, \hat{\lambda}, e(k))]\Big) \end{aligned}$$

and

$$\nabla_{\hat{\lambda}} J_{\text{app}}(\hat{\theta}, \hat{\lambda})$$

$$= \sum_{k=1}^{N-1} \sum_{i=1}^{n} 2\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\Big)\, \nabla_{\hat{\lambda}}\mathfrak{U}\Big(\mathbb{E}[\eta_i(k+1,\hat{\theta},\hat{\lambda},e(k))]\Big)$$

and hence, solve the following approximate identification problem

$$\min_{(\hat{\theta},\hat{\lambda})} \quad J_{\text{app}}(\hat{\theta},\hat{\lambda})$$
$$\text{s.t.} \quad \hat{\lambda} > 0$$

by means of a multi-start gradient-based optimization method, such as a steepest descent method or a Quasi-Newton (DFP, BFGS) method [90].

5.2.3 Example

In this section we present two case studies to discuss the results of the identification using different methods to compute (5.15), namely the approximation method (cf. (5.16) and (5.17)), numerical integration[5], the analytic integration using piecewise polynomial probability density functions (cf. Section 3.1.3), and Monte Carlo simulation.

In the first case study, we consider a uniformly distributed noise vector, which has a bounded domain, and we compare the performance of the above-mentioned approaches with one another. In the second case study, a normally distributed noise vector, which has an unbounded domain, is considered. Note that if we apply the analytic integration approach of Section 3.1.3 to the case with normally distributed random variables, we would need an approximation using piecewise-polynomial functions. This would introduce approximation errors as well as an increase in the computational complexity. To avoid this additional complexity, we did not consider this approach in this example. Hence, we only compare the performance of the approximation method with the approach using numerical integration and with the one using Monte Carlo simulation for the computation of (5.15).

In order to obtain a system of the form (3.17), we consider the following stochastic state space model:

$$x(k) = A(k) \otimes x(k-1) \oplus B(k) \otimes u(k) \tag{5.19}$$
$$y(k) = C(k) \otimes x(k) \tag{5.20}$$

with the system matrices

$$A(k) = \begin{bmatrix} \theta_1(k) & 0 \\ \epsilon & \theta_2(k) \end{bmatrix} \qquad B(k) = \begin{bmatrix} \theta_3(k) \\ \theta_4(k) \end{bmatrix} \qquad C(k) = \begin{bmatrix} 0 & 0 \end{bmatrix}.$$

[5] The numerical integration is based on the Monte Carlo integration [25]

In this example, we generate input-state data by simulating the system for 400 event steps, i.e., for $k = 1, \ldots, 400$. The parameter estimation is done using this data, in which the input signal is a staircase signal [97] with an average slope of about 1.83 given by

$$u(k) = 5.5 \cdot \Big(1 + \lfloor k/3 \rfloor\Big)$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. The input signal $u(k)$ in shown in Figure 5.7 for $k = 1, \ldots, 40$.

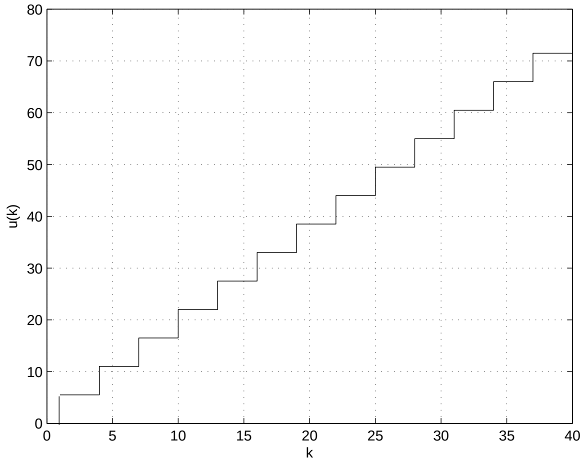


Figure 5.7: The first 40 samples of the input signal $u(k)$ for the first and the second case study of Section 5.2.3.

We estimate the parameters θ and λ assuming that the elements of the noise vector are uniformly and normally distributed, respectively.

Uniform distribution

We choose the true parameter vector θ as

$$\theta \;=\; \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}^T = \begin{bmatrix} 0.3 & 0.3 & 0.7 & 0.6 \end{bmatrix}^T$$

and we assume that its elements are perturbed by the uniformly distributed noise components $e_\ell(k)$ such that $e_\ell(k) \sim \mathcal{U}(-1, 1)$ for $\ell = 1, \ldots, 4$, and with scaling factor

$$\lambda \;=\; \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \end{bmatrix}^T = \begin{bmatrix} 0.3 & 0.3 & 0.3 & 0.3 \end{bmatrix}^T.$$

Hence, the corresponding value of the cost function, using θ and λ to generate the input-state data, is 21.8512.

As a first step, we estimate the parameter θ for a deterministic model, i.e., a noiseless model with $\hat{\lambda} = \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \end{array}\right]^T$, using the residuation-based estimation techniques described in [4, 23, 77]. Note that in this case, we do not expect to have a good estimation since we are ignoring the effect of noise. The optimization yields the following results:

$$\hat{\theta} = \left[\begin{array}{cccc} 0.0167 & 0.0009 & 0.4056 & 0.3011 \end{array}\right]^T.$$

and the corresponding value of the objective function is 90.3505. As expected, due to the absence of a noise model the estimation fails and the estimated parameters are quite far from the true values.

The second step is to estimate the parameters θ and λ for the stochastic system (5.19)-(5.20). We minimize the objective function (5.13) based on the one-step ahead prediction, i.e., we predict the behavior of the system at the event step $k+1$ based on the information that we have at the event step k. We use a multi-start, sequential quadratic programming (SQP) method [80], considering 30 different starting points that are chosen randomly and are both larger and smaller than the real values of θ and λ to start the optimization with, and then, we report the estimated parameters for which the objective function has the lowest value.

We use four different methods to compute (5.15): Monte Carlo simulation [63], numerical integration, the analytic integration method, and the approximation method using the function (5.16). By means of experiments, we found out that $p = 14$ gives a good approximation in this specific example. The results of the optimization are presented in Table 5.3. As shown, the estimated parameter $\hat{\theta}$ is quite close to the exact value of θ for the above-mentioned methods. However, for λ we do not have a good estimation. Note that, in general, in prediction error identification, one can obtain the correct system model, i.e., θ, but it is much more difficult to estimate the noise model, i.e., λ [48, 73].

The reason that the analytic integration method of Section 3.1.3 and numerical integration give different results (cf. Table 5.3) is – apart from the numerical integration accuracy – mainly due to the fact that here we have a non-convex optimization problem in combination with optimization runs with different random starting points. As reported in Table 5.3, for 400 event steps, the average computation time[6] of the optimization procedure, over 30 different starting points, using the approximation method and analytic integration approach is quite close (it is about a factor 1.5 lower for the approximation method). However, the computation time of the optimization problem using the analytic integration approach is about a factor 30 lower than using numerical integration with 10^5 samples. If we increase the number of samples to 10^7 the computation time using numerical integration becomes about

[6] These times are obtained running Matlab 7.11.0 (R2010b) on a 2.33 GHz Intel Core Duo E655 processor.

Optimization results	Monte Carlo simulation	Numerical integration	Analytic integration	Approximation method
$\hat{\theta}$	$\begin{bmatrix} 0.2824 \\ 0.2944 \\ 0.6910 \\ 0.5885 \end{bmatrix}$	$\begin{bmatrix} 0.2976 \\ 0.3018 \\ 0.6967 \\ 0.5898 \end{bmatrix}$	$\begin{bmatrix} 0.2841 \\ 0.2926 \\ 0.6954 \\ 0.5808 \end{bmatrix}$	$\begin{bmatrix} 0.3012 \\ 0.2823 \\ 0.6746 \\ 0.5991 \end{bmatrix}$
$\hat{\lambda}$	$\begin{bmatrix} 0.1575 \\ 0.3882 \\ 0.6224 \\ 0.0027 \end{bmatrix}$	$\begin{bmatrix} 0.2838 \\ 0.4694 \\ 0.4529 \\ 0.0896 \end{bmatrix}$	$\begin{bmatrix} 0.4591 \\ 0.3239 \\ 0.0858 \\ 0.2700 \end{bmatrix}$	$\begin{bmatrix} 0.0596 \\ 0.0670 \\ 0.2613 \\ 0.0479 \end{bmatrix}$
Corresponding J	22.6907	21.8831	21.6958	22.3832
CPU time [s]	73549	44992	1523	1024

Table 5.3: Estimation results for θ and λ, using four different methods to calculate (5.15) *with uniformly distributed noise, corresponding value of the objective function J, and the average computation time (CPU time) of each method using 30 different random starting points.*

a factor 3000 larger than the one using the analytic integration, and for 10^{10} it is not even tractable anymore. For the numerical integration the relative error between the objective functions obtained using the analytic integration approach and the numerical integration with 10^5 samples is 0.03% and using 10^7 samples is 0.004%. Based on a trade-off between the CPU time and the relative error, it has been decided to do the experiments with 10^5 samples. The computation time of the optimization procedure using Monte Carlo simulation, reported in Table 5.3, is also for 10^5 samples and the relative error between the objective functions obtained using the analytic integration and the Monte Carlo simulation using this number of samples is 0.06% and using 10^7 samples, it is 0.008%. Hence, due to the same trade-off as before, we chose 10^5 samples. As a result, by comparing the CPU times of these four methods we can conclude that the analytic integration method and the approximation method are considerably faster (at least 30 times and 45 times, respectively) than numerical integration and the Monte Carlo simulation. Moreover, the results obtained using the approximation method are similar to the ones using analytic integration besides the fact that the CPU time using the approximation method is smaller than the one using the analytic integration approach.

Normal distribution

For the second case, the true parameter vector θ is the same as before, i.e.,

$$\theta \;=\; \left[\begin{array}{cccc} \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{array}\right]^T = \left[\begin{array}{cccc} 0.3 & 0.3 & 0.7 & 0.6 \end{array}\right]^T$$

except that now, each of its elements is perturbed by one of the noise components $e_\ell(k),\ \ell = 1, \ldots, 4$ that are independent and have a standard normal distribution, i.e., $e_\ell(k) \sim \mathcal{N}(0,1)$, with the scaling factor

$$\lambda \;=\; \left[\begin{array}{cccc} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \end{array}\right]^T = \left[\begin{array}{cccc} 0.1 & 0.1 & 0.1 & 0.1 \end{array}\right]^T.$$

and accordingly, the corresponding value of the cost function, using θ and λ to generate the input-state data, is 8.6958.

Similar to the previous case, first we estimate the parameter θ for a deterministic model, using the mentioned residuation-based estimation techniques. The optimization result is as follows:

$$\hat{\theta} = \left[\begin{array}{cccc} 0.0725 & -0.0218 & 0.8416 & 0.7035 \end{array}\right]^T.$$

with the corresponding value of the cost function equal to 55.6288. As we expected and as we have also seen in case of the uniform distribution, by neglecting the effect of noise, we do not obtain a good estimation.

In the next step, we estimate the parameters θ and λ for the stochastic system (5.19)-(5.20). To this end, we minimize the objective function (5.13) using three different methods: Monte Carlo simulation, numerical integration, and the approximation method using the function (5.17). As we did in the case of the uniform distribution, we minimize the objective function based on the one-step ahead prediction, using a multi-start, SQP method with 30 different starting points, and reporting the estimated parameter with the lowest objective function value. We have chosen $p = 30$ for the approximation method. This choice has been also made by means of experiments, as for $p = 30$, we obtain a good approximation in this specific example. The estimation results are presented in Table 5.4.

Comparing the results, we can conclude that the approximation method gives a good estimation for θ that is quite close to the results obtained using numerical integration and Monte Carlo simulation as well as to the true value of θ. Similar to the first case, we obtain an unsatisfactory estimation for λ.

Recall that one of the goals of using the proposed approximation function (5.17) is to decrease the computation time. For 400 event steps, the computation times of the optimization procedure using the three above-mentioned methods are presented

Optimization results	Monte Carlo simulation	Numerical integration	Approximation method
$\hat{\theta}$	$\begin{bmatrix} 0.3030 \\ 0.2984 \\ 0.6881 \\ 0.5940 \end{bmatrix}$	$\begin{bmatrix} 0.3092 \\ 0.2969 \\ 0.6974 \\ 0.5937 \end{bmatrix}$	$\begin{bmatrix} 0.2750 \\ 0.2824 \\ 0.6799 \\ 0.5781 \end{bmatrix}$
$\hat{\lambda}$	$\begin{bmatrix} 0.0449 \\ 0.0417 \\ 0.0479 \\ 0.0398 \end{bmatrix}$	$\begin{bmatrix} 0.0976 \\ 0.1016 \\ 0.1236 \\ 0.0945 \end{bmatrix}$	$\begin{bmatrix} 0.0409 \\ 0.0400 \\ 0.0403 \\ 0.0419 \end{bmatrix}$
Corresponding J	8.6980	8.7403	9.1231
CPU time	110796 s	83890 s	899 s

Table 5.4: Estimation results for θ and λ, using three different methods to calculate (5.15) *with a normally distributed noise, corresponding value of the objective function J, and the average computation time (CPU time) of each method using 30 different random starting points.*

in Table 5.4. The reported CPU time for Monte Carlo simulation and numerical integration in this example is also for 10^5 samples due to the trade-off between the CPU time and the accuracy level of the results. Therefore, the approximation method increases the time efficiency significantly (it is about 80 times faster than the two other methods) while still resulting in a performance that is comparable to the one of the other two methods.

5.3 Summary

In this chapter we have considered the approximate MPC and identification of stochastic MPL systems. The approximation method proposed in Chapter 4 has been applied to compute the expected value of max-plus-scaling functions that appear in the stochastic MPL-MPC optimization and in the identification problem. This approximation method yields an upper bound for the objective function and hence, instead of minimizing the objective function itself, we minimize its upper bound. Since the approximation method results in an analytic solution for distributions with moments for which a closed-form expression exists, the approximate

MPC and identification problem are in general less computationally complex and can thus, be solved more efficiently comparing to the existing methods.

Moreover, we obtain a convex MPC optimization problem that can be solved efficiently by means of the existing convex optimization algorithms since the upper bound of the expected value obtained using this approximation method is convex in the control variables and the exact expression for the gradient with respect to these variables can be computed. In the approximate identification problem, we can also compute the gradients analytically and hence, although it is a non-convex problem, it can be solved using gradient-based algorithms.

For both MPC and identification problems, we have presented examples using two different types of distributions: uniform distribution, which has a bounded domain, and normal distribution, which an unbounded domain. The results shows that for both distributions, the performance of the approximation method is comparable to one of the "exact" solution.

Chapter 6

Approximation Approach for Model Predictive Control for Stochastic Switching Max-Plus-Linear and Stochastic Max-Min-Plus-Scaling Systems

In this chapter, we proceed further with applying the approximation method of Chapter 4 to other classes of MPL systems, namely stochastic switching MPL systems and stochastic MMPS systems. Similar to the previous chapter, our aim here is to reduce the computational complexity as well as the computation time of the MPC optimization problem for these classes of systems.

6.1 Approximate MPC for Stochastic Switching MPL Systems

We have seen in Chapter 3 that stochastic switching MPL systems can be controlled using MPC. Accordingly, at each event step a stochastic MPL-MPC optimization problem has to be solved, which is time-consuming and complex as explained in Section 5.1. This issue, in addition to the number of possible mode switchings, which in practice can be very large, imposes a considerable computational burden. Therefore, we propose some approximation methods in this section in order to decrease the computational difficulties of the stochastic switching MPL-MPC optimization problem. One of these methods addresses the problem caused by the large number of mode switchings and the other method aims at decreasing the complexity imposed by the presence of stochastic system parameters.

6.1.1 Problem Statement

In stochastic switching MPL systems, the MPC optimization problem can be defined as follows (cf. Section 3.1.5):

$$
\begin{aligned}
&\min_{\tilde{u}(k)} \quad J(k) \\
\text{s.t.} \quad & \tilde{y}(k) = \tilde{C}(\tilde{\ell}(k), \tilde{e}(k)) \otimes x(k-1) \oplus \tilde{D}(\tilde{\ell}(k), \tilde{e}(k)) \otimes \tilde{u}(k) \\
& \Delta u(k+j) \geq 0 \qquad \text{for } j = 0, \ldots, N_\mathrm{p} - 1 \\
& \Delta^2 u(k+j) = 0 \qquad \text{for } j = N_\mathrm{c}, \ldots, N_\mathrm{p} - 1 \\
& A_\mathrm{con}(k)\tilde{u}(k) + B_\mathrm{con}(k)\mathbb{E}[\tilde{y}(k)] \leq c_\mathrm{con}(k)
\end{aligned}
\tag{6.1}
$$

where $\tilde{e}$ and $\tilde{\ell}$ are stochastic variables and

$$
J(k) = \sum_{i=1}^{n_y \cdot N_\mathrm{p}} \mathbb{E}[\max(\tilde{y}_i(k) - \tilde{r}_i(k), 0)] - \lambda \sum_{l=1}^{n_u \cdot N_\mathrm{p}} \tilde{u}_l(k) \tag{6.2}
$$

Accordingly, to solve (6.1), we need to find an efficient method to compute the objective function (6.2) and hence, the expected value therein. As mentioned before, the objective function (6.2) can be defined using any other combination of the input and output objective functions in (3.2) and the procedure presented below will be still valid.

Note that in stochastic switching MPL systems, we deal with two different types of random variables, namely discrete and continuous random variables. The mode switching uncertainty is a discrete random variable since we have a finite number of modes n_L and thus, the number of possible mode switching sequences over the whole prediction horizon ($(n_L)^{N_\mathrm{p}}$) is finite. On the contrary, the parametric uncertainty related to processing and transportation times is a continuous random variable. As a result, to compute the expected value, we need to apply a combination of integration over the continuous stochastic variable $\tilde{e}(k)$ and summation over the discrete stochastic variable $\tilde{\ell}(k)$. To this end, we need the joint probability density function of $\tilde{e}(k)$ and $\tilde{\ell}(k)$, which can be defined using the conditional probability theorem [65], as follows:

$$
f_{\mathcal{L},\mathcal{E}}(\tilde{\ell}(k), \tilde{e}(k)) = f_{\mathcal{E}}(\tilde{e}(k))\tilde{P}\big[L = \tilde{\ell}(k) \mid E = \tilde{e}(k)\big] \tag{6.3}
$$

where $\mathcal{L}$ is the (discrete) sample space of all the mode switching sequences $\tilde{\ell}(k)$, $\mathcal{E}$ is the (continuous) sample space of the parametric uncertainty $\tilde{e}(k)$, $f_{\mathcal{E}}(\tilde{e}(k))$ is the probability density function of $\tilde{e}(k)$, and $\tilde{P}\big[L = \tilde{\ell}(k) \mid E = \tilde{e}(k)\big]$ is the probability that we have mode switching sequence $\tilde{\ell}(k)$, given the parametric uncertainty $\tilde{e}(k)$ (cf. (2.10)).

Now let $v_i(\tilde{\ell}(k), \tilde{e}(k)) = \max(\tilde{y}_i(k) - \tilde{r}_i(k), 0)$. Considering (6.3), the expected value of $v_i(\tilde{\ell}(k), \tilde{e}(k))$ can be defined as follows:

$$\mathbb{E}\big[v_i(\tilde{\ell}(k), \tilde{e}(k))\big] = \sum_{\tilde{\ell} \in \mathcal{L}} \Big[\int_{\mathcal{E}} v_i(\tilde{\ell}, \tilde{e}) f_{\mathcal{E}}(\tilde{e}) \tilde{P}\big[L = \tilde{\ell} \mid E = \tilde{e}\big] \, d\tilde{e} \Big] \tag{6.4}$$

where both $f_{\mathcal{E}}(\tilde{e})$ and the mode switching probability $\tilde{P}\big[L = \tilde{\ell} \mid E = \tilde{e}\big]$ are assumed to be known. Note that in general (6.4) can be computed using numerical or analytic integration depending on the probability distribution of $\tilde{e}(k)$ and the (in)dependency of the random variables $\tilde{e}(k)$ and $\tilde{\ell}(k)$, and as mentioned in the previous chapters, both of these methods are complex and time-consuming.

