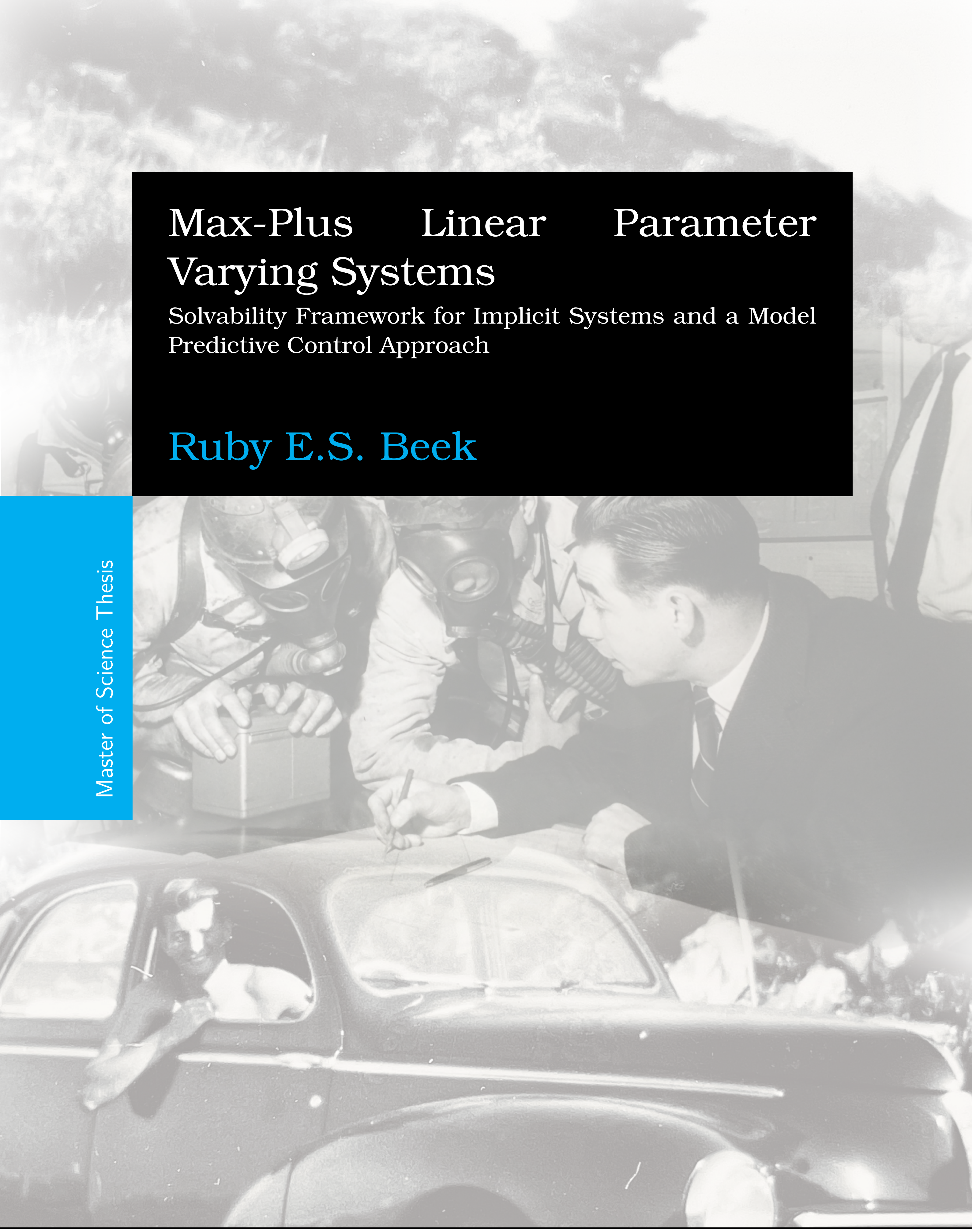


Max-Plus Linear Parameter Varying Systems

Solvability Framework for Implicit Systems and a Model Predictive Control Approach

Ruby E.S. Beek

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Max-Plus Linear Parameter Varying Systems

Solvability Framework for Implicit Systems and a Model Predictive Control Approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Ruby E.S. Beek

June 16, 2022

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Max-Plus Linear Parameter Varying Systems

by

Ruby E.S. Beek

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: June 16, 2022

Supervisor(s): ________________

Dr.ir. A.J.J. van den Boom

____

ir. A. Gupta

Reader(s): ________________

Dr.ir. M.B. Mendel

Abstract

This thesis is devoted to the class of Max-Plus Linear Parameter Varying (MP-LPV) systems. Recently, this class is introduced as an extension of the class of Max-Plus-Linear (MPL) systems in the field of max-plus algebra. The MPL framework is useful when modeling Discrete Event Systems (DES). Describing DES in conventional algebra results in nonlinear system descriptions, but when described as a max-plus linear system, the model becomes 'linear' in max-plus algebra. The extension class of MP-LPV systems is introduced as a tool for parametric modeling, providing the possibility to model uncertain and nonlinear dynamics in a parameterized linear system structure. MP-LPV systems are the max-plus algebraic analogue to the conventional class of Linear Parameter Varying systems. The system matrices of MP-LPV systems can depend on the varying parameter in different manners. Problems arise when the varying parameter depends on the state vector itself. The resulting system description is then implicit. Due to properties of max-plus algebra, it cannot always be guaranteed that such implicit MP-LPV systems have a solution. This leads to the solvability problem. In this thesis, we first define different levels of implicitness in MP-LPV systems, and present frameworks for each level to solve these solvability problems. The results will thereafter be illustrated with a case study that describes an urban railway system. Then, we will present a first model predictive controller for a MP-LPV system with the urban railway system as application. Research about MP-LPV systems has so far been about modeling and system analysis, and little research has been done in control approaches. This model predictive controller can therefore be considered as a first step in controller design for MP-LPV systems. Lastly, the foundation is laid for analyzing closed-loop stability of MP-LPV systems subject to model predictive control.

Table of Contents

Preface and Acknowledgements

Dear reader,

The day I have only dreamed of has come. But, as the wise Albus Dumbledore says, it does not do to dwell on dreams as we then forget to live. Let me therefore keep you awake with this work on max-plus linear parameter varying systems, a promising discrete event system class in the field of max-plus algebra. The last 1,5 years I have devoted myself to this mathematical concept to contribute to the world of Systems and Control, and to finalize the eight year journey at my faculty 3mE of Delft University of Technology. As this was quite a journey, I would like to express my gratitude in the following alineas.

I would like to start of by expressing my graditude to my supervisors Ton and Abhimanyu, who have guided me throughout the process of my thesis and have supported me starting from day one.

Abhimanyu, our weekly meetings in the first year of my thesis process (that sounds longer than it felt) have helped and guided me to where I am now. Your patience, helpfulness and knowledge inspired me to work hard. It is quite a task to navigate a max-plus new born in the complex field of max-plus linear parameter varying systems, and you did an excellent job.

Ton, the last eight months you have been my full time supervisor, and in these months I had the pleasure of having some of the most challenging yet interesting meetings about the advanced mathematics of my subject. Your knowledge in (max-plus) algebra and mathematical skills are admirable, and challenged me to surpass my own capabilities. I am looking forward to reading the book on max-plus linear parameter varying systems.

I would furthermore like to thank Dr. ir. M.B. Mendel, for taking the effort to read and evaluate my work.