6.1.2 Approximation Approach

In this section, we propose approximation methods to compute $\mathbb{E}[v_i(\tilde{\ell}(k), \tilde{e}(k))]$ in a more efficient way than analytic or numerical integration. To this end, we consider two cases: independent and dependent random variables.

Independent Random Variables

In the first case, $\tilde{e}(k)$ and $\tilde{\ell}(k)$ are independent random variables. In other words, the mode switching uncertainty $\tilde{\ell}(k)$ does not depend on the parametric uncertainty $\tilde{e}(k)$, so

$$\tilde{P}\big[L = \tilde{\ell}(k) \mid E = \tilde{e}(k)\big] = \tilde{P}\big[L = \tilde{\ell}(k)\big]. \tag{6.5}$$

Let $\mathcal{L} = \{\tilde{\ell}^1, \tilde{\ell}^2, \dots, \tilde{\ell}^M\}$ denote the set of all possible consecutive mode switching vectors for the problem (6.1) over the prediction horizon N_p with $M = (n_L)^{N_\text{p}}$. Accordingly, we can rewrite (6.4) as follows:

$$\begin{aligned}\mathbb{E}\big[v_i(\tilde{\ell}(k), \tilde{e}(k))\big] &= \sum_{\tilde{\ell} \in \mathcal{L}} \Big[\tilde{P}\big[L = \tilde{\ell}\big] \int_{\mathcal{E}} v_i(\tilde{\ell}, \tilde{e}) f_{\mathcal{E}}(\tilde{e}) d\tilde{e} \Big] \\ &= \sum_{\tilde{\ell} \in \mathcal{L}} \Big[\tilde{P}\big[L = \tilde{\ell}\big] \mathbb{E}_{\tilde{e}}\big[v_i(\tilde{\ell}, \tilde{e})\big] \Big] \\ &= \sum_{m=1}^{M} \Big[\tilde{P}\big[L = \tilde{\ell}^m\big] \mathbb{E}_{\tilde{e}}\big[v_i(\tilde{\ell}^m, \tilde{e})\big] \Big] \end{aligned} \tag{6.6}$$

where $\mathbb{E}_{\tilde{e}}\big[v_i(\tilde{\ell}^m, \tilde{e})\big]$ is the expected value of $v_i(\tilde{\ell}(k), \tilde{e}(k))$ after substitution of a given mode switching sequence $\tilde{\ell}^m \in \mathcal{L},\ m = 1, \dots, M$. Now, the final step is to find efficient methods to compute $\tilde{P}\big[L = \tilde{\ell}^m\big]$ and $\mathbb{E}_{\tilde{e}}\big[v_i(\tilde{\ell}^m, \tilde{e})\big]$.

At first, we can reduce the computational complexity of the mode switching uncertainty using a scenario-based algorithm [106]. In general, we may know that some mode switching sequences are more likely to occur than others. Therefore, we can choose to neglect some mode switching sequences that are not likely to occur, resulting in a reduced set $\mathcal{L}^{\mathrm{red}}$ that should be used in (6.6) instead of $\mathcal{L}$. In fact, in this way we often reduce the number of terms in the sum in (6.6) significantly, and thus the computational complexity as well, while still maintaining an adequate cumulative probability of these mode switching sequences to occur.

We can also approximate $\mathbb{E}_{\tilde{e}}\big[v_i(\tilde{\ell}^m, \tilde{e})\big]$ using the approximation method of Chapter 4. Recall that $v_i(\tilde{\ell}^m, \tilde{e})$ is, indeed, maximum of affine terms in $\tilde{e}(k)$ and hence, depending on the probability distribution of $\tilde{e}(k)$, we can apply (4.6) or (4.8) to approximate $\mathbb{E}_{\tilde{e}}\big[v_i(\tilde{\ell}^m, \tilde{e})\big]$. Therefore, instead of using the objective function $J(k)$ in (6.1), we can apply the approximate objective function using the above-mentioned methods and then, solve the approximate switching MPL-MPC optimization problem by means of convex optimization algorithms in the case that all entries of $B_{\mathrm{con}}(k)$ are nonnegative.

Dependent Random Variables

In the second case, $\tilde{e}(k)$ and $\tilde{\ell}(k)$ are dependent, i.e., the mode switching uncertainty depends on the parametric uncertainty. In this case, we cannot apply the approximation methods of the previous case since the mode switching probability $\tilde{P}\big[L = \tilde{\ell}(k) \mid E = \tilde{e}(k)\big]$ cannot be simplified as is done in (6.5). However, if we assume that $\tilde{P}$ as well as the probability density function $f_{\mathcal{E}}$ are modeled or *approximated*[1] by multi-variable piecewise polynomial functions, possibly multiplied by an exponential function (cf. (6.7) below), that are defined on polyhedral regions, then we can proceed as follows [113].

Since $v_i(\tilde{\ell}^m, \tilde{e})$ is a maximum of affine terms in $\tilde{e}(k)$, it is a piecewise affine function (defined on polyhedral regions) of $\tilde{e}(k)$ and as a consequence, it is of the same (but more simple) form as $\tilde{P}$ and $f_{\mathcal{E}}$. So if we combine the regions of $\tilde{P}$, $f_{\mathcal{E}}$, and $v_i(\tilde{\ell}^m, \tilde{e})$ and multiply the piecewise polynomial and piecewise affine functions and the exponential functions, we find that there exists a polyhedral partition $\{\mathcal{R}_t\}_{t=1}^{n_{\mathcal{R}}}$ of $\mathcal{E}$, i.e., the regions $\mathcal{R}_t$ are non-empty and mutually disjoint and their union equals $\mathcal{E}$, such that

$$v_i(\tilde{\ell}^m, \tilde{e})\, f_{\mathcal{E}}(\tilde{e})\, \tilde{P}\big[L = \tilde{\ell}^m \mid E = \tilde{e}\big] = \zeta_{tm}(\tilde{e}) \quad \text{for } \tilde{e} \in \mathcal{R}_t$$

[1] Note that by considering enough polyhedral regions, we can in general approximate the real probability density function arbitrarily close such that the approximate function is nonnegative and its integral over the domain of the real probability density function is equal to 1.

where ζ_{tm} is a function of the form

$$\zeta_{tm}(\tilde{e}) = \sum_{j=1}^{n_{tm}} \xi_{tmj} \left(\prod_{l=1}^{n_{\tilde{e}}} \tilde{e}_l^{k_{tmjl}} \exp(\eta_{tmjl}\tilde{e}_l) \right) \tag{6.7}$$

for some real-valued constants ξ_{imj} and η_{tmjl}, positive integers n_{tm}, and nonnegative integers k_{tmjl}, and where $n_{\tilde{e}}$ is equal to the number of components of $\tilde{e}$.

As a result, (6.4) can be rewritten as follows:

$$\mathbb{E}\big[v_i(\tilde{\ell}^m, \tilde{e})\big] = \sum_{m=1}^{M} \sum_{t=1}^{n_{\mathcal{R}}} \int_{\mathcal{R}_t} \zeta_{tm}(\tilde{e}) \mathrm{d}\tilde{e} \tag{6.8}$$

Since each $\mathcal{R}_t$ is a polyhedron, we can do a substitution of variables by expressing an arbitrary point $\tilde{e} \in \mathcal{R}_t$ as

$$\tilde{e} = \sum_{x_k \in \mathcal{H}^{\text{cen}}} \lambda_k x_k + \sum_{x_k \in \mathcal{H}^{\text{ext}}} \kappa_k x_k + \sum_{x_k \in \mathcal{H}^{\text{fin}}} \mu_k x_k$$

where $\lambda_k \in \mathbb{R}$, $\kappa_k \geq 0$, $\mu_k \geq 0$ and $\sum_k \mu_k = 1$, $\mathcal{H}^{\text{cen}}$ is the set of central generators, i.e., basis vectors for the lineality space associated to the polyhedron $\mathcal{R}_t$ [95], $\mathcal{H}^{\text{ext}}$ is the set of extreme rays, and $\mathcal{H}^{\text{fin}}$ is the set of finite vertices (which can be computed using, e.g., the double-description method introduced in [79]).

As a result

$$\int_{\mathcal{R}_t} \zeta_{tm}(\tilde{e}) \mathrm{d}\tilde{e}$$

reduces to the repeated integration of a polynomial function, possibly multiplied by an exponential, over the domain of λ, κ, and μ. These integrals can be computed analytically.

Remark 6.1.1 Note that this method can also be applied to approximate spline functions or phase-type distributions. Indeed, spline functions are a special type of piecewise polynomial functions [16] and the probability density function of a variable with a phase-type distribution is an exponential function [84, 85]. As a result both can be approximated by multi-variable piecewise polynomial functions, possibly multiplied by an exponential function (cf. (6.7)), that are defined on polyhedral regions. □

6.1.3 Example

We consider a similar example to Example 2.3.1 of Chapter 2 except that here, we have both stochastic switching and stochastic parameters (processing times). Consider the production system of Figure 6.1. This system consists of three machines

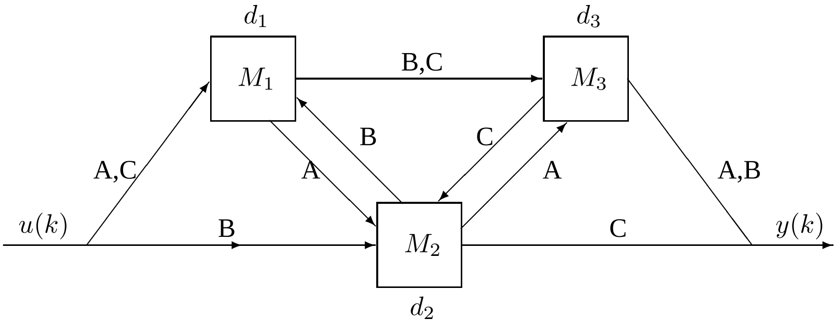


Figure 6.1: A production system with stochastic switching between different recipes and with stochastic system parameters.

M_1, M_2, and M_3. Three products (A, B, and C) can be produced by this system, each with its own recipe or production order.

For product A, the production order is M_1–M_2–M_3, which means that the raw material is fed to machine M_1, where it is processed. The intermediate product is next sent to machine M_2 for further processing, and finally the product is finished in machine M_3. Similarly, for product B, the production order is M_2–M_1–M_3, and for product C the production order is M_1–M_3–M_2. We assume that the type of the k-th product (A, B, or C) is only specified at the start of its production, so that we do not know $\ell(k)$ when computing $u(k)$, e.g., we have a long preprocessing time and the next customer requires particular type of product (user customization).

Each machine starts working as soon as the raw material or the required intermediate products are available, and as soon as the machine is idle (i.e. the previous batch has been finished and has left the machine). We define $u(k)$ as the time instant at which the system is fed for the k-th time, $x_i(k)$ as the time instant at which machine i starts processing for the k-th time, and $y(k)$ as the time instant at which the k-th product leaves the system. Also we assume that all the internal buffers are large enough, and no overflow will occur. We assume that the processing time on machine M_3 is perturbed by noise, i.e., $d_3(k) = 5 + e(k)$ where $e(k) \sim \mathcal{N}(0, 1)$ is a standard normally distributed random variable. The processing times for the other machines (M_1 and M_2) are assumed to be deterministic and given by $d_1 = 4$ and $d_2 = 6$, respectively. Further, the transportation times between the machines are assumed to be negligible. All systems start at time zero, the initial state is given as $x(0) = [0 \quad 0 \quad 0]^T$, the reference signal as $r(k) = 15 + 10k$ for $k = 1, \ldots, 40$, $\lambda = 0.1$, $N_\mathrm{p} = 3$, and $N_\mathrm{c} = 2$.

The mode switching probability from one recipe (mode) to the next one is assumed to be purely stochastic, and given by:

$$
\begin{aligned}
P\big[L(k) = 1 \mid \ell(k-1) = 1, x(k-1), u(k)\big] &= 0.6\\
P\big[L(k) = 1 \mid \ell(k-1) = 2, x(k-1), u(k)\big] &= 0.2\\
P\big[L(k) = 1 \mid \ell(k-1) = 3, x(k-1), u(k)\big] &= 0.2\\
P\big[L(k) = 2 \mid \ell(k-1) = 1, x(k-1), u(k)\big] &= 0.2\\
P\big[L(k) = 2 \mid \ell(k-1) = 2, x(k-1), u(k)\big] &= 0.6\\
P\big[L(k) = 2 \mid \ell(k-1) = 3, x(k-1), u(k)\big] &= 0.2\\
P\big[L(k) = 3 \mid \ell(k-1) = 1, x(k-1), u(k)\big] &= 0.2\\
P\big[L(k) = 3 \mid \ell(k-1) = 2, x(k-1), u(k)\big] &= 0.2\\
P\big[L(k) = 3 \mid \ell(k-1) = 3, x(k-1), u(k)\big] &= 0.6
\end{aligned}
$$

which means that, if we have a specific recipe for the k-th product, then the probability of having the same recipe for the $k+1$-th product is 60%, and the probability to switch to the other recipes is 20%.

Note that since the mode switching depends only on the previous mode $\ell(k-1)$, the mode switching uncertainty $\tilde{\ell}(k)$ and the parametric uncertainty $\tilde{e}(k)$ are independent. Hence, we have the case of independent random variables of Section 6.1.2, and therefore, the scenario-based algorithm and the approximation method of Chapter 4 can be applied in order to simplify the computation.

Figure 6.2 shows the result of a closed-loop simulation for $k = 1, \dots, 40$ where at each event step k, we compute the optimal input sequence by solving a stochastic switching MPL-MPC optimization problem. The optimization has been done using *fmincon* optimizer in Matlab with one initial value since the objective functions (both the exact and the approximate one) are convex and hence, the global minimum will be reached. The “Exact Solution” is obtained using numerical integration and the “Approximation” is obtained using the method explained in Chapter 4. As shown in this figure, the optimization results using the approximation method for $p = 20$ and 30 are closer to the “Exact solution” than the results obtained using the other values of p.

The value of the objective function over the entire simulation period, i.e., $J_{\text{tot}} = \sum_{k=1}^{40} (\max(y(k) - r(k), 0) - \lambda u(k))$, and the relative error[2] between the obtained J_{tot} using numerical integration and the one using the approximation method are reported in Table 6.1.

The values of J_{tot} and the related relative error for each value of p in this table confirm our previous observation that for $p = 20$ and 30 the results obtained using

[2] The relative error is defined here as $\frac{|J_{\text{tot,app}} - J_{\text{tot,ex}}|}{|J_{\text{tot,ex}}|}$ where $J_{\text{tot,app}}$ is obtained using the approximation method and $J_{\text{tot,ex}}$ is obtained using numerical integration.

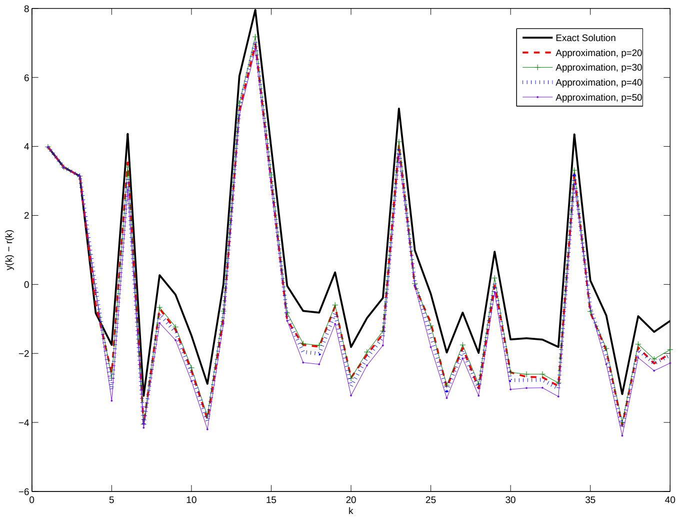


Figure 6.2: Due date error $y(k) - r(k)$ for the closed-loop system using numerical integration and the approximation method of Chapter 4, but without using the scenario-based algorithm.

Method of computing J_{tot}	J_{tot}	Relative error of J_{tot}
Numerical integration	-769.7146	
Approximation method:		
$p = 20$	-767.0465	0.35%
$p = 30$	-767.2644	0.32%
$p = 40$	-766.9419	0.36%
$p = 50$	-766.5062	0.42%

Table 6.1: The obtained value of J_{tot}, and the relative error of J_{tot} using numerical integration and the approximation method with different values of p to compute the expected value in the objective function.

the approximation method are the closest to the results obtained using numerical integration.

The computation time[3] of the total closed-loop optimization using numerical integration is 176318 s compared to 2518 s using the approximation method without

[3] These times are obtained running Matlab 7.11.0 (R2010b) on a 2.33 GHz Intel Core Duo E655 processor.

applying the scenario-based algorithm. Note that in this example, we do not apply the scenario-based algorithm since, due to the rather small prediction horizon, the computation time will not decrease considerably (both computation times with and without scenario-based algorithm are very close[4]) and the difference between the performance with and without the scenario-based algorithm is negligible. It is obvious that the approximation method decreases the computation time significantly. Note that although there is not a big difference between the computation times using the scenario-based algorithm and not using it, this difference will be much more significant if we increase the prediction horizon N_p. Recall that the number of possible mode switching sequences over the whole prediction horizon is $(n_L)^{N_\mathrm{p}}$ where n_L is the total number of possible modes and using the scenario-based algorithm, we will only analyze the most probable mode switching sequences with a cumulative probability of, say, at least 80%. Hence, n_L, which is 27 in our example, reduces to n_red, which is 19 for $N_\mathrm{p} = 3$, and if we choose $N_\mathrm{p} = 4$, then $n_L = 81$ versus $n_\mathrm{red} = 33$ and the computation time will decrease remarkably.

Accordingly, we can conclude that for the case of independent random variables, the combination of the two approximation methods, i.e., scenario-based algorithm and the approximation method of Chapter 4, will improve the computation time considerably while it obtains results comparable to the ones using numerical integration.

6.2 Approximate MPC for Stochastic MMPS Systems

As indicated in Chapter 2, (stochastic) MMPS systems describes a large class of discrete-event and hybrid systems and to control such systems, MPC can be applied. Similar to the stochastic MPL-MPC optimization problem, solving the stochastic MMPS-MPC optimization problem is quite complex due to the stochastic nature of the system. In this section, we show that the approximation method of Chapter 4 can be used to decrease this complexity.

6.2.1 Problem Statement

Recall from Section 3.1.5 that the stochastic MPC-MMPS problem can be defined as follows:

$$\begin{aligned} &\min_{\tilde{u}(k)} \quad \tilde{J}(k) \\ &\text{subject to:} \quad c(\tilde{u}(k), \mathbb{E}[\tilde{y}(k)], k) \leq 0. \end{aligned} \tag{6.9}$$

[4] The computation time of the closed-loop optimization using the approximation method in combination with the scenario-based algorithm is 2508 s.

where[5]

$$\tilde{J}(k) = \sum_{i=1}^{n_y \cdot N_\mathrm{p}} \mathbb{E}[\max(\tilde{y}_i(k) - \tilde{r}_i(k), 0)] - \lambda \sum_{l=1}^{n_u \cdot N_\mathrm{p}} u_l(k)$$

with the prediction horizon N_p and where c is a possibly non-convex function of the input signal and the expected value of the output signal[6]. Since both $x(k+j)$ and $\tilde{y}(k)$ are MMPS functions of $x(k-1), u(k), \dots, u(k+j), e(k), \dots, e(k+j),\ j = 0, \dots, N_\mathrm{p}$, we conclude that $\tilde{J}(k)$ and (in general) $c(\tilde{u}(k), \mathbb{E}[\tilde{y}(k), k)]$ consists of the expected value of MMPS functions. Hence, we need to find an efficient method to compute this expected value in order to solve the optimization problem (6.9). To this end, consider the following proposition:

Proposition 6.2.1 ([30]) *Any scalar-valued MMPS function can be rewritten into a min-max canonical form $g(x) = \min_{i=1,\dots,K} \max_{j=1,\dots,n_i}(\alpha_{ij}^T x + \beta_{ij})$ or into a max-min canonical form $g(x) = \max_{i=1,\dots,L} \min_{j=1,\dots,m_i}(\gamma_{ij}^T x + \delta_{ij})$ for some integers $K, L, n_1, \dots, n_K, m_1, \dots, m_L$, real numbers β_{ij}, δ_{ij}, and vectors α_{ij}, γ_{ij}.*

Hence, we can rewrite the objective function $\tilde{J}(k)$ in (6.9) in its canonical form as follows:

$$\tilde{J}(k) = \mathbb{E}[\max_{i=1,\dots,\ell} \min_{j=1,\dots,m_i} (\alpha_{ij}(k) + \beta_{ij}^T \tilde{u}(k) + \gamma_{ij}^T \tilde{e}(k))] \tag{6.10}$$

where $\alpha_{ij}(k)$ is affine in $x(k-1)$ and $\tilde{r}(k)$. To shorten the notations, let

$$g(\tilde{u}(k), \tilde{e}(k)) = \max_{i=1,\dots,\ell} \min_{j=1,\dots,m_i} \left(\alpha_{ij}(k) + \beta_{ij}^T \tilde{u}(k) + \gamma_{ij}^T \tilde{e}(k)\right),$$

so $\tilde{J}(k) = \mathbb{E}[g(\tilde{u}(k), \tilde{e}(k))]$. Considering the following proposition and corollary, $g(\tilde{u}(k), \tilde{e}(k))$ can be written as a difference of two convex functions.

Proposition 6.2.2 ([67, 116]) *The function $g(x) = \min_{i=1,\dots,K} \max_{j=1,\dots,n_i} \delta_{ij}(x)$, where $\delta_{ij}(x) = \alpha_{ij}^T x + \beta_{ij}$ is an affine function in x, can be written as a difference of two convex functions, i.e., $g(x) = p(x) - q(x)$ where $p(x)$ and $q(x)$ are defined as follows:*

$$p(x) = \sum_{i=1}^{K} \max_{j=1,\dots,n_i} \delta_{ij}(x) \tag{6.11}$$

[5] Recall that we can use any other combination of input and output objective functions in (3.2) and for a time-driven system, the input objective function does not have the minus sign.

[6] Later on, at the end of Section 6.2.2, we will consider the case in which – after elimination of $\mathbb{E}[\tilde{y}(k)]$ – c is convex in $\tilde{u}(k)$.

$$
\begin{aligned}
q(x) &= p(x) - g(x) \\
&= p(x) - \min_{i=1,\dots,K} \max_{j=1,\dots,n_i} \delta_{ij}(x) \\
&= p(x) + \max_{i=1,\dots,K} \big(- \max_{j=1,\dots,n_i} \delta_{ij}(x) \big) \\
&= \max_{i=1,\dots,K} \big(p(x) - \max_{j=1,\dots,n_i} \delta_{ij}(x) \big). \\
&= \max_{i=1,\dots,K} \Big(\sum_{\substack{i'=1 \\ i' \neq i}}^{K} \max_{j=1,\dots,n_{i'}} \delta_{i'j}(x) \Big) \\
&= \max_{l=1,\dots,K} \max_{(j_1,\dots,j_{i-1},j_{i+1},\dots,j_K) \in \mathcal{C}(n_1,\dots,n_{i-1},n_{i+1},\dots,n_K)} \sum_{\substack{i'=1 \\ i' \neq i}}^{K} \delta_{i' j_{i'}}(x). \qquad (6.12)
\end{aligned}
$$

where the set $\mathcal{C}(\ell_1,\dots,\ell_m)$ for some integers $m, \ell_1, \dots, \ell_m$ is defined as

$$
\mathcal{C}(\ell_1,\dots,\ell_m) = \{(q_1,\dots,q_m) | q_k \in \{1,2,\dots,\ell_k\} \ \text{ for } \ k = 1,\dots,m\}.
$$

Based on this proposition, the following corollary is concluded:

Corollary 6.2.3 *The function $g(x) = \max_{i=1,\dots,L} \min_{j=1,\dots,m_i} l_{ij}(x)$, where $l_{ij}(x) = \gamma_{ij}^T x + \xi_{ij}$ is an affine function in x, can be written as $g(x) = s(x) - r(x)$ where $s(x)$ and $r(x)$ are both convex functions defined as follows:*

$$
\begin{aligned}
r(x) &= - \sum_{i=1}^{L} \min_{j=1,\dots,m_i} l_{ij}(x) \\
&= \sum_{i=1}^{L} \max_{j=1,\dots,m_i} (-l_{ij}(x)) \qquad (6.13) \\
s(x) &= r(x) + g(x) \\
&= r(x) + \max_{i=1,\dots,L} \min_{j=1,\dots,m_i} l_{ij}(x) \\
&= \max_{i=1,\dots,L} \big(r(x) + \min_{j=1,\dots,m_i} l_{ij}(x) \big) \\
&= \max_{i=1,\dots,L} \big(r(x) - \max_{j=1,\dots,m_i} (-l_{ij}(x)) \big) \\
&= \max_{i=1,\dots,L} \big(\sum_{\substack{i'=1 \\ i' \neq i}}^{L} \max_{j=1,\dots,m_{i'}} (-l_{i'j}(x)) \big)
\end{aligned}
$$

$$= \max_{l=1,\dots,L} \max_{(j_1,\dots,j_{i-1},j_{i+1},\dots,j_L)\in\mathcal{C}(m_1,\dots,m_{i-1},m_{i+1},\dots,m_L)} \sum_{\substack{i'=1\\ i'\neq i}}^{L} (-l_{i' j_{i'}}(x)). \tag{6.14}$$

Note that the last equality is obtained using the distributive property of addition w.r.t. maximization in which $\mathcal{C}(m_1,\dots,m_{i-1},m_{i+1},\dots,m_L)$ *is defined similarly as before.*

Therefore, we can rewrite (6.10) as:

$$\begin{aligned}\tilde{J}(k) &= \mathbb{E}[g(\tilde{u}(k),\tilde{e}(k))]\\ &= \mathbb{E}[s(\tilde{u}(k),\tilde{e}(k)) - r(\tilde{u}(k),\tilde{e}(k))]\\ &= \mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))] - \mathbb{E}[r(\tilde{u}(k),\tilde{e}(k))]\end{aligned} \tag{6.15}$$

where $s(\tilde{u}(k),\tilde{e}(k))$ and $r(\tilde{u}(k),\tilde{e}(k))$ are defined as given in Corollary 6.2.3, and where the last equality comes from the fact that $\mathbb{E}[\cdot]$ is a linear operator.

Note that the objective function $\tilde{J}(k)$ in (6.15) results in a non-convex optimization problem in its current structure. Now to solve the optimization problem (6.9) it is only left to compute the expected values in (6.15). Note that $s(\tilde{u}(k),\tilde{e}(k))$ and $r(\tilde{u}(k),\tilde{e}(k))$ both consist of a maximization of affine terms and hence, are max-plus-scaling functions. Therefore, our aim is to find an efficient way to compute the following general expression:

$$\mathbb{E}[\max_{j=1,\dots,n}(\xi_j + \gamma_j^T\tilde{e})] \tag{6.16}$$

where $\xi_j \in \mathbb{R}$ and by assumption is an affine term in $\tilde{u}(k)$, $\gamma_j \in \mathbb{R}^{n_{\tilde{e}}}$, and $\tilde{e} \in \mathbb{R}^{n_{\tilde{e}}}$ is a random variable with the given probability density function $f(\tilde{e})$. Note that the dependence of ξ_j, γ_j, and $\tilde{e}$ on $\tilde{u}(k)$ and/or k is dropped for the sake of brevity. In general, $\mathbb{E}[\max_{j=1,\dots,n}(\xi_j + \gamma_j^T\tilde{e})]$ can be computed using numerical or analytic integration (cf. Section 3.1.3); however, since these methods are complex and time-consuming, we propose the approximation method of Chapter 4 as an alternative solution.

6.2.2 Approximation Approach

In the optimization problem (6.9), we minimize $\tilde{J}(k)$, which indeed leads to the minimization of $\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]$ and maximization of $\mathbb{E}[r(\tilde{u}(k),\tilde{e}(k))]$ in (6.15). Considering the approximation method of Chapter 4, we can approximate $\tilde{J}(k)$ by its upper bound. Hence, we need to have an upper bound for $\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]$ and a lower bound for $\mathbb{E}[r(\tilde{u}(k),\tilde{e}(k))]$. Let us consider again the general function $\mathbb{E}[\max_{j=1,\dots,n}(\xi_j + \gamma_j^T\tilde{e})]$ in (6.16). The upper bound of (6.16) can be obtained easily by using (4.4) in Proposition 4.2.3. To compute the lower bound of (6.16), we can apply Theorem 4.2.2 since $\max(\cdot)$ is a convex function. According to Jensen's inequality for convex functions, we have:

$$\max(\mathbb{E}[x_1], \ldots, \mathbb{E}[x_n]) \leq \mathbb{E}\big[\max(x_1, \ldots, x_n)\big].$$

Therefore, the lower and the upper bound of $\mathbb{E}[\max_{j=1,\ldots,n}(\xi_j + \gamma_j^T \tilde{e})]$ can be defined as follows:

$$\mathfrak{L}\Big(\mathbb{E}[\max_{j=1,\ldots,n}(\xi_j + \gamma_j^T \tilde{e})]\Big) = \max_{j=1,\ldots,n}(\mathbb{E}[\xi_j + \gamma_j^T \tilde{e}]) \tag{6.17}$$

$$\mathfrak{U}\Big(\mathbb{E}[\max_{j=1,\ldots,n}(\xi_j + \gamma_j^T \tilde{e})]\Big) = \left(\sum_{j=1}^{n} \mathbb{E}\big[\big(\xi_j + \gamma_j^T \tilde{e} - L\big)^p\big]\right)^{1/p} + L \tag{6.18}$$

for a properly defined L and they are both convex in $\tilde{u}$ (cf. Section 4.4). Recall that if the distribution of $\tilde{e}$ is not preserved under summation, the approximation function (4.8) can be used in place of (6.18).

In the following example, we show how Corollary 6.2.3 and the approximation method can be applied.

Example 6.2.4

Let[7] $g(\tilde{e}) = \max(\min(\gamma_1 \tilde{e} + \xi_1, \gamma_2 \tilde{e} + \xi_2), \min(\gamma_3 \tilde{e} + \xi_3, \gamma_4 \tilde{e} + \xi_4))$, where $\tilde{e}$ is a stochastic vector with independent elements, $\xi_j \in \mathbb{R}$, and $\gamma_j \in \mathbb{R}^{n_{\tilde{e}}}$ for $j = 1, \ldots, 4$. Following Corollary 6.2.3, we can rewrite $g(\tilde{e})$ as $g(\tilde{e}) = s(\tilde{e}) - r(\tilde{e})$ where s and r are convex functions defined as follows (cf. (6.13) and (6.14)):

$$\begin{aligned}
r(\tilde{e}) &= \max(-\gamma_1^T \tilde{e} - \xi_1, -\gamma_2^T \tilde{e} - \xi_2) + \max(-\gamma_3^T \tilde{e} - \xi_3, -\gamma_4^T \tilde{e} - \xi_4) \\
s(\tilde{e}) &= r(\tilde{e}) + g(\tilde{e}) \\
&= \max\Big(r(\tilde{e}) + \min(\gamma_1 \tilde{e} + \xi_1, \gamma_2 \tilde{e} + \xi_2), r(\tilde{e}) + \min(\gamma_3 \tilde{e} + \xi_3, \gamma_4 \tilde{e} + \xi_4)\Big) \\
&= \max\Big(r(\tilde{e}) - \max(-\gamma_1 \tilde{e} - \xi_1, -\gamma_2 \tilde{e} - \xi_2), \\
&\qquad\quad r(\tilde{e}) - \max(-\gamma_3 \tilde{e} - \xi_3, -\gamma_4 \tilde{e} - \xi_4)\Big) \\
&= \max\Big(\max(-\gamma_1^T \tilde{e} - \xi_1, -\gamma_2^T \tilde{e} - \xi_2), \max(-\gamma_3^T \tilde{e} - \xi_3, -\gamma_4^T \tilde{e} - \xi_4)\Big) \\
&= \max\big(-\gamma_1^T \tilde{e} - \xi_1, -\gamma_2^T \tilde{e} - \xi_2, -\gamma_3^T \tilde{e} - \xi_3, -\gamma_4^T \tilde{e} - \xi_4\big).
\end{aligned}$$

Hence, an upper bound for $\mathbb{E}[g(\tilde{e})]$ can be obtained by computing an upper bound for $\mathbb{E}[s(\tilde{e})]$ and a lower bound for $\mathbb{E}[r(\tilde{e})]$ as follows:

$$\mathfrak{U}\Big(\mathbb{E}[s(\tilde{e})]\Big) = \Big(\sum_{j=1}^{4} \mathbb{E}[(-\gamma_j^T \tilde{e} - \xi_j)^p]\Big)^{1/p}$$

[7] The dependency of $g(\cdot)$ on $\tilde{u}$ is not mentioned explicitly here, since it is not related to this example.

$$\begin{aligned}\mathfrak{L}\Big(\mathbb{E}[r(\tilde{e})]\Big) = \max\big(-\gamma_1^T\mathbb{E}[\tilde{e}]-\xi_1,\ -\gamma_2^T\mathbb{E}[\tilde{e}]-\xi_2\big)\\ +\max\big(-\gamma_3^T\mathbb{E}[\tilde{e}]-\xi_3,\ -\gamma_4^T\mathbb{E}[\tilde{e}]-\xi_4\big).\end{aligned}$$

□

Consequently, instead of minimizing the objective function (6.15), we will minimize its upper bound $\tilde{J}_{\mathrm{up}}(k)$, which can be obtained using (6.17) and (6.18). Considering (6.10), let

$$\begin{aligned}s(\tilde{u}(k),\tilde{e}(k)) &= \max_{j=1,\ldots,n}(a_j(k)+b_j^T(k)\tilde{e}(k))\\ r(\tilde{u}(k),\tilde{e}(k)) &= \sum_{i=1}^{M}\max_{l=1,\ldots,m_i}(c_{il}(k)+d_{il}^T(k)\tilde{e}(k))\end{aligned}$$

where $b(k)$ and $d(k)$ are scalar vectors and $a(k)$ and $c(k)$ are affine terms in $\tilde{u}(k)$. In this way, the optimization problem (6.9) turns into a sequence of convex problems as follows:

$$\begin{aligned}&\min_{\tilde{u}(k)}\ \tilde{J}_{\mathrm{up}}(k)\\ &=\min_{\tilde{u}(k)}\Big(\mathfrak{U}\Big(\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]\Big)-\mathfrak{L}\Big(\mathbb{E}[r(\tilde{u}(k),\tilde{e}(k))]\Big)\Big)\\ &=\min_{\tilde{u}(k)}\Big(\mathfrak{U}\Big(\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]\Big)-\mathfrak{L}\Big(\mathbb{E}\Big[\sum_{i=1}^{M}\max_{l=1,\ldots,m_i}(c_{il}(k)+d_{il}^T(k)\tilde{e}(k))\Big]\Big)\Big)\\ &=\min_{\tilde{u}(k)}\Big(\mathfrak{U}\Big(\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]\Big)-\sum_{i=1}^{M}\max_{l=1,\ldots,m_i}(\mathbb{E}[c_{il}(k)+d_{il}^T(k)\tilde{e}(k)])\Big)\\ &=\min_{\tilde{u}(k)}\Big(\mathfrak{U}\Big(\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]\Big)+\sum_{i=1}^{M}\min_{l=1,\ldots,m_i}(-\mathbb{E}[c_{il}(k)+d_{il}^T(k)\tilde{e}(k)])\Big)\\ &=\min_{\tilde{u}(k)}\Big(\mathfrak{U}\Big(\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]\Big)\\ &\qquad+\min_{(\ell_1,\ldots,\ell_M)\in\mathcal{C}(m_1,\ldots,m_M)}\sum_{i=1}^{M}(-\mathbb{E}[c_{il_i}(k)+d_{il_i}^T(k)\tilde{e}(k)])\Big)\\ &=\min_{\tilde{u}(k)}\min_{(\ell_1,\ldots,\ell_M)\in\mathcal{C}(m_1,\ldots,m_M)}\Big(\mathfrak{U}\Big(\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]\Big)\\ &\qquad+\sum_{i=1}^{M}(-\mathbb{E}[c_{il_i}(k)+d_{il_i}^T(k)\tilde{e}(k)])\Big)\\ &=\min_{(\ell_1,\ldots,\ell_M)\in\mathcal{C}(m_1,\ldots,m_M)}\min_{\tilde{u}(k)}\Big(\mathfrak{U}\Big(\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]\Big)\end{aligned}$$

$$+\sum_{i=1}^{M}(-\mathbb{E}[c_{il_i}(k)+d_{il_i}^{T}(k)\tilde{e}(k)])\Big). \qquad (6.19)$$

where $\mathcal{C}(m_1,\ldots,m_M)$ is defined in the same way as the one in Proposition 6.2.2. Note that in (6.19), $\min_{\tilde{u}(k)}(\mathfrak{U}\Big(\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]\Big)+\sum_{i=1}^{M}(-\mathbb{E}[c_{il_i}(k)+d_{il_i}^{T}(k)\tilde{e}(k)])$ is a convex optimization problem since both $\sum_{i=1}^{M}(-\mathbb{E}[c_{il_i}(k)+d_{il_i}^{T}(k)\tilde{e}(k)])$ and $\mathfrak{U}\Big(\mathbb{E}[s(\tilde{u}(k),\tilde{e}(k))]\Big)$ are convex in $\tilde{u}(k)$. As mentioned before, the constraint function $c(\cdot)$ in (6.9) is not convex in general. However, in the case that – after the elimination of $\mathbb{E}[\tilde{y}(k)]$ – $c(\cdot)$ is convex in $\tilde{u}(k)$, we obtain a convex optimization problem, which can be solved efficiently for each $(\ell_1,\ldots,\ell_M)\in\mathcal{C}(m_1,\ldots,m_M)$ using convex optimization algorithms such as an interior point method [83]. Then, to obtain the final optimal solution, we choose the minimum of the obtained results.

6.2.3 Example

In this section, we study an example that is similar to the one in [81]. In our example the noise vector is normally distributed while in [81] a bounded noise without any knowledge about its probability density function has been considered and hence, the worst-case optimization problem was solved.

Consider a room with a base heat source and an additional controlled heat source. Let u be the contribution to the increase in room temperature per time unit caused by the controlled heat source (so $u\geq 0$). For the base heat source, this value is assumed to be constant and equal to 1. The temperature in the room is assumed to be uniform over the room and its evaluation is formulated as follows:

$$\dot{T}(t)=\alpha(T(t))T(t)+u(t)+1+e_1(t)\ ,$$

where $e_1(t)$ denotes the disturbance at time t, and $\alpha(T)$ is the temperature coefficient of the room which is assumed to have a piecewise constant form, i.e., $\alpha(T)=-1/2$ if $T<0$ and $\alpha(T)=-1$ if $T\geq 0$. Let $y(t)$ denote the noisy measurement of the room temperature defined as $y(t)=T(t)+e_2(t)$ where $e_2(t)$ represents the measurement noise at time t. Here, we assume that both $e_1(t)$ and $e_2(t)$ have a standard normal distribution, i.e., $e_1(k),e_2(k)\sim\mathcal{N}(0,1)$.