I would furthermore like to thank my parents from the bottom of my heart. You have been the best support one can have not only throughout my thesis process, but throughout my whole life. I thank you for the numerous (weekly, almost daily) phone calls that kept me motivated and cheered me up when needed. Thank you also to both my brothers, especially Django, who has lead the way throughout my complete academic career since I was twelve. I can say with complete certainty that I would not have written this thesis without you, just by the simple fact that I then might had chosen another study direction. I cannot promise you that I will not follow you anymore in major life decisions, your choices always sound so good to me... But for now: we did it! [1]

I would like to thank my dear friends of the C.P. Tielestraat for brightening the day even in grey pandemic times. I am happy to end the life as a student as the last one, and join you all in the next phase.

A special thanks furthermore to Lars, for supporting me unconditionally during every moment over the last years. This thesis process was a less lonely ride because of you. You stood by me at all times, even on Mount Stupid and in the valley of despair. You have made though days less though, and happy days even more happy, and I am incredibly grateful for you.

Finally, a shout out to my max-plus friends Bart and Mike. As Bart said, we should have kept up the tradition of presenting our weekly progress. A missed opportunity, but I am nevertheless sure that we continue our friendship. For the interested reader, their thesis works have been a great contribution to the field of max-plus algebra [2], [3].

Thank you all,

Ruby

Delft, University of Technology Ruby E.S. Beek
June 16, 2022

“It does not do to dwell on dreams and forget to live.”
— *Albus Dumbledore, Harry Potter and the Philosopher’s Stone*

Chapter 1

Introduction

This chapter provides a thorough introduction of the research topic covered in this thesis. In section 1-1 we present the background information relevant for this thesis. In section 1-2, we present the motivation for this research based on the literature research [4] prior to this thesis, out of which the research questions will arise. We furthermore present the approach implemented in this thesis to investigate these research questions. Finally in section 1-3 we present the outline of the thesis.

1-1 Background

Everywhere around us, we can find implementations of systems and control theory, and the importance of the world of systems and control is bigger than most of us realise. It can be found in many different areas; wherever feedback occurs, control can be used. Examples are energy delivery, computer engineering, transportation, manufacturing, medical engineering and many more. In order to use and implement control, mathematical models of such systems are a necessity. These models capture the dynamics of the control systems, and provide tools to simulate and investigate these systems. Most of these mathematical models are described with differential equations in conventional algebra and can deal with systems which evolve in time due to the influence of various phenomena such as physical, chemical or biological phenomena [5].

There is however a subclass of systems for which it is of great interest to describe it in another type of algebra, namely in max-plus algebra. These types of systems, referred to as 'man-made' systems, are defined as Discrete Event Systems (DES) and evolve along the occurrence of events instead of time. Examples of DES are manufacturing systems, railway networks, logistic systems and many more [6].

Modeling general DES in conventional algebra results almost always in a nonlinear system description. Control techniques for such complex, nonlinear mathematical models are however more difficult to implement and less general than methods for linear systems. Linear systems are the simplest abstraction on which the classical systems and control theory is based [5]. This gives rise to the motivation of this thesis; there exists a subclass of DES that becomes 'linear' when described in max-plus algebra [6]. The systems in this subclass of DES are generally referred to as Max-Plus-Linear (MPL) systems, and are characterized by synchronization and no concurrency or choice. The combination of synchronization and no concurrency means that a new event can only start as soon as all preceding events have finished. No choice means that the system does not contain steps in which a choice must be made between possible events.

Max-plus algebra distinguishes itself from conventional algebra by replacing the 'classical' operations *addition* and *multiplication* with the operations *maximization* and *addition*, respectively. In DES, the maximization operation corresponds to the property of synchronization, and the addition operation to the duration of events [6].

Recently, an extension of MPL systems has been introduced [7]: MPL systems of which the system matrices depend on an exogenous parameter. Systems with such a linear parameter varying structure are known in conventional algebra as Linear Parameter Varying (LPV) systems, but are now translated to their max-plus analogue. The resulting systems are referred to as Max-Plus Linear Parameter Varying (MP-LPV) systems [8]. Because of the varying parameter structure, nonlinear dynamics can be captured in a model in which the signal relations remain linear. The parameter itself can be nonlinear, but the structure of these systems ensures that the system is still linear ín the parameter [9]. The introduction of this class of MP-LPV systems enables the possibility to obtain clear models of a broad selection of systems that up until now resulted in highly nonlinear descriptions. This gives rise to the motivation for investigating MP-LPV systems; if we can obtain clear model descriptions, we can implement controller design and model analysis on these systems.

There is however a downside of modeling systems in max-plus algebra: there does not exist an inverse operation to the maximization operation. To illustrate, in conventional algebra one can always remove a multiplication on one side of an equation, with a division on the other side. Division can be seen as the inverse operator of multiplication, and vice versa. Such an inverse does not exist for the maximization operation. This absence may cause the occurrence of implicitness in the MP-LPV system description. This implicitness gives rise to a solvability problem; we can not always guarantee existence of a solution to the MP-LPV system.

This solvability problem is earlier investigated by the thesis [10], and is therefore the foundation for this current thesis. Let us summarize the current status of research on MP-LPV systems and the solvability problem. In [10] the property of structural solvability is investigated for different levels of implicitness in MP-LPV systems. Based on graph-theoretical methods, this thesis concludes that for implicit MP-LPV systems, it is impossible to conclude structural solvability. The thesis approaches the solvability problem furthermore by comparing the (implicit) MP-LPV system to the hybrid system class of Max-Min-Plus-Scaling (MMPS) systems. From this, conditions arise which ensures solvability of the concerning systems. Finally, this thesis presents a case study concerning an Urban Railway System (URS), which proves itself to be useful as implementation for analyzing solvability of (implicit) MP-LPV systems. However, this URS still is subject to several unrealistic assumptions, of which the most significant assumption is assuming an unlimited capacity of seats in the trains. The concept of the URS is furthermore studied in [8]. Here, the solvability problem is proposed as a general framework compared to the framework proposed in [10]. This framework is however still under strict assumptions and does not guarantee solvability for all levels of implicitness at all times. The question therefore arises whether we can construct a general solvability framework for MP-LPV systems with all levels of implicitnes and if so, whether we can test this framework with a realistic URS model.

The solvability property is one of the building blocks in the field of system analysis for MP-LPV systems. But as this class is only introduced recently, there are still many topics in different directions open for further research. Such a direction is the field of controller design. Thorough research has been done in the method of Model Predictive Control (MPC) for MPL systems [11]. As MP-LPV systems are an extension of MPL systems, it seems only logical to investigate whether we can extend the tools obtained so far. MPC is a popular control approach, as it can handle constraints on the in- and outputs easily and it can be used to track reference signals. All this background information builds up the interest in defining the research topics of this thesis, and therefore gives rise to the following section.

1-2 Problem Description

Now that we have dived into the background of MP-LPV systems and its status of research at the start of this thesis, we can describe the motivation for the research topics that will be discussed. We have seen that there is a downside to modeling systems as MP-LPV systems, as it gives rise to a certain solvability problem. We have furthermore seen that first steps have been taken in obtaining a solvability framework for such systems. However, the framework obtained so far is subject to many assumptions, and is therefore far from ideal.

The solvability framework is up until now tested with a URS that can be modeled as an MP-LPV system. Again, this system is subject to several unrealistic assumptions and therefore gives rise to the interest in obtaining a more realistic model.