Using the Euler discretization method, with a sample time of 1 s and denoting the state by $x(k)=T(k)$, we get the following discrete-time PWA system:

$$x(k+1)=\begin{cases}1/2x(k)+u(k)+e_1(k)+1 & \text{if } x(k)<0\\ u(k)+e_1(k)+1 & \text{if } x(k)\geq 0\end{cases} \qquad (6.20)$$

$$y(k) = x(k) + e_2(k). \tag{6.21}$$

with the following constraints on the input:

$$-4 \leq \Delta u(k) = u(k+1) - u(k) \leq 4 \quad \text{and} \quad u(k) \geq 0 \; \text{ for all } k.$$

The equivalent MMPS representation of (6.20)–(6.21) is the following:

$$\begin{aligned} x(k+1) &= \min(1/2x(k) + u(k) + e_1(k) + 1, u(k) + e_1(k) + 1), \\ y(k+1) &= x(k+1) + e_2(k+1). \end{aligned}$$

Since at sample step k the input $u(k)$ has no influence on $y(k)$, we choose $N_\mathrm{p} = 3, N_\mathrm{c} = 2$, $\tilde{y}(k) = [y(k+1)\ y(k+2)]^T$, $\tilde{r}(k) = [r(k+1)\ r(k+2)]^T$, $\tilde{u}(k) = [u(k)\ u(k+1)]^T$. Let the uncertainty vector be $e(k) = [e_1(k)\ e_2(k+1)]^T$. Therefore, $\tilde{e}(k) = [e^T(k)\ e^T(k+1)]^T$.

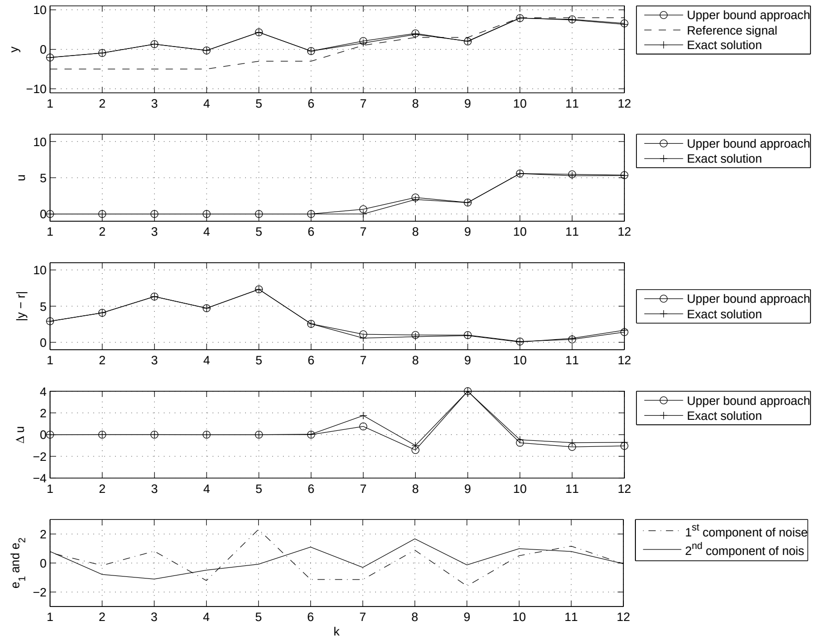


Figure 6.3: Closed-loop optimization results of the stochastic MPC-MMPS problem.

Considering the given choices for the output and input objective functions in (3.2), we define $\tilde{J}(k)$ as follows:

$$\tilde{J}(k) = \mathbb{E}\big[\|\tilde{y}(k) - \tilde{r}(k)\|_\infty + \lambda\|\tilde{u}(k)\|_1\big].$$

Since $u(k) \geq 0$, we have $\|\tilde{u}(k)\|_1 = u(k) + u(k+1)$ and therefore, we obtain the following max-min expression for $\tilde{J}(k)$:

$$\begin{aligned}\tilde{J}(k) &= \mathbb{E}\big[\max(|\tilde{y}(k) - \tilde{r}(k)|) + \lambda(u(k) + u(k+1))\big] \\ &= \mathbb{E}\big[\max\big(y(k+1) - r(k+1) + \lambda u(k) + \lambda u(k+1), \\ &\qquad y(k+2) - r(k+2) + \lambda u(k) + \lambda u(k+1), \\ &\qquad - y(k+1) + r(k+1) + \lambda u(k) + \lambda u(k+1), \\ &\qquad - y(k+2) + r(k+2) + \lambda u(k) + \lambda u(k+1)\big)\big]. \end{aligned} \tag{6.22}$$

We compute the closed-loop MPC controller by minimizing the upper bound of the objective function (6.22) (as shown in (6.19)) over the simulation period $[1, 20]$, with $\lambda = 0.01$, $x(0) = -6$, and $u(-1) = 0$. The reference signal is given as $\{r(k)\}_{k=1}^{20} = \{-5, -5, -5, -5, -3, -3, 1, 3, 3, 8, 8, 8, 8, 10, 10, 7, 7, 7, 4, 3\}$.

Figure 6.3 shows the results of the closed-loop simulation in which the results of the “Exact solution” obtained using numerical integration are compared to those obtained using the upper bound approach in (6.19). The optimization has been done using the *fmincon* optimizer in Matlab with one initial value since the objective functions (both the exact and the approximate one) are convex and hence, the global minimum will be reached. The top plot shows the reference signal, the output of the system using numerical integration (“Exact solution”), and the output of the system using the upper bound approach. The next two plots present the optimal input sequence and the tracking error, respectively, using numerical integration (“Exact solution”) and the upper bound approach. The forth plot illustrates $\Delta u(k) = u(k+1) - u(k)$ and shows that in both approaches, the constraints on $\Delta u(k)$ are satisfied. The last plot shows the mean value of each component of the noise vector. Based on the above plots, the results of the upper bound approach in (6.19) are very close to the ones from numerical integration. To compute the objective function value for the total simulation period, we use 20 different noise realizations and then report the mean value of the objective function, i.e., $\bar{J}_{\text{tot}} = \frac{1}{20}\sum_{i=1}^{20} J_{\text{tot},i}$ with $J_{\text{tot},i} = \sum_{k=1}^{20}\Big(\max(y^{(i)}(k) - r(k), 0) - \lambda u^{(i)}(k)\Big)$ using the input and output results ($y^{(i)}$ and $u^{(i)}$) for the i-th noise realization. Hence, using numerical integration, we obtain $\bar{J}_{\text{tot}} = 7.4278$ with variance 1.4654 and using the upper bound approach, we have $\bar{J}_{\text{tot}} = 7.4488$ with variance 1.4660. Accordingly, the relative error[8] between the objective functions using numerical integration and the upper bound approach is 0.3%. Moreover, the computation time[9] using the up-

[8] The relative error is defined here as $\frac{|J_{\text{tot,up}} - J_{\text{tot,ex}}|}{|J_{\text{tot,ex}}|}$ where $J_{\text{tot,up}}$ is obtained using the upper bound approach and $J_{\text{tot,ex}}$ is obtained using numerical integration.

[9] These times are obtained running Matlab 7.11.0 (R2010b) on a 2.33 GHz Intel Core Duo E655 processor.

per bound approach is 323.02 s compared to 23447 s using the numerical integration (i.e., the "Exact solution"). All these observations confirm that, in this example, the upper bound approach is a reliable method to be applied. Hence, we can solve the stochastic MPC-MMPS problem using (6.19) in order to decrease the computation time while still guaranteeing a good performance.

6.3 Summary

In this chapter, we have extended the application of the approximation method of Chapter 4 to two other classes of MPL systems, namely stochastic switching MPL systems and stochastic MMPS systems. Applying MPC to control these systems, in general, results in a complex and time-consuming optimization problem due to the stochastic properties of these systems.

To tackle these difficulties in the stochastic switching MPL-MPC optimization problem, we have considered the joint probability distribution with both discrete and continuous random variables, which are related to stochastic mode switching and stochastic system parameters, respectively. In the case that these random variables are independent, we can split the joint probability function into two parts and we can apply a combination of two approximation methods, viz., the approximation method of Chapter 4 and the scenario-based algorithm of [106], to simplify the problem. In the case of dependent random variables, we can approximate the probability density function and the switching probability by multi-variable piecewise polynomial functions to obtain an analytic expression for the objective function. In both cases, the approximation approaches simplify the problem considerably.

In the stochastic MMPS-MPC optimization problem, the objective function is defined as an expected value of an MMPS function, which needs to be computed efficiently. As a solution, we proposed to write the MMPS objective function as a difference of two convex functions and then, optimize the upper bound of this new objective function instead of the objective function itself. This leads to computing the upper bound of one of the convex functions by applying an approximation method of Chapter 4, and the lower bound of the other convex function. These bounds results in an analytic expressions in the case the probability distribution of the noise vector has finite moments with a closed-form expression. We have seen in a case study that the upper bound approach decreases the computational complexity and the computation time considerably while we still have a good performance that is comparable to the one using numerical integration.

Chapter 7

Min-Max Optimization and Approximate Stochastic Optimization for Max-Min-Plus-Scaling Systems

In this chapter, we discuss the min-max optimization and stochastic optimization of MMPS systems[1]. We consider different system structures and for each structure, a solution approach is proposed. Moreover, we apply these solution approaches to solve the filtering problem and the reference tracking problem as two applications of this type of systems.

7.1 Problem Statement

Consider the following type of optimization problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \max_{y\in\mathbb{R}^m} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \leq 0 \end{aligned} \tag{7.1}$$

where F is a scalar max-min-plus-scaling (MMPS) function of x and y and G is a vector-valued MMPS function of x and y. For a given $x \in \mathbb{R}^n$, define:

$$\Omega(x) = \left\{ \begin{aligned} \max_{y\in\mathbb{R}^m} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \leq 0 \end{aligned} \right. \tag{7.2}$$

[1] This chapter is inspired by the research visit at the department of Mechanical and Aerospace Engineering of the university of California, San Diego, hosted by Prof.dr. William McEneaney.

provided that the problem is feasible and the maximum is finite. Hence, (7.1) is equivalent to

$$\min_{x \in \mathrm{dom}(\Omega)} \quad \Omega(x) \tag{7.3}$$

where $\mathrm{dom}(\Omega)$ indicates the domain of definition of Ω. Therefore, for (7.3), and hence (7.1), to have a finite solution, $\mathrm{dom}(\Omega)$ should be non-empty and the function Ω should be bounded from below on its domain.

We study the optimization problem (7.1) for three different settings: a non-stochastic setting with maximization and minimization, a stochastic setting with minimization only, and a stochastic setting with both maximization and minimization. These settings are presented in the following three cases.

7.1.1 Case I

We consider an uncertain min-max optimization problem in which both variables x and y belong to convex polytopes and hence, are bounded. Accordingly, the optimization problem can be defined as follows:

$$\begin{aligned} \min_{x \in \mathcal{B}_x} \max_{y \in \mathcal{B}_y} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \leq 0. \end{aligned} \tag{7.4}$$

where $\mathcal{B}_x$ and $\mathcal{B}_y$ are convex polytopes.

7.1.2 Case II

We consider a minimization problem with a stochastic random variable ω that has a given probability density function. Based on our solution approach of Section 6.2.2, in order to obtain an analytic expression for the approximation of the expected value, we only consider probability distributions for which the random variables have finite moments and a closed-form expression of these moments exists, such as the uniform distribution, normal distribution, Beta distribution, etc. (cf. Chapter 4). The minimization problem can be then formulated as follows:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \mathbb{E}_\omega[F(x,\omega)] \\ \text{s.t.} \quad & \mathbb{E}_\omega[G(x,\omega)] \leq 0 \end{aligned} \tag{7.5}$$

where $\mathbb{E}_\omega[\cdot]$ is the expected value with respect to ω.

7.1.3 Case III

In the last case, we have a min-max optimization problem with a stochastic variable ω with a given probability density function from distributions with finite moments for which a closed-form expression exists. Hence, the optimization problem can be defined as

$$\begin{aligned}\min_{x\in\mathbb{R}^n}\ \max_{y\in\mathbb{R}^m}\quad & \mathbb{E}_\omega[F(x,y,\omega)]\\ \text{s.t.}\quad & \mathbb{E}_\omega[G(x,y,\omega)]\leq 0\end{aligned} \tag{7.6}$$

which is, indeed, the most general case (in fact Case I and Case II are special subcases of this case).

7.2 Solution Approaches

In this section we present solution approaches to each of the above mentioned cases.

7.2.1 Solution Approach for Case I

We show that the optimization problem (7.4) can be transformed into a mixed-integer linear programming (MILP) optimization problem. To this end, we proceed as follows.

First we transform F into a mixed-integer linear form. Since F is an MMPS function of its arguments, it is equivalent to a continuous piecewise affine (PWA) function [49, 88], i.e.,

$$F(x,y) = A_i y + B_i x + c_i \quad \text{if} \quad (x,y)\in\mathcal{C}_i,$$

where $\mathcal{C}_i$ is a convex polytope of the form $\mathcal{C}_i = \{(x,y) : S_i y + R_i x \leq q_i\}$ for $i = 1\ldots,s$. This PWA function is well-defined if and only if $\mathcal{C} = \mathcal{B}_x \times \mathcal{B}_y$ can be partitioned in, say s, polytopes $\mathcal{C}_i$ such that

$$\text{int}(\mathcal{C}_i)\cap\text{int}(\mathcal{C}_j)=\emptyset,\ \ \forall i,j\in\{1,\ldots,s\}\ \text{ with }\ i\neq j, \tag{7.7}$$

$$\bigcup_{i=1}^{s}\mathcal{C}_i=\mathcal{C} \tag{7.8}$$

where $\text{int}(\cdot)$ denotes the interior of $\mathcal{C}_i$. Now, we define binary variables $\delta_i\in\{0,1\}$ for $i=1,\ldots,s$ such that

$$[\delta_i = 1] \Leftrightarrow [(x,y)\in\text{int}(\mathcal{C}_i)]. \tag{7.9}$$

where for points (x,y) on the boundary of several regions, one δ_i is taken equal to 1 and the rest are set equal to 0. The conditions (7.7)-(7.9) are equivalent to [8]:

$$S_i y + R_i x - q_i \leq M_i^*(1-\delta_i) \tag{7.10}$$

$$\sum_{i=1}^{s}\delta_i = 1, \tag{7.11}$$

where $M_i^* = \max_{(x,y)\in\mathcal{C}_i}(S_i y + R_i x - q_i)$. Note that M_i^* is finite since x and y belong to a polytope. In fact, M_i^* can be obtained by solving a linear program. Now, we can write F as

$$F(x,y) = \begin{cases} A_1 y + B_1 x + c_1 & \text{if } \ \delta_1 = 1, \\ \vdots & \\ A_s y + B_s x + c_s & \text{if } \ \delta_s = 1, \end{cases} \tag{7.12}$$

and since for each point (x,y) exactly one δ_i is equal to 1 and the others are 0, we can write (7.12) as

$$F(x,y) = \sum_{i=1}^{s}(A_i y + B_i x + c_i)\delta_i \tag{7.13}$$

which is non-linear due to the presence of the product between the binary variables δ_i and the real-valued variables x and y. However, it can be translated into equivalent mixed-integer linear inequalities. If we define

$$t_i = (A_i y + B_i x + c_i)\delta_i, \tag{7.14}$$

then (7.13) reduces to

$$F(x,y) = \sum_{i=1}^{s} t_i. \tag{7.15}$$

The non-linear equation (7.14) can be transformed into a mixed-integer linear form as follows [8]. Define

$$M_i = \max_{(x,y)\in\mathcal{C}_i}(A_i y + B_i x + c_i) \tag{7.16}$$

$$m_i = \min_{(x,y)\in\mathcal{C}_i}(A_i y + B_i x + c_i) \tag{7.17}$$

Note that M_i and m_i are finite and that they can be computed by solving a linear program since $\mathcal{C}_i$ is a convex polytope. Now, it is easy to verify that (7.14) is equivalent to

$$\begin{aligned} & t_i \leq M_i \delta_i, \\ & t_i \geq m_i \delta_i, \\ & t_i \leq A_i y + B_i x + c_i - m_i(1-\delta_i) \\ & t_i \geq A_i y + B_i x + c_i - M_i(1-\delta_i) \end{aligned} \tag{7.18}$$

Therefore, (7.10), (7.11), (7.15), and (7.18) represent $F(x,y)$ in a mixed-integer linear form.

We can also transform the constraint $G(x,y) \leq 0$ into a mixed-integer linear form as follows.

First, we write $G(x, y)$ in its max-min canonical form as follows [30]:

$$G_\ell(x, y) = \max_{i=1,\dots,m_\ell} \bigl(\min_{j=1,\dots,n_\ell} (\alpha_{ij\ell}^T y + \beta_{ij\ell}^T x + \xi_{ij\ell}) \bigr) \le 0 \quad \text{for } \ell = 1, \dots, q \tag{7.19}$$

where m_ℓ and n_ℓ are integers, $\alpha_{ij\ell} \in \mathbb{R}^{n_y}$, $\beta_{ij\ell} \in \mathbb{R}^{n_x}$, $\xi_{ij\ell} \in \mathbb{R}$, and q is the number of entries of G.

In the second step, we transform (7.19) into a mixed-integer linear form. To this end, we can proceed as follows [8]. In the sequel, we consider a single fixed ℓ for simplicity. Note that (7.19) is equivalent to

$$\min_{j=1,\dots,n_\ell} (\alpha_{ij\ell}^T y + \beta_{ij\ell}^T x + \xi_{ij\ell}) \le 0 \quad \text{for } \ i = 1, \dots, m_\ell. \tag{7.20}$$

Define the convex polytopes $\mathcal{D}_{ij\ell}$ for $i = 1, \dots, m_\ell$ and $j = 1, \dots, n_\ell$ as

$$\begin{aligned} \mathcal{D}_{ij\ell} = \{(x, y) : \alpha_{ij\ell}^T y + \beta_{ij\ell}^T x + \xi_{ij\ell} \le \alpha_{ij'\ell}^T y + \beta_{ij'\ell}^T x + \xi_{ij'\ell} \\ \text{for } \ j' = 1, \dots, j-1, j+1, \dots, n_\ell\}. \end{aligned}$$

Now as before, we define binary variables $\delta_{ij\ell} \in \{0, 1\}$ for $i = 1, \dots, m_\ell$ and $j = 1, \dots, n_\ell$ such that

$$[\delta_{ij\ell} = 1] \Leftrightarrow [(x, y) \in \operatorname{int}(\mathcal{D}_{ij\ell})] \tag{7.21}$$

and such that for a point (x, y) on the boundary of several regions, only one $\delta_{ij\ell}$ is taken equal to 1 and the others are set equal to 0. For each $i = 1, \dots, m_\ell$ and each $j = 1, \dots, n_\ell$, these relations are then equivalent to

$$\begin{aligned} (\alpha_{ij\ell}^T - \alpha_{ij'\ell}^T) y + (\beta_{ij\ell}^T - \beta_{ij'\ell}^T) x + \xi_{ij\ell} - \xi_{ij'\ell} \le M_{ij\ell}^* (1 - \delta_{ij\ell}) \\ \text{for } \ j' = 1, \dots, j-1, j+1, \dots, n_\ell \end{aligned} \tag{7.22}$$

$$\sum_{j=1}^{n_\ell} \delta_{ij\ell} = 1 \tag{7.23}$$

where

$$M_{ij\ell}^* \equiv \max_{\substack{j'=1,\dots,n_\ell \\ j' \ne j}} \max_{x \in \mathcal{B}_x, y \in \mathcal{B}_y} (\alpha_{ij\ell}^T - \alpha_{ij'\ell}^T) y + (\beta_{ij\ell}^T - \beta_{ij'\ell}^T) x + \xi_{ij\ell} - \xi_{ij'\ell}.$$

Note that $M_{ij\ell}^*$ is finite since x and y belong to a polytope. In fact, $M_{ij\ell}^*$ can be obtained using a finite number ($n_\ell - 1$) of linear programs. We can now write (7.20) as

$$\sum_{j=1}^{n_\ell} \delta_{ij\ell} (\alpha_{ij\ell}^T y + \beta_{ij\ell}^T x + \xi_{ij\ell}) \le 0 \quad \text{for } \ i = 1, \dots, m_\ell \tag{7.24}$$

which is non-linear due to the presence of the product between the binary variables $\delta_{ij\ell}$ and the real-valued variables x and y. However, it can be translated into the equivalent mixed-integer linear inequalities in a similar way as (7.13). Let

$$z_{ij\ell} = \delta_{ij\ell}(\alpha_{ij\ell}^T y + \beta_{ij\ell}^T x + \xi_{ij\ell}), \tag{7.25}$$

then (7.24) reduces to

$$\sum_{j=1}^{n_\ell} z_{ij\ell} \leq 0 \quad \text{for} \;\; i = 1, \ldots, m_\ell. \tag{7.26}$$

Although (7.25) is nonlinear, it can be transformed into linear equations as follows. Define for $i = 1, \ldots, m_\ell$,

$$\begin{aligned} M_{i\ell} &= \max_{j=1,\ldots,n_\ell} \Big(\max_{x \in \mathcal{B}_x, y \in \mathcal{B}_y} (\alpha_{ij\ell}^T y + \beta_{ij\ell}^T x + \xi_{ij\ell}) \Big) \\ m_{i\ell} &= \min_{j=1,\ldots,n_\ell} \Big(\min_{x \in \mathcal{B}_x, y \in \mathcal{B}_y} (\alpha_{ij\ell}^T y + \beta_{ij\ell}^T x + \xi_{ij\ell}) \Big). \end{aligned}$$

Note that $M_{i\ell}$ and $m_{i\ell}$ are finite and each of them can be computed by solving n_ℓ linear programs. Then, for $i = 1, \ldots, m_\ell$, (7.25) is equivalent to [8]

$$\begin{aligned} & z_{ij\ell} \leq M_{i\ell}\delta_{ij\ell}, \\ & z_{ij\ell} \geq m_{i\ell}\delta_{ij\ell}, \\ & z_{ij\ell} \leq \alpha_{ij\ell}^T y + \beta_{ij\ell}^T x + \xi_{ij\ell} - m_{i\ell}(1 - \delta_{ij\ell}) \\ & z_{ij\ell} \geq \alpha_{ij\ell}^T y + \beta_{ij\ell}^T x + \xi_{ij\ell} - M_{i\ell}(1 - \delta_{ij\ell}). \end{aligned} \tag{7.27}$$

Therefore, (7.22), (7.23), (7.26), and (7.27) represent (7.20), and accordingly (7.19), in a mixed-integer linear form.