Finally we have seen the motivation for obtaining an MPC approach for MP-LPV systems, as such an approach is already designed for the predecessor class of MPL systems and the MPC approach is a useful and applicable control technique.

We are now ready to formally introduce the research questions that will be answered in the remaining of this thesis.

1-2-1 Research Questions

In this work, we focus on three main research questions, which give rise to several subquestions. Let us present these research questions, after which we will define the approach to answer these questions.

1. *How can we define a solvability framework for general implicit MP-LPV systems?*
   (a) What is the definition of solvability for MP-LPV systems?
   (b) What levels of implicitness can we define for MP-LPV systems?
   (c) Which problems arise due to the different levels of implicitness?
   (d) What are the assumptions in the current solvability framework?

   (e) How can we relax these assumptions, resulting in a new and more general solvability framework?

2. *How can we maintain solvability of the Urban Railway System, while considering a limited capacity of the trains?*

   (a) How can we include a limited capacity in the system description of the URS?

   (b) What assumptions of the current URS description can furthermore be relaxed, resulting in a more realistic URS?

   (c) How can we analyze the solvability of the new URS?

3. *What conditions on MP-LPV systems are necessary such that we can design an MPC controller?*

   (a) Which existing MPC strategies can be extended to the application of MP-LPV systems?

   (b) How can we implement an MPC control design with the updated URS as the concerning system?

   (c) What different disturbances can be added to the system, to investigate if the MPC works properly?

   (d) What conditions are necessary to obtain a stabilizing controller?

1-2-2 Approach

In the first subsection of this chapter, we have seen the motivation and the global background of the research questions. However, before we can answer the research questions, we need a more complete overview of the mathematical background of max-plus algebra and MP-LPV systems in general. We furthermore analyze several structures of hybrid systems, which will show its relevance during the remaining of the thesis. From this mathematical foundation, we can start analyzing the current URS system. For this, we will start by analyzing the URS as an MPL system. As the class of MP-LPV systems are an extension of the class of MPL systems, we can build on from this initial simplified model. By relaxing assumptions step by step, we obtain the URS model with limited capacity, answering research questions 2.(a) and 2.(b). After we obtain this model, we will proceed with the first research question. We implement this order of research, because the step by step formulation of the URS as an MP-LPV model provides us with great insight and knowledge of the MP-LPV system structure. This knowledge is necessary before we can move to a next step in MP-LPV system analysis. For research question 1., the subquestions 1.(a) until 1.(e) provide us a systemic manner in answering the main question. We will approach the solvability problem in a set-theoretical way. The goal of research question 1. is to obtain conditions that can guarantee solvability for all levels of implicitness. After we have obtained the new and more general solvability framework, we can analyze solvability of the updated URS, which answers question 2.(c). After finalizing the solvability analysis we arrive at the phase of implementation. We will use the discussed classes of hybrid systems to develop an MPC approach for the URS in MATLAB using GUROBI and YALMIP, answering question 3.(a) and 3.(b). Thereafter, we will simulate several scenarios to investigate the controllers' performance and its resistance to different disturbances, answering question 3.(c). Finally, we will prove that under mild assumptions, we can prove closed-loop stability for the MP-LPV system answering the final subquestion 3.(d).

1-3 Outline

This thesis is organized as follows. Chapter 2 is devoted to the mathematical background of max-plus algebra and hybrid systems. This chapter furthermore introduces the first version of the URS, here considered in the MPL framework. Chapter 3 focuses on LPV systems in the conventional and the max-plus algebraic sense, and introduces the different levels of implicitness that are considered in the rest of the thesis. This chapter furthermore provides two case studies, both providing a more elaborate model of the URS, now in the MP-LPV framework as well as in a hybrid model format. Chapter 4 is devoted to the updated solvability framework. Per level of implicitness, a solvability framework is described. In Chapter 5, the MPC design for the URS is described, including the simulations of the different scenarios. Finally, we conclude on the thesis work in Chapter 6, and suggestions for future work are collected in Chapter 7.

Chapter 2

Max-Plus Algebra and Hybrid Systems

In this chapter we introduce several important definitions and properties of max-plus algebra and some important classes of hybrid systems. Max-plus algebra is introduced as a tool for Discrete Event Systems (DES) in the early 1980's [12]. The motivation for max-plus algebra originates from the synchronization property in DES, which means that a new event can only start as soon as all preceding events are finished. This property is a non smooth and nonlinear phenomenon when modeled in the 'standard' system theory, but when modeled in a max-plus algebraic structure, this property can be defined by equations linear in max-plus algebra. This chapter contains all important max-plus algebraic mathematical notions and definitions that will be used in the chapters hereafter. This chapter furthermore describes three classes of hybrid systems. Hybrid system theory is introduced to model systems that involve both continuous as discrete dynamics and it can describe time-driven as well as event-driven systems. The last decades, multiple model structures have been designed to model different types of hybrid systems, each structure with its own benefits. The classes that will be presented in this chapter, will be relevant for the remainder of this thesis. The max-plus algebraic part of this chapter is mostly based on [13], unless stated otherwise. The hybrid system theory is based on [14]. We therefore recommend these references if additional reading and understanding is desired. This chapter is organized as follows. In the first section 2-1 the basic concepts, definitions and algebraic properties of max-plus algebra are introduced. Section 2-2 introduces max-plus algebraic matrices and operations with max-plus algebraic matrices. Thereafter in section 2-3 spectral theory is extended to max-plus matrices. Section 2-4 introduces a special type of systems, namely Max-Plus-Linear (MPL) systems. Section 2-5 is devoted to the hybrid system theory. Finally in section 2-6 an example of an MPL system is given.

2-1 Basic Concepts and Definitions

In this section multiple properties and concepts of max-plus algebra are introduced, including some max-plus analogues of basic properties as known in conventional algebra. This section is completely based on [13, paragraph 1.1]. Max-plus algebra is characterized by the use of only two binary operations, namely maximization and addition. These operations are represented by $\oplus$ (pronounced as 'oplus') and $\otimes$ (pronounced as 'otimes') respectively:

$$x \oplus y \stackrel{\text{def}}{=} \max(x, y) \qquad\qquad x \otimes y \stackrel{\text{def}}{=} x + y \tag{2-1}$$

With $x, y \in \mathbb{R}_\varepsilon$ and $\mathbb{R}_\varepsilon$ defined as the set $\mathbb{R} \cup \{-\infty\}$ and $\mathbb{R}$ the set of real numbers. This notation is adopted from [15] and is different than the notation used in the book [13], which uses the notation $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$. It will be clear that the operators $\oplus$ and $\otimes$ have a remarkable resemblance with $+$ and $\times$ in the conventional algebra.

Just as in conventional algebra, $\otimes$ has priority over $\oplus$. Furthermore max-plus algebra defines a zero element as $\varepsilon \stackrel{\text{def}}{=} -\infty$ and a unit element as $e \stackrel{\text{def}}{=} 0$. One can easily observe that $\max(x, -\infty) = x \oplus \varepsilon = x$ and $x + (-\infty) = x \otimes \varepsilon = \varepsilon \; \forall x \in \mathbb{R}_\varepsilon$. For $\otimes$ this means that the zero element ε is *absorbing* for $\otimes$.