Now, collect all variables y, t, z, δ in a vector V and define a vector c_{tot} such that $\sum_{i=1}^{s} t_i = c_{\text{tot}}^T V$. Accordingly, we can rewrite the inner optimization problem of (7.4), i.e.,

$$\Omega(x) = \begin{cases} \max\limits_{y \in \mathcal{B}_y} \; F(x,y) \\ \qquad\quad \text{s.t.} \quad G(x,y) \leq 0 \end{cases} \tag{7.28}$$

as follows:

$$\begin{aligned} \max_{V \in \mathbb{R}^{n_V}} \quad & c_{\text{tot}}^T V \\ \text{s.t.} \quad & A_{\text{tot}} V \leq b_{\text{tot}} + C_{\text{tot}} x \\ & V_i \in \{0,1\} \quad \text{for } i \in \mathcal{I} \\ & V_i \in \mathbb{R} \qquad\;\; \text{for } i \notin \mathcal{I} \end{aligned}$$

where n_V denotes the number of entries of V and for appropriately defined matrices $A_{\text{tot}}, C_{\text{tot}}$, vector b_{tot}, and set $\mathcal{I}$. Therefore, the inner optimization problem (7.28) has been transformed into a multi-parametric mixed-integer linear programming (mp-MILP) problem, which can be solved using the algorithm in [33]. The optimal value function, i.e., Ω, is then PWA; however, it may not be continuous [12].

Remark 7.2.1 Note that for a continuous PWA function, we can consider all regions $\mathcal{X}_i$ in which the function is affine to be defined by non-strict inequalities (as on the boundaries the function is continuous). On the other hand, for a discontinuous PWA function, we have to distinguish explicitly between strict and non-strict inequalities (as on the boundaries only one affine function can be active.) □

Therefore, to solve the outer optimization problem of (7.4), i.e., $\min\limits_{x \in \mathcal{B}_x} \Omega(x)$ we can apply two approaches. One way is to use the approach explained in [8] to transform the strict inequalities, which exist due to the discontinuities in the PWA representation of Ω, into non-strict ones[2], and then use a similar procedure as above to obtain an MILP problem and solve it using the available MILP solvers that are based on e.g. branch-and-bound methods or cutting plane methods [2, 72].

Another way for solving the outer optimization problem is to consider the PWA form of Ω as follows:

$$\Omega(x) = a_i^T x + b_i \quad \text{ if } \; x \in \chi_i$$

where $\chi_i = \{x : S_i x \leq p_i\}$ for $i = 1, \dots, n_\chi$ is a convex polytope. Then for each i, we solve the following linear programming (LP) problem:

$$\begin{aligned} \min_{x \in \chi_i} \; & a_i^T x + b_i \\ \text{s.t.} \quad & S_i x \leq p_i. \end{aligned}$$

using e.g. the simplex method or an interior point algorithm [90, 118]. In this way we obtain n_χ solutions, and we choose the one with the lowest value of Ω as the optimal solution of the outer problem.

7.2.2 Solution Approach for Case II

To solve the stochastic optimization problem (7.5), we can apply the approximation methods of Section 6.2.2. Since F and G are MMPS functions, we can write them as a difference of two convex functions and then approximate the expected value of the objective function F by its upper bound $U_F(x)$ and the expected value of the constraint function G by its upper bound $U_G(x)$. Hence, we obtain the following approximate optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & U_F(x) \\ \text{s.t.} \quad & U_G(x) \leq 0 \end{aligned} \tag{7.29}$$

[2] In this approach, a strict inequality of the form $x < a$ is replaced by $x \leq a - \varepsilon$ where ε is the machine precision.

Note that the constraint $U_G(x) \leq 0$ in (7.29) assures that the original constraint in (7.5) will be satisfied. In general, U_F and U_G are non-linear, non-convex functions. Hence, (7.29) is a non-linear optimization problem that can be solved using multi-start sequential quadratic programming (SQP) [80], genetic algorithms [46], simulated annealing [34], etc.

7.2.3 Solution Approach for Case III

For the last case, we rewrite the optimization problem (7.6), using the upper bounds for the expected value of the objective function and the constraint (cf. Section 6.2.2), as follows:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \max_{y\in\mathbb{R}^m} \quad & U_F(x,y) \\ \text{s.t.} \quad & U_G(x,y) \leq 0 \end{aligned}$$

which is a multi-parametric, non-linear (mp-NLP) optimization problem that can be solved using mp-NLP algorithms [32, 51–53, 94].

Remark 7.2.2 Since mp-NLP problems are highly complex and hard to solve, the third case will not be discussed anymore in the remainder of this chapter. □

7.3 Applications

In this section we present two types of problems in which we need to deal with the optimization problem of an MMPS function. One is a filtering problem and the other one is a reference tracking problem.

7.3.1 Filtering Problem

The first problem is a filtering problem for which we consider the setting of a two-player game. The aim of one player is to estimate the final state of the system using a series of measurements observed over time, while the other player tends to distort this estimation by disturbing the measurements of the first player. Assume, e.g., that player 1 has a truck and player 2 has an observing plane[3]. The first player is directing the truck during the discrete time span $\{1, \dots, \mathcal{T}\}$ and does not want his final location $x(\mathcal{T})$ to be detected by the plane. There exist also a control action w for moving the truck around, and a cloaking or jamming action v that perturbs the

[3] This is in fact an example of a pursuer-evader game. The various guises pursuit games may assume in warfare are ship and submarine, missile and bomber, or tank and jeep [59]. Other typical examples of a pursuer-evader game are the Homicidal Chauffeur game and the Lady in the Lake problem [70], or the Princess and Monster game [59].

measurements of the second player. Both moving the truck and cloaking/jamming costs energy for the first player and hence, he tends to minimize these actions. On the other hand, the second player wants to determine an estimate $\hat{x}_{\mathcal{T}}$ of $x(\mathcal{T})$ as accurately as possible using the obtained measurements of the truck's location at each time step k for $k = 1, \ldots, \mathcal{T}$.

Consider the following system variables: the state of the system is denoted by $x(k) \in \mathbb{R}^n$, the control variable by $w(k) \in \mathbb{R}^n$, the measurement process by $y(k) \in \mathbb{R}^l$, the measurement disturbance by $v(k) \in \mathbb{R}^l$, and the disturbance-free measurement model by $g(\cdot)$. Filtering begins at time step $k = 0$ and we assume that for the second player the estimated initial state $\hat{x}_0$ and the measurement time-history $\{y(k)\}_{k=1}^{\mathcal{T}}$ are known and the initial state $x(0)$, the final state $x(\mathcal{T})$, and $w(\cdot)$ are unknown. The estimated final state $\hat{x}_{\mathcal{T}}$ is the decision variable for the second player.

Here we consider only additive noise (in order to stay within the MMPS framework), and also we consider two types of systems: Type I and Type II. A system of Type I is defined as

$$x(k) = f(x(k-1)) + w(k) \tag{7.30}$$

$$y(k) = g(x(k)) + v(k) \tag{7.31}$$

where w and v are control variables and f and g are MMPS functions in x. A system of Type II is defined as

$$x(k) = f(x(k-1)) + \delta f(x(k-1)) + w(k) \tag{7.32}$$

$$y(k) = g(x(k)) + \delta g(x(k)) + v(k) \tag{7.33}$$

where again f and g are MMPS functions, w and v are noise signals, and δf and δg are bounded. This system can be considered for example when $f + \delta f$ is an approximation of a nonlinear function F with f an MMPS function and $\delta f = F - f$ nonlinear but small or bounded, i.e., $\|\delta f(x)\|_\infty \leq \varepsilon_1$ for some $\varepsilon_1 > 0$ and $\forall x \in \mathbb{R}^n$. The same explanation holds for δg with $\|\delta g(x)\|_\infty \leq \varepsilon_2$ for some $\varepsilon_2 > 0$ and $\forall x \in \mathbb{R}^n$. We first proceed with the systems of Type I, and later we can include Type II.

Remark 7.3.1 By treating δf and δg as bounded disturbances and including them in w and v, one can recast a system of Type II as a system of Type I. Hence, if both w and v are bounded as well, i.e., we are in Case I, then the recast problem is also in the form of Case I. However, if w and v are stochastic variables, i.e., we are in Case II, then the recast problem is in the form of Case III. As a result, we will not discuss systems of Type II separately in the remainder of this chapter. □

Remark 7.3.2 We use the ∞-norm in this section. However, similar results hold if the 1-norm (or a mix of the 1-norm and the ∞-norm) is considered. □

Now we are going to study Case I and Case II for this problem.

Uncertain System with a Bounded Noise

In the first step, we consider Case I of Section 7.1, where $w(k)$ and $v(k)$ are assumed to be included in convex polytopes $\mathcal{B}_w = \{w \in \mathbb{R}^n : A_w w \leq b_w\}$ and $\mathcal{B}_v = \{v \in \mathbb{R}^m : A_v v \leq b_v\}$, respectively. We also assume that the initial state $x(0)$ belongs to the ball $\mathcal{B}_\infty(\hat{x}_0, C)$, i.e., $\|x(0) - \hat{x}_0\|_\infty \leq C$ for a given $C < \infty$ where $\hat{x}_0$ is the estimate of the initial state and it is assumed to be known. Consider the following equation:

$$\begin{aligned} x(k) &= f(x(k-1)) + w(k) \\ &= f(f(x(k-2)) + w(k-1)) + w(k) \\ &= f(f(f(x(k-3)) + w(k-2)) + w(k-1)) + w(k) \\ &\vdots \\ &= f(f(f(\dots(f(x(0)) + w(1)) + w(2)) + \dots) + w(k-1)) + w(k) \\ &= h_k(x(0), w(1), \dots, w(k)) \end{aligned}$$

where h_k is an MMPS function of $x(0), w(1), \dots, w(k)$. Hence, we can write the system (7.30) for the final state $x(\mathcal{T})$ as

$$\begin{aligned} x(\mathcal{T}) &= h(x(0), w(1), \dots, w(\mathcal{T})) \\ &= h(x(0), \tilde{w}) \end{aligned} \tag{7.34}$$

where $h_\mathcal{T}$ is denoted by h and $\tilde{w} = [w^T(1), \dots, w^T(\mathcal{T})]^T$. Since each $w(k)$ belongs to the convex polytope $\mathcal{B}_w$, $\tilde{w}$ is also in a convex polytope $\mathcal{B}_{\tilde{w}} = (\mathcal{B}_w)^\mathcal{T}$.

Consider the following definition for the norms of matrices:

Definition 7.3.3 ([47]) *For a matrix $A \in \mathbb{R}^{m\times n}$, the matrix 1-norm and ∞-norm are defined as*

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}| \ , \qquad \|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^{n} |a_{ij}|$$

respectively.

Moreover, the following inequalities hold [47]:

$$\|Ax\|_1 \leq \|A\|_1 \ \|x\|_1, \qquad \|Ax\|_\infty \leq \|A\|_\infty \ \|x\|_\infty.$$

for any $x \in \mathbb{R}^n$ where the $\|x\|_1 = \sum_{i=1}^n |x_i|$ and $\|x\|_\infty = \max_{i=1,\dots,n} |x_i|$ (cf. Definition 4.2.1).

Recall from Section 2.4 that MMPS functions are equivalent to continuous PWA functions. It is been shown in [42] that a continuous piecewise affine function f

with the corresponding Jacobian matrices $J^{(\cdot)}$, defined in Definition 2.4.4, satisfies a Lipschitz condition, i.e., the following inequality holds for every x and y in $\mathbb{R}^n$:

$$\|f(x) - f(y)\|_\infty \leq L\|x - y\|_\infty \tag{7.35}$$

where L is the maximum of the matrix ∞-norms of $J^{(1)}, \ldots, J^{(r)}$ and it is the smallest value for which (7.35) holds. Hence, it is the Lipschitz constant of f.

Consequently, the MMPS functions f and g, which are equivalent to continuous PWA functions, satisfy a Lipschitz condition.

Lemma 7.3.4 *If f is Lipschitz, then there exists a scalar $M < \infty$ such that for any $x \in \mathrm{dom}(f)$, we have $\|f(x)\|_\infty \leq M(\|x\|_\infty + 1)$.*

Proof : Let L be the Lipschitz constant of f. Hence, by considering the triangle inequality and the Lipschitz condition, we have for any $x, y \in \mathrm{dom}(f)$:

$$\begin{aligned}\|f(x)\|_\infty &= \|f(x) - f(y) + f(y)\|_\infty \\ &\leq \|f(x) - f(y)\|_\infty + \|f(y)\|_\infty \\ &\leq L\|x - y\|_\infty + \|f(y)\|_\infty\end{aligned}$$

Using the triangle inequality once more, we obtain:

$$\|f(x)\|_\infty \leq L\|x\|_\infty + L\|y\|_\infty + \|f(y)\|_\infty$$

Consider a point $y_0 \in \mathrm{dom}(f)$ and define $Q = L\|y_0\|_\infty + \|f(y_0)\|_\infty$. Let $M = \max(L, Q)$, which is clearly finite. Then, we have

$$\|f(x)\|_\infty \leq M(\|x\|_\infty + 1)$$

and the proof is complete. □

Now, we are going to determine an upper bound for $\|\hat{x}_T\|_\infty$. Let

$$U_w = \max_{w \in \mathcal{B}_w} \|w\|_\infty,$$

which is finite since $\mathcal{B}_w$ is bounded[4]. Then, we have:

$$\begin{aligned}\|x(k)\|_\infty &= \|f(x(k-1)) + w(k)\|_\infty \\ &\leq \|f(x(k-1))\|_\infty + \|w(k)\|_\infty\end{aligned}$$

[4] Note that $\max\limits_{w \in \mathcal{B}_w} \|w\|_\infty$ does not correspond to a convex problem. However, it can be solved as follows: first for $i = 1, \ldots, n_w$ with n_w the number of elements of w, we solve the linear optimization problem: $\max\limits_{w_i \in \mathcal{B}_w} \ w_i$ and $\max\limits_{w_i \in \mathcal{B}_w} \ -w_i$. Hence, we need to solve $2n_w$ linear programming problems and then choosing the largest objective function value yields U_w.

$$
\begin{aligned}
&\overset{(*)}{\leq} M(\|x(k-1)\|_\infty + 1) + U_w \\
&\leq M(\|f(x(k-2)) + w(k-1)\|_\infty + 1) + U_w \\
&\leq M(\|f(x(k-2))\|_\infty + \|w(k-1)\|_\infty) + M + U_w \\
&\overset{(**)}{\leq} M^2(\|x(k-2)\|_\infty + 1) + U_w(M+1) + M \\
&\leq M^2(\|f(x(k-3)) + w(k-2)\|_\infty + 1) + U_w(M+1) + M \\
&\leq M^3(\|x(k-3)\|_\infty + 1) + U_w(M^2+M+1) + M^2 + M \\
&\vdots \\
&\leq M^k(\|x(0)\|_\infty + 1) + U_w \sum_{i=0}^{k-1} M^i + \sum_{j=1}^{k-1} M^j
\end{aligned}
$$

where both $(*)$ and $(**)$ are obtained using Lemma 7.3.4. Hence, an upper bound for $\|x(\mathcal{T})\|_\infty$ is given by

$$
\|x(\mathcal{T})\|_\infty \leq M^{\mathcal{T}} \ \|x(0)\|_\infty + L_{\mathcal{T}} \tag{7.36}
$$

where

$$
L_{\mathcal{T}} = \begin{cases} M^{\mathcal{T}} + U_w \dfrac{M^{\mathcal{T}} - 1}{M-1} + \dfrac{M(M^{\mathcal{T}-1} - 1)}{M-1} & \text{if } \ M \neq 1 \\ \mathcal{T} \ (U_w + 1) & \text{if } \ M = 1 \end{cases}
$$

Moreover, based on our assumption, $\|x(0) - \hat{x}_0\|_\infty \leq C$ for some $C < \infty$. Hence, by using the triangle inequality, we have

$$
\begin{aligned}
\|x(0)\|_\infty &= \|x(0) - \hat{x}_0 + \hat{x}_0\|_\infty \\
&\leq \|x(0) - \hat{x}_0\|_\infty + \|\hat{x}_0\|_\infty \\
&\leq C + \|\hat{x}_0\|_\infty
\end{aligned}
$$

Since we may assume without loss of generality that $\hat{x}_{\mathcal{T}}$ should not be outside the maximal range set of $x(\mathcal{T})$ given by (7.36), we obtain the following upper bound for $\|\hat{x}_{\mathcal{T}}\|_\infty$:

$$
\|\hat{x}_{\mathcal{T}}\|_\infty \leq M^{\mathcal{T}} C + M^{\mathcal{T}} \|\hat{x}_0\|_\infty + L_{\mathcal{T}}.
$$

In other words, $\hat{x}_{\mathcal{T}}$ belongs to the convex polytope $\mathcal{B}_{\hat{x}_{\mathcal{T}}} = \{\hat{x}_{\mathcal{T}} : \|\hat{x}_{\mathcal{T}}\|_\infty \leq M^{\mathcal{T}} C + M^{\mathcal{T}} \|\hat{x}_0\|_\infty + L_{\mathcal{T}}\}$.