Moreover, it clearly holds that $\max(x, y) = \max(y, x)$ and thus $x \oplus y = y \oplus x \; \forall x, y \in \mathbb{R}_\varepsilon$. Similarly it holds that $x + y = y + x$ and also $x \otimes y = y \otimes x \; \forall x, y \in \mathbb{R}_\varepsilon$ for both $\oplus$ and $\otimes$. This property for both $\oplus$ and $\otimes$ is called *commutativity.* Furthermore we can extend the property of *associativity* in conventional algebra to max-plus algebra. This property is characterized in conventional algebra by the condition that $x + (y + z) = (x + y) + z$ and $x \cdot (y \cdot z) = (x \cdot y) \cdot z \; \forall \; x, y, z \in \mathbb{R}$. In max-plus algebra, we equivalently have $x \oplus (y \oplus z) = (x \oplus y) \oplus z$ or $\max(x, \max(y, z)) = \max(\max(x, y), z)$ and furthermore $x \otimes (y \otimes z) = (x \otimes y) \otimes z$ or $x + (y + z) = (x + y) + z \; \forall \; x, y, z \in \mathbb{R}_\varepsilon$. Finally it can be observed that $\max(x, x) = x \oplus x = x$ $\forall x, y \in \mathbb{R}_\varepsilon$, which is called *idempotency* of $\oplus$. Combining the nonempty set $\mathbb{R}_\varepsilon$ and the binary operations $\oplus$ and $\otimes$ defines the commutative and idempotent max-plus *semiring* $\mathbb{R}_{\max}$:

$$\mathbb{R}_{\max} = (\mathbb{R}_\varepsilon, \oplus, \otimes) \tag{2-2}$$

From [13, Lemma 1.2] we see that idempotency of $\oplus$ implies that there does not exist an inverse element with respect to $\oplus$. This can be visualized with the use of an example:

Example 2.1. *We try to find the solution x in the following equation:*

$$4 \oplus x = 2 \tag{2-3}$$

But as Equation 2-3 reads as $\max(4, x) = 2$*, it can be concluded that there exists no number $x \in \mathbb{R}_\varepsilon$ that this equation holds; if $x \leq 4$, $4 \oplus x = 4$, and if $x > 4$, $4 \oplus x = x > 4$.*

There does however exist an inverse element for $x \in \mathbb{R}$ with respect to $\otimes$ in the same manner as in conventional algebra. An example of this is $3 \otimes x = 3 + x = 1$ in which $x = (-2) \in \mathbb{R}$. There does not exist an inverse element for ε with respect to $\otimes$ as ε is absorbing for $\otimes$.

Finally we introduce a definition for max-plus algebraic powers. Let $x \in \mathbb{R}_\varepsilon$, then we define for $r \in \mathbb{R}$:

Definition 2.1 (Max-Plus Algebraic Power)**.**

$$x^{\otimes^r} \stackrel{\text{def}}{=} \underbrace{x \otimes x \otimes \ldots \otimes x}_{r \text{ times}} = \underbrace{x + x + \ldots + x}_{r \text{ times}} \tag{2-4}$$

Notice that the algebraic power $x^{\otimes^r}$ reads as $r \times x$ in conventional algebra. Furthermore, we define $x^{\otimes^e} \stackrel{\text{def}}{=} e = 0 \; \forall x \in \mathbb{R}_\varepsilon$ and therefore also by definition $\varepsilon^{\otimes^e} = e$. Finally we say that $\varepsilon^{\otimes^r} = \varepsilon$ for $r > e$ since ε is absorbing for $\otimes$, and $\varepsilon^{\otimes^r}$ with $r < e$ not defined.

2-2 Max-Plus Algebraic Matrices

In this section the max-plus operations $\oplus$ and $\otimes$ are extended to matrix operations. This section is completely based on [13, Paragraph 1.2]. Let us define a $n \times m$ matrix $\in \mathbb{R}_\varepsilon^{n\times m}$ with $n, m \in \mathbb{Z}_+$ and $\mathbb{Z}_+$ the set of positive integers. This matrix A can be written as:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \qquad \text{with element } a_{ij} = [A]_{ij}, \;\; i \in n, j \in m \tag{2-5}$$

The $\oplus$ and $\otimes$ operations are extended to matrices in the following manner:

$$\begin{aligned} [A \oplus B]_{ij} &= a_{ij} \oplus b_{ij} \\ &= \max(a_{ij}, b_{ij}), \\ [A \otimes C]_{ij} &= \bigoplus_{k=1}^{m} a_{ik} \otimes c_{kj} \\ &= \max_{k=1,\dots,m} (a_{ik} + c_{kj}), \end{aligned} \tag{2-6}$$

With matrices $A, B \in \mathbb{R}_\varepsilon^{n\times m}$, matrix $C \in \mathbb{R}_\varepsilon^{m\times l}$ and $i \in n$, $j \in m$ and $k \in l$ and $n, m, l \in \mathbb{Z}_+$. These max-plus matrix operations are analogue to the matrix sum and multiplication respectively in conventional algebra, which is illustrated in the following example:

Example 2.2. *Let* $A = \begin{bmatrix} 1 & 2 & \varepsilon \\ e & 3 & 4 \\ -5 & \varepsilon & 6 \end{bmatrix}$ *and* $B = \begin{bmatrix} 4 & \varepsilon & 5 \\ 6 & -7 & e \\ \varepsilon & -1 & 2 \end{bmatrix}$. *Using Equation 2-6 gives the following result:*

$$\begin{aligned} A \oplus B &= \begin{bmatrix} 1 \oplus 4 & 2 \oplus \varepsilon & \varepsilon \oplus 5 \\ e \oplus 6 & 3 \oplus -7 & 4 \oplus e \\ -5 \oplus \varepsilon & \varepsilon \oplus -1 & 6 \oplus 2 \end{bmatrix} = \begin{bmatrix} \max(1,4) & \max(2,\varepsilon) & \max(\varepsilon,5) \\ \max(e,6) & \max(3,-7) & \max(4,e) \\ \max(-5,\varepsilon) & \max(\varepsilon,-1) & \max(6,2) \end{bmatrix} \\ &= \begin{bmatrix} 4 & 2 & 5 \\ 6 & 3 & 4 \\ -5 & -1 & 6 \end{bmatrix} \end{aligned} \tag{2-7}$$

$$
\begin{aligned}
A \otimes B &= \begin{bmatrix} 1 \otimes 4 \oplus 2 \otimes 6 \oplus \varepsilon \otimes \varepsilon & 1 \otimes \varepsilon \oplus 2 \otimes -7 \oplus \varepsilon \otimes -1 & 1 \otimes 5 \oplus 2 \otimes e \oplus \varepsilon \otimes 2 \\ e \otimes 4 \oplus 3 \otimes 6 \oplus 4 \otimes \varepsilon & e \otimes \varepsilon \oplus 3 \otimes -7 \oplus 4 \otimes -1 & e \otimes 5 \oplus 3 \otimes e \oplus 4 \otimes 2 \\ -5 \otimes 4 \oplus \varepsilon \otimes 6 \oplus 6 \otimes \varepsilon & -5 \otimes \varepsilon \oplus \varepsilon \otimes -7 \oplus 6 \otimes -1 & -5 \otimes 5 \oplus \varepsilon \otimes e \oplus 6 \otimes 2 \end{bmatrix} \\
&= \begin{bmatrix} \max((1+4),(2+6),(\varepsilon+\varepsilon)) & \max((1+\varepsilon),(2+-7),(\varepsilon+-1)) & \max((1+5),(2+e),(\varepsilon+2)) \\ \max((e+4),(3+6),(4+\varepsilon)) & \max((e+\varepsilon),(3+-7),(4+-1)) & \max((e+5),(3+e),(4+2)) \\ \max((-5+4),(\varepsilon+6),(6+\varepsilon)) & \max((-5+\varepsilon),(\varepsilon+-7),(6+-1)) & \max((-5+5),(\varepsilon+e),(6+2)) \end{bmatrix} \\
&= \begin{bmatrix} 5 \oplus 8 \oplus \varepsilon & \varepsilon \oplus -5 \oplus \varepsilon & 6 \oplus 2 \oplus \varepsilon \\ 4 \oplus 9 \oplus \varepsilon & \varepsilon \oplus -4 \oplus -3 & 5 \oplus 3 \oplus 6 \\ -1 \oplus \varepsilon \oplus \varepsilon & \varepsilon \oplus \varepsilon \oplus -5 & e \oplus \varepsilon \oplus 8 \end{bmatrix} = \begin{bmatrix} 8 & -5 & 6 \\ 9 & -4 & 6 \\ -1 & -5 & 8 \end{bmatrix}
\end{aligned} \tag{2-8}
$$