As mentioned before, the first player tends to minimize the control and cloaking/jamming energy and to maximize the difference between the final state $x(\mathcal{T})$ and the second player's estimation of the final state $\hat{x}_{\mathcal{T}}$, while the second player tends to minimize this difference and has to take the worst-case control and cloaking/jamming action by the first player into account as well as the worst case uncertainty about the initial state $x(0)$. Hence, the overall objective function, which will be minimized by player 2 and maximized by player 1, is defined as follows:

$$J(\hat{x}_{\mathcal{T}}, x(\mathcal{T}), \tilde{w}, \tilde{v}) = -\gamma \sum_{k=1}^{\mathcal{T}} \big[\|w(k)\|_\infty + \|v(k)\|_\infty\big] + \|x(\mathcal{T}) - \hat{x}_{\mathcal{T}}\|_\infty$$

for some $\gamma > 0$, or equivalently by eliminating $\tilde{v}$, as

$$J(\hat{x}_{\mathcal{T}}, x(0), \tilde{w}) = -\gamma \sum_{k=1}^{\mathcal{T}} \big[\|w(k)\|_\infty + \|y(k) - g(x(k))\|_\infty\big] + \|h(x(0), \tilde{w}) - \hat{x}_{\mathcal{T}}\|_\infty \tag{7.37}$$

Hence, we obtain the following optimization problem:

$$\min_{\hat{x}_{\mathcal{T}} \in \mathcal{B}_{\hat{x}_{\mathcal{T}}}} W(\hat{x}_{\mathcal{T}}) \tag{7.38}$$

where

$$\begin{aligned} W(\hat{x}_{\mathcal{T}}) = \max_{x(0) \in \mathbb{B}_\infty(\hat{x}_0, C)} \max_{\tilde{w} \in \mathcal{B}_{\tilde{w}}} & J(\hat{x}_{\mathcal{T}}, x(0), \tilde{w}) \\ \text{s.t.} \quad & y(k) - g(x(k)) \in \mathcal{B}_v \;\; k = 1, \ldots, \mathcal{T}. \end{aligned} \tag{7.39}$$

Remark 7.3.5 Since $v \in \mathcal{B}_v$, we have the additional constraint $y(k) - g(x(k)) \in \mathcal{B}_v$ for a given realization of the control action vector $\tilde{w}$, a given measurement time-history $\{y(k)\}_{k=1}^{\mathcal{T}}$, and a given terminal state $x(\mathcal{T})$. Note that this constraints cannot cause infeasibility. Indeed, for a given realization of $\tilde{w}$, a given $\{y(k)\}_{k=1}^{\mathcal{T}}$, and a given $x(\mathcal{T})$, we can always find an $x(0)$ that satisfies the constraint. □

Note that since all variables in (7.39) are bounded, $W(\hat{x}_{\mathcal{T}})$ is bounded as well, and since $\hat{x}_{\mathcal{T}}$ is bounded, the solution of (7.38) exists and is finite for any $\gamma > 0$.

The worst-case optimization problem (7.38)-(7.39) is a problem of the form (7.4) and therefore, it can be solved as indicated in Section 7.2.1.

Stochastic System

After discussing the filtering problem for the first case where w and v were assumed to be bounded, we consider now a system of Type I in which w, v, and $x(0)$ are assumed to be stochastic variables with a given probability density function. In the framework of the truck-plane example, we can now assume that the plane (the second player) knows typical probability density functions of w, v, and $x(0)$ based on previous experience or a priori knowledge. As mentioned before, we only consider distributions with finite moments for which a closed-form expression exists. Having this assumption, we redefine the optimization problem (7.38) as follows:

$$\min_{\hat{x}_{\mathcal{T}} \in \mathbb{R}^n} \mathbb{E}_{x(0),\tilde{w}}[J(\hat{x}_{\mathcal{T}}, x(0), \tilde{w})]. \tag{7.40}$$

where $J(\hat{x}_{\mathcal{T}}, x(0), \tilde{w})$ is an MMPS function of its arguments as defined in (7.37). This is a problem of the form (7.5), and hence, as explained in Section 7.2.2, it can be solved using the approximation method of Section 6.2.2.

7.3.2 Reference Tracking Problem

The second problem that we consider here is a reference tracking problem. In this problem the reference signal is given and the aim is to minimize the difference between the output of the system and the reference signal. The system is defined as follows:

$$x(k) = f(x(k-1), u(k)) + w(k) \tag{7.41}$$

$$y(k) = g(x(k)) + v(k) \tag{7.42}$$

where $x(k)$ is the state of the system, $y(k)$ is the output, and $u(k)$ is the input of the system at time or event step $k \in \{1, \ldots, \mathcal{T}\}$. We also consider external noise vectors $w(k)$ and $v(k)$ that perturb the system. We assume that the initial state of the system $x(0)$ is known. Similar to Section 7.3.1, f and g are MMPS functions of their arguments. The objective function is defined as follows:

$$J(\tilde{u}, \tilde{w}, \tilde{v}) = \sum_{k=1}^{\mathcal{T}} \lambda_k \|y(k) - r(k)\|_\infty \tag{7.43}$$

where $r(k)$ is the reference signal and λ_k is a weighting factor at time or event step k, and $\tilde{u} = [u^T(1), \ldots, u^T(\mathcal{T})]^T$ ($\tilde{w}$ and $\tilde{v}$ are defined similarly). Assume that the constraints on $\tilde{u}, \tilde{w}$, and/or $\tilde{v}$ are gathered in an expression of the form

$$G(\tilde{u}, \tilde{w}, \tilde{v}) \le 0$$

where $G(\cdot)$ is an MMPS function of its arguments.

We can rewrite the objective function (7.43) by substituting (7.41) in (7.42) for each event step k. To this end, we compute $x(k)$ as follows:

$$\begin{aligned} x(k) &= f(x(k-1), u(k)) + w(k) \\ &= f(f(x(k-2), u(k-1)) + w(k-1), u(k)) + w(k) \\ &= f(f(f(x(k-3), u(k-2)) + w(k-2), u(k-1)) + w(k-1), u(k)) + w(k) \end{aligned}$$

$$\begin{aligned}&\vdots\\&= f(f(f(\dots(f(x(0),u(1))+w(1),u(2))+\dots))+w(k-1),u(k))+w(k)\\&= h_k(u(1),\dots,u(k),w(1),\dots,w(k))\end{aligned}$$

where $h_k(\cdot)$ is an MMPS function of its arguments. Accordingly, we can rewrite $y(k)$ as follows:

$$\begin{aligned}y(k) &= g(x(k)) + v(k)\\&= g(h_k(u(1),\dots,u(k),w(1),\dots,w(k))) + v(k)\\&= H_k(u(1),\dots,u(k),w(1),\dots,w(k),v(k))\end{aligned}\tag{7.44}$$

where $H_k(\cdot)$ is also an MMPS function of its arguments. Hence, (7.43) can be written in the following form:

$$\begin{aligned}J(\tilde{u},\tilde{w},\tilde{v}) &= \sum_{k=1}^{T}\lambda_k\|H_k(u(1),\dots,u(k),w(1),\dots,w(k),v(k))) - r(k)\|_\infty\\&= \mathcal{H}(\tilde{u},\tilde{w},\tilde{v})\end{aligned}\tag{7.45}$$

with $\mathcal{H}(\cdot)$ being an MMPS function of its arguments. This implies that we intend to track the reference signal by minimizing over the input $\tilde{u}$ while environment plays against us by maximizing over the noise signals $\tilde{w}$ and $\tilde{v}$.

Now, we are going to study Case I and Case II for this problem.

Uncertain System with a Bounded Noise

In this case (cf. Section 7.1.1) we assume that $w(k), v(k)$, and $u(k)$ belong to convex polytopes $\mathcal{B}_w = \{w \in \mathbb{R}^n : A_w w \leq b_w\}, \mathcal{B}_v = \{v \in \mathbb{R}^m : A_v v \leq b_v\}$, and $\mathcal{B}_u = \{u \in \mathbb{R}^n : A_u u \leq b_u\}$, respectively. Hence, the optimization problem can be defined in the following form:

$$\begin{aligned}\min_{\tilde{u}\in\mathcal{B}_{\tilde{u}}}\ \max_{\tilde{w}\in\mathcal{B}_{\tilde{w}},\tilde{v}\in\mathcal{B}_{\tilde{v}}}\quad &\mathcal{H}(\tilde{u},\tilde{w},\tilde{v})\\ \text{s.t.}\quad &G(\tilde{u},\tilde{w},\tilde{v}) \leq 0\end{aligned}\tag{7.46}$$

where $\mathcal{B}_{\tilde{u}} = (\mathcal{B}_u)^{\mathcal{T}}, \mathcal{B}_{\tilde{w}} = (\mathcal{B}_w)^{\mathcal{T}}$, and $\mathcal{B}_{\tilde{v}} = (\mathcal{B}_v)^{\mathcal{T}}$ are convex polytopes, $\mathcal{H}(\tilde{u},\tilde{w},\tilde{v})$ is defined in (7.45), and $G(\cdot)$ is an MMPS function of its arguments. Note that this optimization problem has a finite solution due to the fact that all the optimization variables belong to a polytope and hence, are bounded. Consequently, we obtain a problem of form (7.4) which can be solved using an mp-MILP solver (cf. Section 7.2.1).

Stochastic System

In this case we assume that both $w(k)$ and $v(k)$ are stochastic variables with a given probability distribution with finite moments for which a closed-form expression exists. Hence, the optimization problem is defined as follows:

$$\begin{aligned} \min_{\tilde{u}\in\mathbb{R}^n} \quad & \mathbb{E}_{\tilde{w},\tilde{v}}[\mathcal{H}(\tilde{u},\tilde{w},\tilde{v})] \\ \text{s.t.} \quad & \mathbb{E}_{\tilde{w},\tilde{v}}G(\tilde{u},\tilde{w},\tilde{v}) \leq 0 \end{aligned} \tag{7.47}$$

This is a problem of the form (7.5), and hence, as explained in Section 7.2.2, it can be solved using the approximation method of Section 6.2.2, and then applying non-linear optimization algorithms such as SQP, genetic algorithms, or simulated annealing.

7.4 Summary

In this chapter, we have studied the min-max optimization of MMPS systems. Three cases were considered: uncertain min-max optimization problems, stochastic minimization problems and stochastic min-max optimization problems. For each case, we have proposed solution approaches, namely the mixed-integer linear programming for the first case and the approximation method of Section 6.2.2 for the second and the third case.

Afterwards, we have presented two sample problems that can be modeled using MMPS systems and that result in a min-max or stochastic optimization of such systems. The first problem is the filtering problem and we have considered a two-player game setting in which one player tries to estimate the final state of the system using a series of measurements observed over time, while the other player is disturbing the measurements of the first player. The objective function of this problem has been defined as an MMPS function and hence, we have obtained a min-max and a stochastic optimization problem using this objective function. To solve these problems, we considered the above-mentioned solution approaches for each case.

The second problem is a reference tracking problem. The idea of this problem is to track a given reference signal by minimizing the difference between the system's output and the reference signal. The objective function here has been also defined as an MMPS function and we have considered again the above cases to solve the min-max and stochastic optimization problem of this system.

Chapter 8

Conclusions and Recommendations

This chapter concludes the thesis by summarizing the main contributions and also presenting some interesting topics for future research.

8.1 Conclusions

The focus of this PhD thesis was on improving the computational aspects of MPC and identification of stochastic MPL systems as a special class of discrete-event systems. Since models of such systems are linear in max-plus algebra, control and identification of these systems has attracted many attention and many studies have been done on these topics. One of the control methods that has been applied to (stochastic) MPL systems, is MPC. In both MPC and identification framework, the objective function is defined as an expected value of a stochastic max-plus-scaling function, which is quite complex and time-consuming to compute using available methods such as numerical integration, analytic integration, etc.

Hence, our aim was to find an efficient method to compute this expected value. To that end, we have proposed an approximation method based on moments of random variables. This method was inspired by the relation between the p-norm and the ∞-norm of vectors. This approximation approach results in an analytic expression if the distribution of random variables has finite moments for which a closed-form expression exists, such as the uniform distribution, normal distribution, Beta distribution, etc. We have also assumed that the elements of the stochastic vector are independent random variables. This is a key assumption in our approach since otherwise we would have to deal with the joint expected value, for which this method is not applicable anymore.

Furthermore, we have shown that the approximation function obtained using this approximation approach is convex in its arguments and that the error of this approximation method is bounded both from below and from above. This assures us that this approximation function, which is indeed an upper bound for the expected value of stochastic max-plus-scaling functions, does not grow unboundedly.

We have then applied this approximation method to different classes of discrete-event systems, namely stochastic MPL systems, stochastic switching MPL systems, and stochastic MMPS systems, and the results are summarized below.

MPC for stochastic MPL systems

As mentioned above, the stochastic MPL-MPC optimization problem is computationally complex due to the presence of the expected value of stochastic max-plus-scaling functions in the objective function. In order to simplify the problem, we have approximated this objective function by its upper bound using the approximation method.

Since the approximation function is convex in its arguments, by considering the constraints such that they are nondecreasing affine functions of the system output, we have obtained a convex optimization problem, which can be solved efficiently.

Moreover, by choosing the appropriate order of moments we can decrease the approximation error to some extent. In this thesis, this choice was made by means of numerical experiments and the approximation results for different order of moments has been illustrated in a worked example. In this example, the performance of the approximation method has been compared to the one using numerical or analytic integration, Monte Carlo simulation, and nominal MPC. It was concluded that the results obtained using the approximation method are comparable to the other ones.

Identification of stochastic MPL systems

We have shown that the identification of stochastic MPL systems can also be simplified computationally using the approximation method. Accordingly, we have solved the approximate identification problem in which the objective function was an upper bound of the original function, and since we have obtained an explicit expression for the gradients, the parameter estimation can be done using gradient-based optimization methods despite having a non-convex objective function.

In two examples, we have compared the identification results using numerical integration, analytic integration, Monte Carlo simulation, and the approximation method for two types of random variables, namely the uniformly distributed random variables and the normally distributed random variables. We aimed to identify the unknown system parameters and the noise amplitude. In both examples, the identified parameters using the approximation method were very close to the ones using numerical or analytic integration or Monte Carlo simulation. Furthermore, the computation time of the optimization procedure using the approximation method was much less than the one using the other methods. Note that the estimated noise amplitude using any of the mentioned methods were quite far away from the exact values, since in general, in prediction error identification, one can obtain the correct system model, but it is much more difficult to estimate the noise model [48, 73].

MPC for stochastic switching MPL systems

We have applied the approximation method to another class of discrete-event systems, namely stochastic switching MPL systems in which the mode of the system can switch from one to another. We have considered both stochastic switching and stochastic system parameters at the same time. Since the latter random variable is a continuous one and the mode switching uncertainty is a discrete random variable, to compute the expected value we need to apply a combination of integration over the continuous stochastic variable and summation over the discrete stochastic variable. To that end, we have studied two cases: the case in which the random variables of mode switching and the systems parameters are independent and the case in which they are dependent.

For the case that the random variables are independent, we have applied the combination of both the approximation method in order to approximate the expected value of the parametric uncertainty and a scenario-based algorithm in order to reduce the computational complexity of the computation of the expected value of the mode switching uncertainty. Both methods simplify the problem and decrease the computation time while still yielding a performance that is comparable to the one using numerical or analytic integration.

In the second case with dependent random variables, the computation is more complex. We have shown that if the mode switching probability and the joint probability density function could be modeled or approximated by multi-variable piecewise polynomial functions, possibly multiplied by an exponential function, that are defined on polyhedral regions, then we can obtain an analytic expression for the expected value of the max-plus-scaling function appearing in objective function.

MPC for stochastic MMPS systems

Further, we have extended MPC to stochastic MMPS systems and as before, due to the presence of the expected value in the objective function, the optimization problem is computationally complex. To reduce this complexity, we have first proposed to write the MMPS function as a difference of two convex functions and then to compute its upper bound by finding an upper bound for the first convex function using the approximation method and finding a lower bound for the second convex function, which is subtracted. Both of these upper and lower bounds are convex in their arguments.

This procedure resulted in a sequence of convex problems and in the case of having convex constraints, we have obtained a sequence of convex optimization problems, while the original problem was not convex due to the MMPS objective function.

Min-max and stochastic optimization for MMPS systems

The last problem that we have considered was a general min-max optimization problem with an MMPS objective function. We studied three cases: min-max optimization of an MMPS function with bounded variables, minimization of an MMPS function with stochastic variables, and min-max optimization of an MMPS function with stochastic variables.

In the first case, we have shown that the optimization problem can be transformed into a mixed-integer linear programming (MILP) optimization problem and then can be solved using the available algorithms [2, 72]. In the second case, the MMPS objective function was approximated by the approximation method applied to stochastic MMPS systems and then the optimization problem has been solved using non-linear optimization algorithms. In the last case also, the objective function has been transformed into an approximate one using the method applied to stochastic MMPS systems and hence, the optimization problem turned into a multi-parametric, non-linear optimization problem, which can be solved using the available algorithms [32, 51–53, 94].

At the end, we considered two specific instances of the above general problems, namely the filtering problem and the reference tracking problem. In the first problem, we have a two-player game structure in which the aim is to estimate the final position of the system by one player while the measurements are perturbed by the second player. In the second problem, the aim is to track the reference signal by the output and to minimize the difference between these two.

8.2 Recommendations for Future Research

In this section, we first discuss some directions to improve our approximation method. Afterwards, there will be some other interesting topics that are related and can be considered for the future research.

Approximation method

It is very useful to find an efficient, and preferably analytic, way to determine an appropriate order for the moments used in the approximation method in order to obtain a better approximation and to decrease the approximation error. Note that in this thesis, this choice has been made by means of numerical experiments. One possible approach is to analyze how the upper bounds, i.e., (4.6) or (4.8), evolve as p evolves.

Other interesting issue is to find narrower upper bounds for the error of the approximation method that also depends on the type of the distribution of the random variables. Note that in this case, we have more information since we can use the properties of the distribution in specifying the approximation error. As such, a narrow bound may be obtained. This can be done by finding an alternative upper bound for Jensen's inequality that can be applied to random variables with both bounded and unbounded domain, or use the properties of each specific objective function to specify the error of the approximation method.

MPC and identification of stochastic MPL systems

In the worked examples presented in Section 5.1, we have compared the approximation method with other methods such as numerical and analytic integration, Monte Carlo simulation, and nominal MPC to solve the MPL-MPC optimization problem. It will be useful to perform a more extensive assessment and comparison of the approximation method with other computation methods such as the method based on variability expansion [110]. Moreover, it is interesting to compare the result of the MPC using these methods with the one obtained from the real solution. In this case, the ”real” solution can be obtained in two ways. One approach is to either compute it analytically (such as in the case of the uniform distribution) or to approximate it arbitrarily close by performing an extensive Monte Carlo simulation with a very large number of samples. The other approach is to consider the noise realization to be known in advance and to determine the real (exact) MPL-MPC solution using this noise realization[1].

Monte Carlo simulation can be made very precise by increasing the sample size. However, the computation time then also increases drastically (as explained in the examples of Section 5.1). Therefore, the approximation method proposed in this thesis – despite providing only an approximate and not an exact solution – is still preferable to the Monte Carlo simulation. In the examples of Section 5.1, we have not made use of the max-plus structure when performing Monte Carlo simulations. As shown in [13], the simulation of max-plus systems can be speeded up taking the max-plus structure into account. The method of [13] is called the *perfect sampling* and it continues the simulation until its output has the distribution of the stationary profile of the max-plus system. This approach has also been applied in [50], which proposes an algorithm that involves the computation of the expected value of specific max-plus expressions. Since the *perfect sampling* method results

[1] This is in fact a deterministic optimization problem since the noise realization is known.

in a fast and efficient computation, it is interesting to apply this method in order to compare its computational efficiency with the one of the proposed approximation method.

Recall that the identification problem of stochastic MPL systems resulted in a non-convex optimization problem that could be solved using multi-start gradient-based local optimization methods due to the presence of the gradients of the objective function. Obviously, the result of such non-convex optimization problems depends on the problem setting, such as the setting of the system, the control signals, or the noise structure. In this thesis, we have not studied the effect of using different settings and different optimization methods on the parameter estimation. Hence, it is still an open topic for further research to perform multiple experiments with different systems and/or different signals and noise besides applying different optimization methods and then, compare their performance.

We have also seen in the worked examples related to identification of stochastic MPL systems (cf. Section 5.2) that the estimated noise amplitude is quite far away from the exact one. Although it is mentioned in the literature that in general it is hard to obtain a good estimation for the noise amplitude, it is still desired to explore the possibilities of improving the estimation of the noise amplitude in the identification problem. One possible research direction is to investigate the input design process for stochastic MPL systems, or first consider a simpler problem such as the estimation of the noise variance using Kalman Filter in order to obtain an estimate of the possible outcome.

MPC for stochastic switching MPL systems

Yet another topic for future research is to explore other methods to compute the expected value of a max-plus-scaling function in a stochastic switching MPL system in the case that both the mode switching uncertainty and the parametric uncertainty are dependent on each other. The method proposed in this thesis is still quite complex and time-consuming. One way would be to use properties of the conditional expected value/probability and also the expected value of multivariate functions to find another approximation method that is more efficient than the one proposed in this thesis.