Furthermore, we can define a max-plus algebraic null matrix $\mathcal{E}$ and a max-plus algebraic identity matrix E:

$$
\begin{aligned}
\mathcal{E} &= \begin{bmatrix} \varepsilon & \varepsilon & \dots & \varepsilon \\ \varepsilon & \varepsilon & \dots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \dots & \varepsilon \end{bmatrix} \\
E &= \begin{bmatrix} e & \varepsilon & \dots & \varepsilon \\ \varepsilon & e & \dots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \dots & e \end{bmatrix}
\end{aligned} \tag{2-9}
$$

It can be seen that for any matrix $A \in \mathbb{R}_\varepsilon^{n \times m}$ with $n, m \in \mathbb{Z}_+$ we have, equivalently to conventional algebra, the following:

$$
\begin{aligned}
A \oplus \mathcal{E}_{n \times m} &= A = \mathcal{E}_{n \times m} \oplus A \\
A \otimes E_{m \times m} &= A = E_{n \times n} \otimes A
\end{aligned} \tag{2-10}
$$

We finally define the max-plus algebraic power for a matrix $A \in \mathbb{R}_\varepsilon^{n \times n}$ with $n, k \in \mathbb{Z}_+$ as:

$$
A^{\otimes^k} \stackrel{\text{def}}{=} \underbrace{A \otimes A \otimes \ldots \otimes A}_{k \text{ times}} = A \otimes A^{\otimes^{k-1}} \tag{2-11}
$$

And we furthermore define $A_{n \times n}^{\otimes^0} \stackrel{\text{def}}{=} E_{n \times n}$.

2-3 Spectral Theory

In this section the spectral theory of max-plus matrices is discussed. The section is organized as follows. First, we shortly introduce the basic notions of graph theory in subsection 2-3-1 based on [13, paragraph 2.1]. These notions are necessary for understanding the next subsection, subsection 2-3-2, which introduces eigenvalues and eigenvectors of max-plus matrices, based on [13, paragraph 2.2].

2-3-1 Graph Theory

A *graph* is defined as $\mathcal{G} = (\mathcal{N}, \mathcal{D})$, with $\mathcal{N}$ the set of *nodes* (or *vertices*) and $\mathcal{D} \subset \mathcal{N} \times \mathcal{N}$ the set of *links*. A graph can be *directed* or *undirected*, and the links can be either *weighted* or *unweighted*. A *path* is defined as a sequence of nodes $i_0, i_1, ..., i_l$ with all nodes in $\mathcal{N}$, for which holds that $i_h \neq i_k$ for all $0 \leq h < k \leq l$, but possibly $i_0 = i_l$. If indeed $i_0 = i_l$, the path is called a *circuit*. A graph $\mathcal{G}$ is called *strongly connected* if for any two nodes $i, j \in \mathcal{N}$, there exists a path from node i to j. The *average weight* of a path is defined by the sum of the weights of all links constituting the path, divided by the length of this path. The average weight of a circuit is defined equivalently.

For every square matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$, a graph can be constructed, which then is called the *communication graph* of matrix A, defined as $\mathcal{G}(A) = (\mathcal{N}, \mathcal{D})$. The numerical value of entry $a_{ji} \in A$ represents the weight of the link from node i to j, and if there is no link we have that $a_{ji} = \varepsilon$. Matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$ is called *irreducible* if its communication graph $\mathcal{G}(A)$ is strongly connected. Analogously, matrix A is called *reducible* if $\mathcal{G}(A)$ is not strongly connected. In that case, we can often transform $A \in \mathbb{R}_\varepsilon^{n\times n}$ intro the *Frobenius normal form* $\tilde{A}$ [16]:

$$P \otimes A \otimes P^{\otimes^{-1}} = \tilde{A} = \begin{bmatrix} A_{11} & \varepsilon & \dots & \varepsilon \\ A_{21} & A_{22} & \dots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ A_{r1} & A_{r2} & \dots & A_{rr} \end{bmatrix} \tag{2-12}$$

In which $P \in \mathbb{R}_\varepsilon$ is a suitable max-plus permutation matrix and $A_{11}, ..., A_{rr}$ are irreducible square submatrices of A. This transformation is not always possible, as not all matrices are max-plus invertible (a matrix is max-plus invertible if in each row an column a single finite element is present). We summarize the definitions stated in this subsection with the use of an example:

Example 2.3. *Let matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$ be defined as*:

$$A = \begin{bmatrix} 2 & 1 & \varepsilon \\ \varepsilon & \varepsilon & 4 \\ 6 & e & \varepsilon \end{bmatrix} \tag{2-13}$$

The communication graph $\mathcal{G}(A)$ associated with matrix A can be found in Figure 2-1, with node set $\mathcal{N} = \{1, 2, 3\}$ and link set $\mathcal{D} = \{(1,1), (1,3), (2,1), (2,3), (3,2)\}$. As there is a weight and a direction associated to each link, we have a directed, weighted graph. It can furthermore be observed that as for any two nodes $i, j \in \mathcal{N}$, a path exists from node i to j, and therefore the graph is strongly connected. Because of this, the matrix A is irreducible.

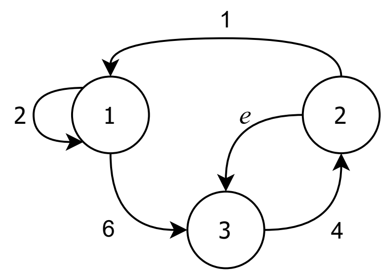


Figure 2-1: Communication Graph of Equation 2-13

2-3-2 Eigenvalues and Eigenvectors

One of the remarkable analogue properties of max-plus algebra in comparison with conventional algebra is the existence of an *eigenvalue* and *eigenvector*. We use [13, Definition 2.4] to state the following:

Definition 2.2 (Max-Plus Algebraic Eigenvalue and Eigenvector)**.** *Let matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$. If there exist scalar $\lambda \in \mathbb{R}_\varepsilon$ and vector $v \in \mathbb{R}_\varepsilon^n$ that contains at least one finite (not equal to ε) entry such that*

$$A \otimes v = \lambda \otimes v, \tag{2-14}$$

then λ is the max-plus algebraic eigenvalue *and v the corresponding max-plus* eigenvector *of matrix A.*

Using the definitions stated in the previous subsection, we can analyze the meaning of a max-plus eigenvalue. For this, we state the following theorem, based on [12, paragraph 3.2.4]:

Theorem 2.1. *If A is irreducible (and thus communication graph $\mathcal{G}(A)$ is strongly connected), there exists one and only one eigenvalue λ and at least one associated eigenvector v. This eigenvalue λ is equal to the maximum average weight over all elementary circuits.*

In the proof of this theorem, [12, paragraph 3.2.4] also proves the general existence of (nonzero) λ and v. The proof furthermore states that if matrix A is not irreducible and therefore $\mathcal{G}(A)$ not strongly connected, the eigenvalue λ is *not* unique and there can be multiple eigenvalues. In this reducible case, each (finite) eigenvalue still represents the average weight of some circuit of the communication graph. In [15] it is stated that the total number of eigenvalues of $A \in \mathbb{R}_\varepsilon^{n\times n}$, including multiplicities, is often less than n. This is in contrast to the eigenvalue problem known in conventional algebra, for which holds that the total number of eigenvalues (including multiplicities) is always equal to the dimension of the matrix.