Other research topics

Stochastic MPL systems are not intrinsically stable. Hence, one interesting topic would be to study the stability of stochastic MPL systems in general and the stability of closed-loop MPC of stochastic MPL systems when the approximation method is used in the MPC optimization problem.

Another important issue is the timing issue of the stochastic MPL systems. This issue plays an important role in real applications such as scheduling a train network, or controlling a printer performance. Note that in the approximation method, we do not take the timing into account (cf. Section 3.1.4) since we would then deal with the updated distributions with unknown properties and as a result, we would then not be able to apply the approximation method anymore. Therefore, it is very interesting to find alternative approximation methods that will deal with the timing issue as well. One approach could consist in approximating the updated distribution using another distribution for which the current approximation method can be applied, such as the normal distribution or piecewise uniform distribution.

Development of identification algorithms for stochastic MPL systems based on input-output data (instead of input-state data) or with only partial state information is also another interesting topic for future research.

Appendix A

Order of Complexity of the Approximation Method Based on Variability Expansion

In Section 3.1.3, we have presented the order of complexity of the approximation method, which is based on the method of variability expansion [110]. The order of complexity of this method is

$$O\Big(\binom{n_{\tilde{e}}}{M}\, n_v \mathcal{V}(2M + n_v - 1, M)\Big) \tag{A.1}$$

where $\mathcal{V}(\cdot,\cdot)$ is defined in (3.13). In this appendix, we show that (A.1) is equivalent to

$$O\Big(\frac{n_{\tilde{e}}^M n_v^{\frac{M}{2}+1}}{M!(\frac{M}{2})!}\Big). \tag{A.2}$$

Here, we assume $n_v \gg n_{\tilde{e}} \gg M \geq 0$. Note that $n_v \gg n_{\tilde{e}}$ was also assumed in the complexity analysis of the method based on analytic integration in Section 3.1.3; moreover, M will always be less than or equal to $n_{\tilde{e}}$ [56]. Here, we assume that $n_{\tilde{e}} \gg M \geq 0$ since for large M, the variability expansion method is computationally very expensive. The first factor in (A.1) can be rewritten as follows:

$$\begin{aligned}\binom{n_{\tilde{e}}}{M} &= \frac{n_{\tilde{e}}!}{M!(n_{\tilde{e}} - M)!} = \frac{\overbrace{n_{\tilde{e}}(n_{\tilde{e}} - 1)\ldots(n_{\tilde{e}} - M + 1)}^{M \text{ terms}}}{M!} \\ &\approx O\Big(\frac{n_{\tilde{e}}^M}{M!}\Big) \qquad\qquad \text{if} \quad n_{\tilde{e}} \gg M \geq 0 \end{aligned} \tag{A.3}$$

The last factor in (A.1) can be rewritten in the following way (see (3.13)):

$$\mathcal{V}(2M+n_v-1,M)=\binom{2M+n_v-1-\lfloor\frac{M+1}{2}\rfloor}{M+n_v-1}+\binom{2M+n_v-1-\lfloor\frac{M+2}{2}\rfloor}{M+n_v-1}. \tag{A.4}$$

If $M = 2q$ where q is a positive integer, then $\lfloor\dfrac{2q+1}{2}\rfloor = q$ and if $M = 2q+1$ then $\lfloor\dfrac{2q+2}{2}\rfloor = q+1$. In the same way $\lfloor\dfrac{2q+2}{2}\rfloor = \lfloor\dfrac{2q+3}{2}\rfloor = q+1$. Hence, $\lfloor\dfrac{M+1}{2}\rfloor \approx \lfloor\dfrac{M+2}{2}\rfloor \approx \dfrac{M+1}{2}$. By replacing this in (A.4), we obtain:

$$\begin{aligned}
\mathcal{V}(2M+n_v-1,M) &= 2\binom{\frac{3}{2}M+n_v-\frac{3}{2}}{M+n_v-1} \\
&= 2\frac{(\frac{3}{2}M+n_v-\frac{3}{2})!}{(M+n_v-1)!(\frac{M-1}{2})!} \\
&= \frac{2(\frac{3}{2}M+n_v-\frac{3}{2})\dots(\frac{3}{2}M+n_v-\frac{3}{2}-\frac{M}{2}-\frac{1}{2}+1)}{(\frac{M-1}{2})!} \\
&\approx \frac{(\frac{3}{2}M+n_v-\frac{3}{2})^{\frac{M-1}{2}}}{(\frac{M-1}{2})!} \\
&\approx O\Big(\frac{n_v^{\frac{M-1}{2}}}{(\frac{M-1}{2})!}\Big) \qquad \text{if } \; n_v \gg M \geq 0
\end{aligned} \tag{A.5}$$

Hence, combining (A.3) and (A.5), we can approximate (A.1) as follows:

$$\begin{aligned}
O\Big(\binom{n_{\tilde{e}}}{M} n_v\mathcal{V}(2M+n_v-1,M)\Big) &\approx O\Big(\frac{n_{\tilde{e}}^M}{M!}\cdot n_v \cdot \frac{n_v^{\frac{M-1}{2}}}{(\frac{M-1}{2})}!\Big) \\
&= O\Big(\frac{n_{\tilde{e}}^M \; n_v^{\frac{M+1}{2}}}{M! \; (\frac{M-1}{2})!}\Big).
\end{aligned} \tag{A.6}$$

Notation

This section lists some symbols and abbreviation that are frequently used in this thesis.

List of symbols

Sets

$\emptyset$	empty set
int(X)	interior of the set X
$X \subseteq Y$	X is a subset of Y
$x \in X$	x is an element of the set X
$\mathbb{R}$	set of real numbers
$\mathbb{N} \setminus \{0\}$	set of positive integers: $\{1, 2, 3, \dots\}$
$\lfloor x \rfloor$	the largest integer less than or equal to x

Matrices and Vectors

$\mathbb{R}^n$	set of real-valued vectors of size n
$\mathbb{R}^{n\times m}$	set of n by m real-valued matrices
x^T	transpose of the vector x
x_i	i-th element of the vector x
A_{ij}	entry of the matrix A on the i-th row and the j-th column
$\|x\|_p$	p-norm of the vector x
$\|x\|_\infty$	∞-norm of the vector x
$\|A\|_p$	p-norm of the matrix A
$\|A\|_\infty$	∞-norm of the matrix A

Functions

$f : X \to Y$	function f with domain X and codomain Y
$\mathrm{dom}(f)$	domain of the function f
$f(e)$	probability density function of the stochastic vector e
J	objective function
$\mathbb{E}[x]$	expected value of the random variable x
$\mathbb{E}[x^k]$	k-th moment of the random variable x

Max-Plus Algebra

$\mathbb{R}_\varepsilon$	$\mathbb{R} \cup \{-\infty\}$
$\mathcal{S}_{\mathrm{mps}}$	set of max-plus-scaling functions
$\oplus$	max-plus addition
$\otimes$	max-plus multiplication
ε	zero element of the max-plus addition
0	identity element of the max-plus multiplication
$A^{\otimes^k}$	k-th max-plus power of the matrix A

Acronyms

MILP	Mixed-Integer Linear Programming
MMPS	Max-Min-Plus-Scaling
mp-NLP	multi-parametric Nonlinear Programming
mp-MILP	multi-parametric Mixed-Integer Linear Programming
MPC	Model Predictive Control
MPL	Max-Plus-Linear
PWA	Piecewise Affine

Bibliography

[1] M. A. Abramowitz and I. Stegun. *Handbook of Mathematical Functions*. National Bureau of Standards, US Government Printing Office, Washington DC, 1964.

[2] A. Atamtürk and M. W. P. Savelsbergh. Integer-programming software systems. *Annals of Operations Research*, 140(1):67–124, November 2005.

[3] M. Athans. The role and use of the stochastic linear-quadratic-gaussian problem in control system design. *IEEE Transaction on Automatic Control*, 16(6):529–552, 1971.

[4] F. Baccelli, G. Cohen, G. J. Olsder, and J. P. Quadrat. *Synchronization and Linearity*. John Wiley & Sons, New York, 1992.

[5] F. Baccelli and D. Hong. Analytic expansions of max-plus lyapunov exponents. *Annals of Applied Probability*, 10(3):779–827, 2000.

[6] A. Bemporad, F. Borrelli, and M. Morari. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on Automatic Control*, 48(9):1600–1606, 2003.

[7] 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.

[8] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[9] E. A. Bender. *An Introduction to Mathematical Modelling*. John Wiley & Sons, New York, 1978.

[10] W. H. Beyer. *CRC Standard Mathematical Tables*. CRC Press, Boca Raton, Florida, 28th edition, 1987.

[11] J. L. Boimond, L. Hardouin, and P. Chiron. A modeling method of SISO discrete-event systems in max-algebra. In *Proceedings of the 3rd European Control Conference*, pages 2023–2026, Rome, Italy, September 1995.

[12] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences*. Springer, Berlin, Germany, 2003.

[13] A. Bouillard and B. Gaujal. Backward coupling in bounded free-choice nets under markovian and non-markovian assumptions. *Discrete Event Dynamic Systems*, 18(4):473–498, 2008.

[14] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, London, UK, 2004.

[15] S. P. Boyd and C. H. Barratt. *Linear Controller Design: Limits of Performance*. Prentice Hall, Englewood Cliffs, New Jersey, 1991.

[16] J. H. Bramble and S. R. Hilbert. Estimation of linear functionals on sobolev spaces with application to fourier transforms and spline interpolation. *SIAM Journal on Numerical Analysis*, 7(1):112–124, 1970.

[17] B. Büeler, A. Enge, and K. Fukuda. Exact volume computation for convex polytopes: A practical study. In G. Kalai and G.M. Ziegler, editors, *Polytopes – Combinatorics and Computation*, pages 131–154. Birkäuser Verlag, Basel, Switzerland, 2000.

[18] E. F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry*. Springer-Verlag, Berlin, Germany, 1995.

[19] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.

[20] L. O. Chua and A. C. Deng. Canonical piecewise-linear representation. *IEEE Transactions on Circuits and Systems*, 35(1):101–111, January 1988.

[21] L. O. Chua and R. L. P. Ying. Canonical piecewise-linear analysis. *IEEE Transactions on Circuits and Systems*, 30(3):125–140, March 1983.

[22] C. Cordier, H. Marchand, R. Laundy, and L. A. Wolsey. *bc-opt*: A branch-and-cut code for mixed integer programs. *Mathematical Programming, Series A*, 86(2):335–353, 1999.

[23] R. A. Cuninghame-Green. *Minimax Algebra*. Springer-Verlag, Berlin, Germany, 1979.

[24] R. David. Modeling of dynamic systems by Petri nets. In *Proceedings of the 1st European Control Conference*, pages 136–147, Grenoble, France, July 1991.

[25] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, New York, 2nd edition, 1984.

[26] B. De Schutter and B. De Moor. Minimal state space realization of SISO systems in the max algebra. Technical Report 93-57, ESAT-SISTA, K.U.Leuven, Leuven, Belgium, October 1993.

[27] 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.

[28] B. De Schutter, T. van den Boom, and V. Verdult. State space identification of max-plus-linear discrete event systems from input-output data. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 4024–4029, Las Vegas, Nevada, December 2002.

[29] B. De Schutter and T. J. J. van den Boom. Model predictive control for max-min-plus-scaling systems — Efficient implementation. In M. Silva, A. Giua, and J.M. Colom, editors, *Proceedings of the 6th International Workshop on Discrete Event Systems (WODES'02)*, pages 343–348, Zaragoza, Spain, October 2002.

[30] B. De Schutter and T. J. J. van den Boom. MPC for continuous piecewise-affine systems. *Systems & Control Letters*, 52(3–4):179–192, July 2004.

[31] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester. *A Modern Introduction to Probability and Statistics*. Springer, London, UK, 2005.

[32] V. Dua, K. P. Papalexandri, and E. N. Pistikopoulos. Global optimization issues in multiparametric continuous and mixed-integer optimization problems. *Journal of Global Optimization*, 30(1):59–89, 2004.

[33] V. Dua and E. N. Pistikopoulos. An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research*, 99(1-4):123–139, 2000.

[34] R. W. Eglese. Simulated annealing: A tool for operations research. *European Journal of Operational Research*, 46(3):271–281, 1990.

[35] S. S. Farahani, T. van den Boom, and B. De Schutter. An approximation approach for identification of stochastic max-plus linear systems. In *Proceedings of the 18th IFAC World Congress*, pages 8241–8246, Milan, Italy, August–September 2011.

[36] S. S. Farahani, T. van den Boom, and B. De Schutter. Model predictive control for stochastic max-min-plus-scaling systems – An approximation approach. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 391–396, Orlando, USA, December 2011.

[37] S. S. Farahani, T. van den Boom, and B. De Schutter. Exact and approximate approaches to the identification of stochastic max-plus-linear systems. *Submitted to Discrete Event Dynamic Systems*, February 2012.

[38] S. S. Farahani, T. van den Boom, H. van der Weide, and B. De Schutter. An approximation approach for model predictive control of stochastic max-plus linear systems. In *Proceedings of the 10th International Workshop on Discrete Event Systems (WODES'10)*, pages 386–391, Berlin, Germany, 2010.

[39] S. S. Farahani, T. van den Boom, H. van der Weide, and B. De Schutter. An approximation approach for model predictive control of stochastic max-plus linear systems. *Submitted to Systems and Control Letters*, December 2011.

[40] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. John Wiley & Sons, New York, 2nd edition, 1950.

[41] H. Flanders. Differentiation under the integral sign. *The American Mathematical Monthly*, 80(6):615–627, 1973.

[42] T. Fujisawa and E. S. Kuh. Piecewise-linear theory of nonlinear networks. *SIAM Journal on Applied Mathematics*, 22(2):307–328, 1972.

[43] F. Gallot, J. L. Boimond, and L. Hardouin. Identification of simple elements in max-algebra: Application to SISO discrete event systems modelisation. In *Proceedings of the European Control Conference (ECC'97)*, Brussels, Belgium, paper 488, July 1997.

[44] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice — A survey. *Automatica*, 25(3):335–348, May 1989.

[45] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Boston, Massachusetts, 1997.

[46] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.

[47] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, 3rd edition, 1996.

[48] G. C. Goodwin and R. L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York, 1977.

[49] V. V. Gorokhovik and O. I. Zorko. Piecewise affine functions and polyhedral sets. *Optimization*, 31(3):209–221, 1994.

[50] R. M. P. Goverde, B. Heidergott, and G. Merlet. A coupling approach to estimating the lyapunov exponent of stochastic max-plus linear systems. *European Journal of Operational Research*, 210(2):249–257, 2011.

[51] A. Grancharova, T. Johansen, and P. Tondel. Computational aspects of approximate explicit nonlinear model predictive control. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *Lecture Notes in Control and Information Sciences*, pages 181–192. Springer, Berlin, Germany, 2007.

[52] A. Grancharova and T. A. Johansen. Explicit approximate approach to feedback min-max model predictive control of constrained nonlinear systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 4848–4853, San Diego, California, December 2006.

[53] E. T. Hale. *Numerical Methods for d-Parametric Nonlinear Programming with Chemical Process Control and Optimization Applications*. PhD thesis, The University of Texas at Austin, Austin, Texas, August 2005.

[54] J. Harris, J. L. Hirst, and M. Mossinghoff. *Combinatorics and Graph Theory*. Undergraduate Texts in Mathematics. Springer, New York, 2nd edition, 2008.

[55] W. P. M. H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001.

[56] B. Heidergott. *Max-Plus Linear Stochastic Systems and Perturbation Analysis*. Springer, New York, 2006.

[57] B. Heidergott, G. J. Olsder, and J. van der Woude. *Max Plus at Work*. Princeton University Press, Princeton, New Jersey, 2006.

[58] Y. C. Ho, editor. *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World*. IEEE Press, Piscataway, New Jersey, 1992.

[59] R. Isaacs. *Differential Games*. The SIAM Series in Applied Mathematics. John Wiley & Sons, New York, 1965.

[60] A. Jean-Marie and G. J. Olsder. Analysis of stochastic min-max systems: Results and conjectures. *Mathematical and Computer Modelling*, 23(11/12):175–189, 1996.

[61] M. Johansson. *Piecewise Linear Control Systems*, volume 284 of *Lecture Notes in Control and Information Sciences*. Springer, Berlin, Germany, 2003.

[62] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions, Vol. 2*. Wiley, New York, 2nd edition, 1995.

[63] M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods*. Wiley-VCH, Weinheim, Germany, 2008.

[64] 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, December 1992.

[65] J. Kingman and S. Taylor. *Introduction to Measure and Probability*. Cambridge University Press, London, UK, 1966.

[66] A. Korotayev, A. Malkov, and D. Khaltourina. *Introduction to Social Macrodynamics: Compact Macromodels of the World System Growth*. URSS, Moscow, Russia, 2006.

[67] A. Kripfganz and R. Schulze. Piecewise affine functions as a difference of two convex functions. *Optimization*, 18(1):23–29, 1987.

[68] J. B. Lasserre. Integration on a convex polytope. *Proceedings of the American Mathematical Society*, 126(8):2433–2441, August 1998.

[69] D. M. W. Leenaerts and W. M. G. van Bokhoven. *Piecewise Linear Modeling and Analysis*. Kluwer Academic Publishers, Boston, Massachusetts, 1998.

[70] J. Lewin. *Differential Games: Theory and Methods for Solving Game Problems With Singular Surfaces*. Springer-Verlag, New York, 1994.

[71] C. C. Lin and L. A. Segel. *Mathematics Applied to Deterministic Problems in the Natural Sciences*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pennsylvania, 1988.

[72] J. T. Linderoth and T. K. Ralphs. Noncommercial software for mixed-integer linear programming. In J. Karlof, editor, *Reinforcement Learning: State-Of-The-Art*, CRC Press Operations Research Series, pages 253–303. 2005.

[73] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Upper Saddle River, New Jersey, 2nd edition, 1999.

[74] J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Harlow, England, 2002.

[75] J. Mairesse. Stochastic linear systems in the (max,+) algebra. In *11th International Conference on Analysis and Optimization of Systems Discrete Event Systems*, volume 199 of *Lecture Notes in Control and Information Sciences*, pages 257–265. Springer-Verlag, London, UK, 1994.

[76] T. H. Mattheiss and D. S. Rubin. A survey and comparison of methods for finding all vertices of convex polyhedral sets. *Mathematics of Operations Research*, 5(2):167–185, May 1980.

[77] E. Menguy, J. L. Boimond, L. Hardouin, and J. L. Ferrier. A first step towards adaptive control for linear systems in max algebra. *Discrete Event Dynamic Systems*, 10(4):347–368, 2000.

[78] D. S. Mitrinović, J. E. Pečrić, and A. M. Fink. *Classical and New Inequalities in Analysis*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993.

[79] T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall. The double description method. In H.W. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games*, number 28 in *Annals of Mathematics Studies*, pages 51–73. Princeton University Press, Princeton, New Jersey, 1953.

[80] S. G. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York, 1996.

[81] I. Necoara, B. De Schutter, T. van den Boom, and H. Hellendoorn. Model predictive control for uncertain max-min-plus-scaling systems. *International Journal of Control*, 81(5):701–713, May 2008.

[82] I. Necoara, B. De Schutter, T. J. J. van den Boom, and J. Hellendoorn. Model predictive control for perturbed continuous piecewise affine systems with bounded disturbances. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 1848–1853, Paradise Island, The Bahamas, December 2004.

[83] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, Pennsylvania, 1994.

[84] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models:An Algorithmic Approach*. Johns Hopkins Series in the Mathematical Sciences. Johns Hopkins University Press, Mineola, New York, 1981.

[85] Colm Art O'Cinneidea. Phase-type distributions: open problems and a few properties. *Communications in Statistics. Stochastic Models*, 15(4):731–757, 1999.

[86] G. J. Olsder. Eigenvalues of dynamic max-min systems. *Discrete Event Dynamic Systems: Theory and Applications*, 1(2):177–207, September 1991.

[87] 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, March 1990.

[88] S. Ovchinnikov. Max-min representation of piecewise linear functions. *Beiträge zur Algebra und Geometrie/Contributions to Algebra and Geometry*, 43(1):297–302, 2002.

[89] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Singapore, 1991.