It remains to show how to obtain max-plus algebraic eigenvalues and eigenvectors. There are multiple algorithms to obtain the eigenvalues. A first one can clearly be found in Theorem 2.1, which implies that to obtain the eigenvalue of an irreducible matrix, the maximum average weight over all elementary circuits needs to be obtained. To obtain this weight, *Karp's Algorithm* can be used, which can be found in [17]. This algorithm however does not provide the corresponding max-plus algebraic eigenvector. [13, Lemma 2.7] provides a method to obtain a corresponding eigenvector to a max-plus algebraic eigenvalue. This method makes use of multiple definitions, which will be introduced shortly as they will be useful later on. First of all, this method makes use of the *Kleene star* operator *:

Definition 2.3. [Kleene star operator *] *Let matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$ and E the max-plus algebraic identity matrix of corresponding size. Then the* Kleene star operator * *for A is defined as A^*:*

$$A^* = E \oplus A \oplus \ldots \oplus A^n \oplus A^{n+1} \oplus \ldots = \bigoplus_{k\geq 0} A^{\otimes k} \tag{2-15}$$

The entries of $[A^*]_{ij}$ are equal to the maximum weight of any path between node j and i of arbitrary length. Based on the Kleene star operator, we define the following:

Definition 2.4.

$$\begin{aligned} A^* &= E \oplus A^+ \\ A^+ &= A \oplus \ldots \oplus A^n \oplus A^{n+1} \oplus \ldots \\ &= A \otimes (E \oplus A \oplus \ldots \oplus A^n \oplus A^{n+1} \oplus \ldots) \\ &= A \otimes A^* \end{aligned} \tag{2-16}$$

Furthermore, we define the *normalized* matrix of matrix A with finite non$-\varepsilon$ eigenvalue λ as A_λ with $[A_\lambda]_{ij} = a_{ij} - \lambda$. The maximum average weight of the normalized matrix is clearly equal to e. The Kleene star operator can be applied to the normalized matrix in the following manner:

$$A_\lambda^* = E \oplus A_\lambda^+ \quad \text{and} \quad A_\lambda^+ = A_\lambda \otimes A_\lambda^* \tag{2-17}$$

With these definitions, the steps prior to [13, Lemma 2.7] can be followed, such that the following is obtained:

Definition 2.5. *Let matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$ with communication graph $\mathcal{G}(A)$ with finite maximal average circuit weight λ. Then λ is an eigenvalue of A and the column $[A_\lambda^*]_{\cdot\eta}$ is an* eigenvector *of A associated with λ, for any node η in $\mathcal{G}^c(A)$.*

In which the graph $\mathcal{G}^c(A) = (\mathcal{N}^c, \mathcal{D}^c)$ is defined as the critical graph of matrix A, containing all critical circuits. A circuit of $\mathcal{G}(A)$ is critical if its average weight is maximal. Another algorithm that yields directly both the max-plus algebraic eigenvalue and the corresponding eigenvector of an irreducible matrix, is the *power algorithm* which can be found extensively in [18]. Finally *Howard's algorithm* [13, Paragraph 6.1], or also known as the policy iteration algorithm, can be used to obtain the generalized eigenmode of a max-plus algebraic matrix that is not strongly connected.

2-4 Max-Plus Linear Systems

As previously mentioned a big advantage of formulating certain types of systems in max-plus algebra, is that those models become "linear". This section describes this specific type of systems, namely the Discrete Event Systems (DES) in which only synchronization and no choice can occur. First, we will explain more extensively this type of systems and its characteristics. Thereafter we will introduce two types of systems of MPL equations, after which we will formulate a DES in max-plus algebra, resulting in a state-space description of an MPL system.

2-4-1 Discrete Event Systems

As mentioned before, we are looking at Discrete Event Systems (DES) in which only synchronization and no choice can occur. A DES is a dynamic, event-driven system, thus the state of the system is based on the occurrence of discrete events. Synchronization means that a new event can only start as soon as all preceding events have been finished. No choice clearly means that the system cannot contain a step in which a choice must be made between possible events. This particular class of DES is only a small subclass of all DES, but it will follow that there are many interesting methods available to analyze and control such systems. Examples of this subclass of DES are manufacturing systems such as production systems, railroad networks, queuing systems and legged robots ([12], [19], [13] and [20] respectively).

2-4-2 Systems of Max-Plus Linear Equations

There are many different types of systems of MPL equations. In this subsection, we will discuss the solution of two types of systems, which will be useful later on in the analysis of different types of system models. It will be clear that, in contrast with their conventional analogue, these systems of max-plus linear equations not always has a solution. This is due to the absence of the inverse of the maximization operation. The two types of systems that we will discuss are $A \otimes x = b$ and $x = A \otimes x \oplus b$, based on [12].

$\boldsymbol{A \otimes x = b}$

Let matrix $A \in \mathbb{R}_\varepsilon^{n \times n}$ and vector $b \in \mathbb{R}_\varepsilon^n$. One can observe that the system of linear equations $A \otimes x = b$ does not always have a solution $x \in \mathbb{R}_\varepsilon^n$. This can be shown with the use of an example:

Example 2.4. *Let* $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ *and* $b = \begin{bmatrix} \varepsilon \\ e \end{bmatrix} = \begin{bmatrix} -\infty \\ 0 \end{bmatrix}$. *Then we have for* $A \otimes x = b$, *following the matrix operations as in section 2-2:* $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \otimes x_1 \oplus 2 \otimes x_2 \\ 3 \otimes x_1 \oplus 4 \otimes x_2 \end{bmatrix}$. *One can see that there exists no* $x \in \mathbb{R}_\varepsilon^n$ *such that this is equal to* $\begin{bmatrix} \varepsilon \\ e \end{bmatrix}$.

It can however be observed that there can be found a so called *subsolution*, which is defined as the following:

Definition 2.6 (Subsolution)**.** *Let matrix* $A \in \mathbb{R}_\varepsilon^{n \times n}$ *and vector* $b \in \mathbb{R}_\varepsilon^n$. *The solution* $x \in \mathbb{R}_\varepsilon^n$ *is called a* subsolution *of the system of max-plus linear equations* $A \otimes x = b$ *if it holds that* $A \otimes x \leq b$.