[90] P. M. Pardalos and M. G. C. Resende, editors. *Handbook of Applied Optimization*. Oxford University Press, Oxford, UK, 2002.

[91] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[92] J. A. C. Resing, R. E. de Vries, G. Hooghiemstra, M. S. Keane, and G. J. Olsder. Asymptotic behavior of random discrete event systems. *Stochastic Processes and Their Applications*, 36(2):195–216, December 1990.

[93] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[94] V. Sakizlis, K. I. Kouramas, N. P. Faisca, and E. N. Pistikopoulos. Towards the design of parametric model predictive controllers for nonlinear constrained systems. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *Lecture Notes in Control and Information Sciences*, pages 193–205. Springer, Berlin, Germany, 2007.

[95] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, UK, 1986.

[96] G. Schullerus and V. Krebs. Diagnosis of batch processes based on parameter estimation of discrete event models. In *Proceedings of the European Control Conference 2001 (ECC'01)*, pages 1612–1617, Porto, Portugal, September 2001.

[97] G. Schullerus and V. Krebs. Input signal design for discrete event model based batch process diagnosis. In *Proceedings of the 4th IFAC Workshop on On-Line Fault Detection and Supervision in the Chemical Process Industries*, Seoul, Korea, 2001.

[98] G. Schullerus, V. Krebs, B. De Schutter, and T. van den Boom. Optimal input signal design for identification of max-plus-linear systems. In *Proceedings of the 2003 European Control Conference (ECC'03)*, Cambridge, UK, September 2003. Paper 026.

[99] S. Simić. On a converse of Jensen's discrete inequality. *Journal of Inequalities and Applications*, 2009. Article ID 153080.

[100] S. Simić. On an upper bound for Jensen's inequality. *Journal of Inequalities in Pure and Applied Mathematics*, 10(2), 2009. Article 60.

[101] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, West Sussex, UK, 2nd edition, 2005.

[102] E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, April 1981.

[103] 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.

[104] 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.

[105] T. van den Boom and B. De Schutter. Stabilizing model predictive controllers for randomly switching max-plus-linear systems. In *Proceedings of the European Control Conference 2007 (ECC'07)*, pages 4952–4959, Kos, Greece, July 2007.

[106] T. van den Boom and B. De Schutter. Model predictive control for randomly switching max-plus-linear systems using a scenario-based algorithm. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 2298–2303, Atlanta, Georgia, December 2010.

[107] T. van den Boom and B. De Schutter. Modeling and control of switching max-plus-linear systems with random and deterministic switching. *Accepted for publication in Discrete Event Dynamic Systems: Theory and Applications*, 2011.

[108] T. J. J. van den Boom and B. De Schutter. Modelling and control of discrete event systems using switching max-plus-linear systems. *Control Engineering Practice*, 14(10):1199–1211, October 2006.

[109] T. J. J. van den Boom, B. De Schutter, and V. Verdult. Identification of stochastic max-plus-linear systems. In *Proceedings of the 2003 European Control Conference (ECC'03)*, Cambridge, UK, September 2003. Paper 104.

[110] T. J. J. 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.

[111] 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, London, UK, 2000.

[112] S. van Loenhout. Stochastic switching max-plus-linear systems, theory and aspects. Master's thesis, Delft Center for Systems and Control, Delft University of Technology, 2011.

[113] S. van Loenhout, T. van den Boom, S. S. Farahani, and B. De Schutter. Model predictive control for stochastic switching max-plus-linear systems. In *Proceedings of the 11th International Workshop on Discrete Event Systems (WODES'12)*, Guadalajara, Mexico, October 2021.

[114] G. A. Wainer and P. Mosterman Eds. *Discrete-Event Modeling and Simulation: Theory and Applications*. CRC Press, Boca Raton, Florida, 1st edition, 2010.

[115] J. Wang. *Timed Petri Nets: Theory and Application*. The International Series on Discrete Event Dynamic Systems. Springer, New York, 1998.

[116] C. Wen, S. Wang, F. Li, and M. J. Khan. A compact f-f model of high-dimensional piecewise-linear function over a degenerate intersection. *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications*, 52(4):815–821, April 2005.

[117] R. Willink. Normal moments and Hermite polynomials. *Statistics & Probability Letters*, 73(3):271–275, July 2005.

[118] S. J. Wright. *Primal-Dual Interior Point Methods*. SIAM, Philadelphia, Pennsylvania, 1997.

Summary

Approximation Methods in Stochastic Max-Plus Systems

Many physical systems, such as traffic networks, manufacturing systems, chemical systems, or biological systems, are characterized by dynamic phenomena. To model such systems, we are interested in the classes of nonlinear dynamic models that are capable of describing continuous and/or discrete dynamics, i.e., the classes of discrete-event models and hybrid models. The former class consists of systems the evolution of which depends on the occurrence of discrete events over time and the latter class is characterized by the interaction of time-continuous models on the one hand, and logic rules and discrete-event models on the other hand. Hence, the evolution of a hybrid system may depend both on the progress of time and the occurrence of events.

In this PhD thesis, we study some special classes of discrete-event systems, namely stochastic max-plus-linear (MPL) systems and switching MPL systems, and a special class of hybrid systems, namely stochastic max-min-plus-scaling (MMPS) systems. The main operators in these systems are maximization, addition, and minimization (only in MMPS systems). In the identification and control problem of these systems, the objective function appearing in the optimization problem can be written as the expected value of the maximum of affine expressions. The focus of this thesis is on finding an efficient method to compute this expected value since the currently available methods are both too complex and too time-consuming.

To address this issue, this PhD thesis proposes an approximation method based on the higher-order moments of a random variable. By considering the relationship between the infinity-norm and the p-norm of vectors, we obtain an upper bound for the expected value of the maximum of affine expressions. This approximation method can be applied to any distribution that has finite moments and in the case that these moments have a closed form (such as for a uniform distribution, normal distribution, beta distribution, or gamma distribution), the approximation method results in an analytic expression. We also show that the approximate function obtained using this method is convex in the control variable and also discuss the error of this approximation method and show that this error is bounded both from below

and from above.

MPC for and identification of stochastic MPL system

Model predictive control (MPC) among other control approaches for MPL systems such as the residuation methods has an advantage of being capable of handling constraints both on inputs and outputs. Therefore, in this PhD thesis, MPC is chosen as a control approach for stochastic MPL systems. For identification of such systems, we consider state space models and also take the stochastic properties of the system into account in the identification process.

In both identification and MPC, we replace the objective function defined as the expected value of the maximum of affine expressions by the function obtained using the proposed approximation method based on higher-order moments. This approximate function is indeed an upper bound for the objective function and is minimized instead of the objective function itself. Moreover, since the approximate objective function is convex in the control variable and since the gradients with respect to this variable can be computed analytically, we obtain a convex MPC optimization problem that can be solved efficiently using the available convex optimization algorithms. The identification problem as well, can be solved using non-convex gradient-based algorithms due to the existence of analytic expressions for the gradients of the approximate non-convex objective function.

In order to examine the efficiency and accuracy of the approximation method, we present case studies for both MPC and identification problem in which the approximation method is applied using two different noise distributions, namely the uniform distribution and the normal distribution. We compare the results of the approximation method with the ones obtained from other available methods, such as analytic and numerical integration, and Monte Carlo simulation. This comparison shows that in terms of time-efficiency the approximation method is faster in most of the cases while having a performance that is comparable to the performance of the other methods.

MPC for stochastic switching MPL systems and stochastic MMPS systems

We further extend the application of the proposed approximation method to two other classes of discrete-event and hybrid systems, namely stochastic switching MPL systems and stochastic MMPS systems, respectively. Due to the stochastic properties of these systems, applying MPC to control these systems, in general, results in a complex and time-consuming optimization problem.

In stochastic switching MPL systems, we consider both discrete and continuous random variables, which are related to stochastic mode switching and stochastic system parameters, respectively. Hence, the objective function, which is defined as the expected value of the maximum of affine expressions, involves the joint probability distribution of these two types of random variables. In the case that the mode switching variables and the system parameters are independent random variables, we propose to apply the combination of two approximation methods to simplify the MPC optimization problem. The first approximation method is based on a scenario-based algorithm that chooses only the most probable modes that may occur and the second one is the proposed approximation method based on higher-order moments to approximate the expected value of the maximum of affine expressions. In the case that the mentioned random variables are dependent on each other, we propose to approximate the joint probability distribution by multi-variable piecewise polynomial functions to obtain an analytic expression for the objective function. In both cases, the approximation approaches simplify the problem considerably.

In the stochastic MMPS-MPC optimization problem, to be able to apply the proposed approximation method, we first need to rewrite the MMPS objective function as a difference of two convex functions. In this way, it is possible to approximate the expected value of this function by its upper bound. To this end, we approximate the first convex function by its upper bound using the proposed approximation method and the second convex function by its lower bound using Jensen's inequality. In the case that the random variables in the MMPS function have finite moments with closed forms, these upper and lower bounds both lead to analytic expressions. In this case, the whole MMPS-MPC optimization problem can be considered as a set of convex problems and each problem can be solved applying gradient-based convex optimization algorithms since the analytic expressions for the subgradients exist. In an example, we show that this upper bound approach decreases the computational complexity and the computation time considerably.

Min-max optimization and approximate stochastic optimization for MMPS systems

In the last part of this PhD thesis, we study min-max optimization of MMPS systems for the following cases: non-stochastic min-max optimization problems, stochastic minimization problems, and stochastic min-max optimization problems. For each case, we propose solution approaches, namely the mixed-integer linear programming for the first case and the approximation method proposed for MMPS systems for the second and the third case. In this way, the second case can be rewritten as a set of convex optimization problems that can be solved efficiently using convex optimization algorithms and the third case results in a multi-parametric, non-linear (mp-NLP) optimization problem that can be solved using mp-NLP algorithms. Furthermore, we present two applications of such problems where the first problem is an example of a two-player game and the second one is a reference tracking problem.

Samenvatting

Benaderingsmethoden voor Stochastische Max-Plus Systemen

Veel fysische systemen, zoals verkeersnetwerken, productiesystemen, chemische systemen, of biologische systemen, worden gekarakteriseerd door dynamische verschijnselen. Met het oog op het modelleren van zulke systemen zijn we geïnteresseerd in de klassen van niet-lineaire dynamische modellen die in staat zijn om continue en/of discrete dynamica te beschrijven, namelijk de klassen van discrete-gebeurtenismodellen en hybride modellen. De eerste klasse bestaat uit systemen waarvan de evolutie afhankelijk is van het plaatsvinden van discrete gebeurtenissen in de tijd. De tweede klasse wordt gekarakteriseerd door de interactie tussen continuetijdmodellen aan de ene kant en logische regels en discrete-gebeurtenis-modellen aan de andere kant. De evolutie van een hybride systeem kan derhalve van zowel de tijdsprogressie als van het zich voordoen van gebeurtenissen afhangen.

In dit proefschrift bestuderen we enkele speciale klassen van discrete-gebeurtenissystemen, namelijk stochastische max-plus-lineaire (MPL) systemen en schakelende MPL systemen, en een speciale klasse van hybride systemen: de stochastische max-min-plus-schaling (MMPS) systemen. De hoofdoperatoren in deze systemen zijn maximalisatie, optelling, en – voor MMPS systemen – minimalisatie. Bij identificatie en regeling van deze systemen kan de doelfunctie die in het optimalisatieprobleem voorkomt, geschreven worden als de verwachte waarde van het maximum van een aantal affiene uitdrukkingen. De focus van dit proefschrift ligt op het vinden van een efficiënte methode voor het berekenen van deze verwachte waarde, aangezien de huidige hiervoor beschikbare methoden zowel te complex als te rekenintensief zijn.

Ten einde dit probleem aan te pakken, stelt dit proefschrift een benaderingsmethode voor die gebaseerd is op de hogere-orde momenten van een stochastische variabele. Door de relatie tussen de oneindig-norm en de p-norm van vectoren te beschouwen, kunnen we een bovengrens bepalen voor de verwachte waarde van het maximum van een aantal affiene uitdrukkingen. Deze benaderingsmethode kan worden toegepast voor elke distributie met eindige momenten en, in het geval dat deze momenten een gesloten vorm hebben (zoals voor een uniforme distributie, normale distributie, beta-distributie of gamma-distributie), resulteert de benaderingsmethode in een analytische uitdrukking. We laten tevens zien dat de benaderende functie verkregen door het gebruik van deze methode convex is in de regelvariabelen en we bespreken ook de fout van deze benaderingsmethode en laten zien dat deze fout van zowel beneden als van boven begrensd is.

MPC en identificatie van stochastische MPL systemen

Modelgebaseerde voorspellende regeling (in het Engels: *Model Predictive Control* – MPC) heeft in vergelijking met andere regelmethoden voor MPL systemen zoals de residuatiemethoden het voordeel dat het in staat is om met beperkingen op zowel ingangen als uitgangen om te gaan. Daardoor is MPC in dit proefschrift gekozen als regelaanpak voor stochastische MPL systemen. Voor de identificatie van dergelijke systemen beschouwen we toestandsruimtemodellen en nemen we ook de stochastische kenmerken van de systemen in beschouwing in het identificatieproces.

Bij zowel identificatie als MPC vervangen we de doelfunctie, die gedefinieerd is als de verwachte waarde van het maximum van affiene uitdrukkingen, door de functie die verkregen wordt door de voorgestelde benaderingsmethode die gebaseerd is op hogere-orde momenten. Deze benaderende functie is inderdaad een bovengrens voor de doelfunctie en deze bovengrens wordt vervolgens geminimaliseerd in plaats van de doelfunctie zelf. Aangezien de benaderende doelfunctie convex is in de regelvariabelen en aangezien de gradiënten met betrekking tot deze variabelen analytisch berekend kunnen worden, verkrijgen bovendien we een convex MPC optimalisatieprobleem dat efficiënt opgelost kan worden door de beschikbare convexe optimalisatiealgoritmen te gebruiken. Omwille van het bestaan van analytische uitdrukkingen voor de gradiënten van de benaderende niet-convexe doelfunctie, kan het identificatieprobleem ook opgelost worden door het gebruik van niet-convexe gradiënt-gebaseerde algoritmen.

Met het oog op het onderzoeken van de efficiëntie en nauwkeurigheid van de benaderingsmethode, stellen we casestudy's voor MPC en identificatieproblemen voor waarin de benaderingsmethode wordt toegepast door twee verschillende ruisdistributies te gebruiken, namelijk de uniforme distributie en de normale distributie. We vergelijken de resultaten van de benaderingsmethode met deze die verkregen zijn met andere bestaande methoden, zoals analytische en numerieke integratie en Monte-Carlo-simulatie. Deze vergelijking laat zien dat de benaderingsmethode in de meeste gevallen sneller is qua rekentijd, terwijl zij een prestatie heeft die vergelijkbaar is met die van de andere methoden.

MPC voor stochastische schakelende MPL systemen en stochastische MMPS systemen

We breiden de toepassing van de voorgestelde benaderingsmethode verder uit tot twee andere klassen van discrete-gebeurtenis- en hybride systemen, namelijk stochastische schakelende MPL systemen en stochastische MMPS systemen. Door de stochastische kenmerken van deze systemen resulteert de toepassing van MPC voor het regelen van deze systemen in het algemeen in een complex en rekentijdintensief optimalisatieprobleem.

In stochastische schakelende MPL systemen beschouwen we zowel discrete als continue stochastische variabelen, die gerelateerd zijn met respectievelijk de stochastische schakeling tussen verschillende werkingsregimes en stochastische systeemparameters. Daardoor bevat de doelfunctie, die gedefinieerd is als de verwachte waarde van het maximum van affiene uitdrukkingen, de gecombineerde kansdistributie van deze twee typen stochastische variabelen. In het geval dat de variabelen die het schakelen tussen verschilende werkingsregimes bepalen en de systeemparameters onafhankelijke stochastische variabelen zijn, stellen we voor om de combinatie van twee benaderingsmethoden toe te passen voor het MPC-optimalisatieprobleem.

De eerste benaderingsmethode is gebaseerd op een scenario-afhankelijk algoritme dat alleen de meest waarschijnlijke werkingsregimes kiest die kunnen voorkomen en de tweede methode behelst de voorgestelde benaderingsmethode gebaseerd op hogere-orde momenten om de verwachte waarde van het maximum van affiene uitdrukkingen te benaderen. In het geval dat de genoemde stochastische variabelen afhankelijk van elkaar zijn, stellen we voor om de gezamenlijke kansdistributie te benaderen door een multi-variabele stuksgewijze polynomiale functie om zo een analytische uitdrukking te verkrijgen voor de doelfunctie. In beide gevallen vereenvoudigen de benaderingsmethoden het probleem aanzienlijk.

Om in staat te zijn de voorgestelde benaderingsmethode in het stochastische MMPS-MPC optimalisatieprobleem toe te passen, is het nodig om eerst de MMPS doelfunctie om te schrijven als het verschil tussen twee convexe functies. Op deze manier is het mogelijk om de verwachte waarde van deze functie te benaderen door haar bovengrens. Met dit doel benaderen we de eerste convexe functie door haar bovengrens door gebruik te maken van de voorgestelde benaderingsmethode en de tweede convexe functie benaderen we door haar ondergrens door gebruik te maken van de ongelijkheid van Jensen. In het geval dat de stochastische variabelen in de MMPS functie eindige momenten hebben met een gesloten vorm, leiden deze boven- en ondergrenzen tot analytische uitdrukkingen. In dit geval kan het gehele MMPS-MPC optimalisatieprobleem beschouwd worden als een verzameling convexe problemen en elk probleem kan opgelost worden door gradiënt-gebaseerde convexe optimalisatiealgoritmen toe te passen, aangezien analytische uitdrukkingen bestaan voor de subgradiënten. In een voorbeeld laten we zien dat deze bovengrensaanpak de rekenkundige complexiteit en de rekentijd aanzienlijk verlaagt.

Min-max optimalisatie en benaderende stochastische optimalisatie voor MMPS systemen

In het laatste deel van dit proefschrift bestuderen we min-max optimalisatie van MMPS systemen voor de volgende gevallen: niet-stochastische min-max optimalisatieproblemen, stochastische minimalisatieproblemen en stochastische min-max optimalisatieproblemen. Voor elk geval stellen we oplossingsmethoden voor, namelijk lineair programmeren met gehele en reële getallen voor het eerste geval en de benaderingsmethode voorgesteld voor MMPS systemen voor het tweede en het derde geval. Op deze manier kan het tweede geval herschreven worden als een verzameling van convexe optimalisatieproblemen die efficiënt opgelost kunnen worden met behulp van convexe optimalisatiealgoritmen. Het derde geval resulteert in een multi-parametrisch, niet-lineair (mp-NLP) optimalisatieprobleem dat opgelost kan worden met behulp van mp-NLP algoritmen. Daarnaast presenteren we twee toepassingen van dergelijke problemen, waarbij het eerste probleem een voorbeeld is van een spel met twee spelers en waarbij het tweede een referentie-volgprobleem is.

Curriculum Vitae

Samira S. Farahani was born on 1 February 1983 in Tehran, Iran. She obtained her BSc degree in applied mathematics from Sharif University of Technology in June 2005 in Tehran. In August 2006, she received a full MSc scholarship from the department of Applied Mathematics at Delft University of Technology in the Netherlands. She graduated in July 2008 specializing in Risk and Environmental Modeling. During her MSc program, she did an internship at HKV consultants focusing on modeling the corrosion process in steel structures and analyzing the effect of imperfect maintenance on deterioration process. The results of that internship appeared in her MSc thesis. In September 2008, she started her PhD study at Delft Center for Systems and Control (DCSC) at Delft University of Technology. Her PhD project was focused on providing approximation methods for stochastic max-plus systems. During her first year, she succeeded in obtaining the DISC certificate from the Dutch Institute of Systems and Control (DISC) graduate school. In 2011, she spent 2 months as a visiting scholar at the Department of Mechanical and Aerospace Engineering at University of California, San Diego (UCSD). During this visit, which was supervised by Prof.dr. William McEneaney, she applied her approximation method to robust filtering problem. Her main research interests are max-plus systems, model predictive control, and stochastic systems.