In Example 2.4, a subsolution is given by $x = [\varepsilon \quad \varepsilon]^T$, such that we obtain for the system of linear equations $A \otimes x = [\varepsilon \quad \varepsilon]^T \leq [\varepsilon \quad e]^T$. It is always possible to determine the largest subsolution which is given by [12, Theorem 3.21]:

Definition 2.7 (Largest subsolution)**.** *Let matrix* $A \in \overline{\mathbb{R}}_\varepsilon^{n \times n}$ *and vector* $b \in \overline{\mathbb{R}}_\varepsilon^n$ *with* $\overline{\mathbb{R}}_\varepsilon = \mathbb{R}_\varepsilon \cup \{+\infty\}$. *The largest subsolution to the system of max-plus linear equations* $A \otimes x = b$ *exists and is given by:*

$$-x_j = \max_i(-b_i + A_{ij}) = \bigoplus_{i \geq 1}^{n} (-b_i \otimes A_{ij}) \tag{2-18}$$

$\boldsymbol{x = A \otimes x \oplus b}$

Let again matrix $A \in \mathbb{R}_\varepsilon^{n \times n}$ and vector $b \in \mathbb{R}_\varepsilon^n$. We will now consider an implicit form of a system of MPL equations, namely $x = A \otimes x \oplus b$. Based on [12, Theorem 3.17] we define the following solution $x \in \mathbb{R}_\varepsilon^n$:

Lemma 2.1. *Let $A \in \mathbb{R}_\varepsilon^{n \times n}$ be a matrix of which the communication graph $\mathcal{G}(A)$ only has circuits of non-positive weight. Then the solution x of $x = A \otimes x \oplus b$ is given by:*

$$x = A^* \otimes b \tag{2-19}$$

In which A^* is defined as in Equation 2-15. Furthermore, if the circuit weights are negative instead of non-positive, this solution x is unique.

2-4-3 Max-Plus Linear State Space Description

Based on [6], we see that we can model a DES with only synchronization and no choice by a max-plus algebraic model in the following manner:

Definition 2.8 (Max-Plus Linear State-Space Description)**.**

$$\begin{aligned} x(k) &= A \otimes x(k-1) \oplus B \otimes u(k) \\ y(k) &= C \otimes x(k) \end{aligned} \tag{2-20}$$

In which $A \in \mathbb{R}_\varepsilon^{n \times n}$, $B \in \mathbb{R}_\varepsilon^{n \times m}$, $C \in \mathbb{R}_\varepsilon^{l \times n}$ with m the number of inputs and l the number of outputs. Furthermore, the vector $x(k) \in \mathbb{R}_\varepsilon^n$ represents the state, the vector $u(k) \in \mathbb{R}_\varepsilon^m$ the input and vector $y(k) \in \mathbb{R}_\varepsilon^l$ the output. Finally it is important to note that $x(k)$, $y(k)$ and $u(k)$ are event times and k is an event counter. Note that this is different than the general state-space description in conventional algebra, in which k represents a time variable. The $\oplus$ operator corresponds to the synchronization property, and the $\otimes$ operator corresponds to the duration of events. As shown in section 2-2 and 2-3, there is a great analogy between the conventional operations $+$ and $\times$ and the max-plus algebraic operations $\oplus$ and $\otimes$ respectively. Because of this, it can clearly be seen that this state space model Equation 2-20 "looks" linear. We will show in the following example based on [21] how a DES can be written in max-plus equations, resulting in an MPL system:

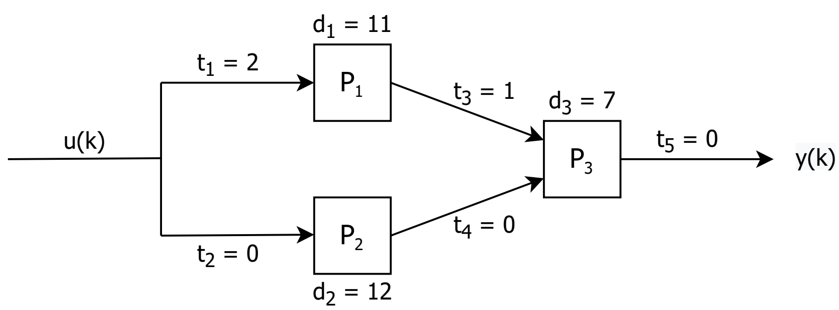


Figure 2-2: Production system of Example 2.5.

Example 2.5. *Figure 2-2 shows a production system containing three processing steps P_1, P_2 and P_3. The system works in batches, with one batch one finished product. The time units d_1, d_2 and d_3 represent the processing times of each processing step. The time units t_1, t_2, ..., t_5 represent the time it takes for the material to go from one process to another. A processing step can only start working on a new product if it is finished with the previous product. Each processing step starts working as soon as all parts are available. The input $u(k)$ represents the k-th time raw material is fed into the system. The states $x_i(k)$ will represent the k-th time that process P_i is executed. The output $y(k)$ represents the k-th time that a finished product is delivered. We can represent this production system by the following max-plus equations:*

$$\begin{aligned} x_1(k) &= \max(x_1(k-1)+11, u(k)+2) \\ x_2(k) &= \max(x_2(k-1)+12, u(k)+0) \\ x_3(k) &= \max(x_1(k)+11+1, x_2(k)+12+0, x_3(k-1)+7) \\ &= \max(x_1(k)+12, x_2(k)+12, x_3(k-1)+7) \\ y(k) &= x_3(k)+7+0 \end{aligned} \tag{2-21}$$

Or equivalently in max-plus notation:

$$\begin{aligned} x_1(k) &= 11 \otimes x_1(k-1) \oplus 2 \otimes u(k) \\ x_2(k) &= 12 \otimes x_2(k-1) \oplus u(k) \\ x_3(k) &= 23 \otimes x_1(k-1) \oplus 24 \otimes x_2(k-1) \oplus 7 \otimes x_3(k-1) \oplus 14 \otimes u(k) \\ y(k) &= 7 \otimes x_3(k) \end{aligned} \tag{2-22}$$

We can rewrite this into the MPL state space description as defined in Definition 2.8:

$$\begin{aligned} x(k) &= \begin{bmatrix} 11 & \varepsilon & \varepsilon \\ \varepsilon & 12 & \varepsilon \\ 23 & 24 & 7 \end{bmatrix} \otimes x(k-1) \oplus \begin{bmatrix} 2 \\ e \\ 14 \end{bmatrix} \otimes u(k) \\ y(k) &= \begin{bmatrix} \varepsilon & \varepsilon & 7 \end{bmatrix} \otimes x(k) \end{aligned} \tag{2-23}$$

The state space model in Definition 2.8 is an *explicit model*; the state vector $x(k)$ is only depending on past state values $x(k-1)$. However, we have seen in subsection 2-4-2 that there also exist systems written in an *implicit form*. We can generalize the MPL state space description defined in Definition 2.8 as an implicit system. The state vector of such a system is represented as [13]:

$$x(k) = \bigoplus_{i=0}^{M} (A_i \otimes x(k-i)) \oplus B \otimes u(k), \tag{2-24}$$

with $M \geq 0$ the recurrence order, representing the number of past cycles that are contained in the state vector. It can be seen that the state vector $x(k)$ now not only can depend on previous cycles such as $x(k-1)$, but it also depends on $x(k)$, the current cycle. When applying control to such implicit systems computational difficulties can occur, as you have to iterate multiple times in order to determine the event times $x(k)$ correctly [22]. However, if we approach this implicit model form similarly as the system of MPL equations presented in Lemma 2.1, we can rewrite the state vector into the explicit model as in Equation 2-20. We will follow the steps of [13, Paragraph 4.5], which makes use of the Kleene star:

We start with the definition of A^+, as given in Definition 2.4 and [13, Lemma 2.2], which states that if the communication graph $\mathcal{G}(A)$ of $A \in \mathbb{R}_\varepsilon^{n\times n}$ has maximal average circuit weight $\leq e$, it holds that:

$$A^+ = A \oplus A^{\otimes 2} \oplus \ldots \oplus A^{\otimes n} \in \mathbb{R}_\varepsilon^{n\times n}, \tag{2-25}$$

Which implies that then:

$$A_0^* = \bigoplus_{i=0}^{n-1} A_0^{\otimes i}. \tag{2-26}$$

We now reduce Equation 2-24 to:

$$\begin{aligned} x(k) &= \bigoplus_{i=0}^{M} (A_i \otimes x(k-i)) \oplus B \otimes u(k) \\ &= A_0 \otimes x(k) \oplus \underbrace{\bigoplus_{i=1}^{M} (A_i \otimes x(k-i)) \oplus B \otimes u(k)}_{b(k)} \\ &= A_0 \otimes x(k) \oplus b(k). \end{aligned} \tag{2-27}$$

Now using Lemma 2.1, we can say that:

$$x(k) = A_0^* \otimes b(k) = A_0^* \otimes \bigoplus_{i=1}^{M} (A_i \otimes x(k-i)) \oplus B \otimes u(k). \tag{2-28}$$

Clearly, this equation does not contain the current cycle $x(k)$ on the right hand side, meaning that we transformed the implicit model Equation 2-24 into an explicit model.

2-5 Hybrid Systems

In this section we discuss the general definitions of three hybrid system model structures, namely Max-Min-Plus-Scaling (MMPS), Piecewise Affine (PWA) and Mixed Logical Dynamical (MLD) systems, as defined in [14]. As stated in the introduction of this chapter, these classes will be relevant for the chapters hereafter. Hybrid system theory covers modeling frameworks for systems that contain both continuous and discrete dynamics. The model techniques can furthermore be implemented on systems that are time as well as event driven. As max-plus algebra is introduced as a tool to model event-driven systems, we will see some remarkable equivalences between the max-plus system structures and the hybrid system structures later on. Each hybrid model structure has its own advantages and can be implemented depending on the characteristics of the concerning system. After we introduced the three modeling frameworks, we will show that the frameworks are equivalent to one another under some mild assumptions. We will finally present optimization strategies that are useful when optimizing the concerning hybrid systems.

2-5-1 Max-Min-Plus-Scaling Systems

In this subsection, we introduce the class of MMPS systems based on [14, paragraph 2.5]. MMPS systems are of great use as its model can be considered as a generalized framework covering many subclasses of DES and hybrid systems. After we have stated the formal definitions, it will be clear that MPL systems as defined in Definition 2.8 are actually a subclass of MMPS systems. The MMPS framework is therefore able to model many different applications. We define an MMPS function as:

Definition 2.9 (Max-Min-Plus Scaling Function)**.** *An* MMPS function $\mathbb{R}^n \to \mathbb{R}^m$ *is given by the following* recursive *definition:*

$$f : x_i|\alpha|\max(f_k, f_l)|\min(f_k, f_l)|f_k + f_l|\beta f_k, \tag{2-29}$$

for $i = 1, ..., n$, $\alpha, \beta \in \mathbb{R}$ scalars, $f_k : \mathbb{R}^n \to \mathbb{R}^m$ and $f_l : \mathbb{R}^n \to \mathbb{R}^m$ again MMPS functions. The operation $|$ stands for "or" here, and the max *and* min *operations are executed entry-wise.*

As can be seen in the definition and as the name states, an MMPS function is only allowed to use the operations maximization, minimization, addition and multiplication (scaling). It can therefore be seen as a (less strict) extension on max-plus algebra, in which strictly maximization and addition is allowed. We now define an MMPS system as:

Definition 2.10 (Max-Min-Plus Scaling System)**.** *A* constrained MMPS system *is defined by the following state-space model:*

$$\begin{aligned} x(k) &= \mathcal{M}_x(x(k-1), u(k)) \\ y(k) &= \mathcal{M}_y(x(k), u(k)), \end{aligned} \tag{2-30}$$

with the constraint:

$$\mathcal{M}_c((x(k), u(k)) \leq c \tag{2-31}$$

with $\mathcal{M}_x, \mathcal{M}_y$ and $\mathcal{M}_c$ MMPS functions, $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $y(k) \in \mathbb{R}^l$ the state, input and output vector and k an event counter. If the constraint in Equation 2-31 is absent, we have an unconstrained MMPS system.

Remark 1. In this thesis, we will only consider MMPS systems that are structurally finite, which ensures that the system remains bounded for bounded initial states (and thus cannot become ε). This is a safe assumption as physical (real) systems are generally structurally finite [8].

One can observe that MPL systems indeed are a (strict) subclass of the framework defined in Equation 2-30; MPL systems are only allowed to use the operations maximization and addition, which are included in the MMPS operations in Definition 2.9.

2-5-2 Piecewise Affine Systems

In this subsection we introduce the class of PWA systems, based on [14, paragraph 2.1]. PWA systems can be considered as the 'simplest' extension of linear systems that can model hybrid as well as nonlinear dynamics with arbitrary accuracy. PWA systems are very popular, since a rich class of hybrid systems can be described in the PWA framework and the framework is of great use for stability analysis. Before we can formally define PWA systems, it is necessary to introduce the following concepts:

Definition 2.11 (Convex Set)**.** *A subset $\mathcal{C} \subset \mathbb{R}^n$ is a* convex set *if the following holds:*

$$\mathcal{C} = \{c \mid c = \alpha x + (1-\alpha)y,\ \forall x, y \in \mathcal{C},\ \forall \alpha \in [0,1]\} \tag{2-32}$$

Furthermore we have:

Definition 2.12 (Halfspace)**.** *A set (closed and convex) $\mathcal{V} \subset \mathbb{R}$ is a* closed halfspace *if it is of the form:*

$$\mathcal{V} = \{x \mid a^T x \leq b\} \tag{2-33}$$

with nonzero vector a and scalar b. In other words: a closed halfspace is a linear inequality. A halfspace is considered open if the inequality in Equation 2-33 is a strict inequality with $<$.

We can now introduce a *polyhedral set*:

Definition 2.13 (Polyhedral Set)**.** *A nonempty set $\mathcal{X} \subset \mathbb{R}$ is a* polyhedral set *if it is constructed by the intersection of a finite number of halfspaces:*

$$\mathcal{X} = \{x \mid a_i^T x \leq b_i,\ i = 1, \ldots, n\} \tag{2-34}$$

with $a_1, \ldots, a_n$ vectors and $b_1, \ldots, b_n$ scalars. In other words, a set of linear inequalities.

And finally we can give the definition of a *polytope*:

Definition 2.14 (Polytope)**.** *A* polytope *is a bounded polyhedral set.*

We can now state the formal definition of a continuous PWA function:

Definition 2.15 (Continuous Piecewise Affine Function)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a scalar-valued function. This function is a* continuous PWA function *if and only if the following conditions hold:*

1. *The domain space $\mathbb{R}^n$ is divided into a finite number of polyhedral regions $R_1, \ldots, R_N$, with a polyhedral region defined as in Definition 2.13.*
2. *For each $i \in \{1, \ldots, N\}$, the function f can be expressed as an affine function $f(x) = \alpha_i^T x + \beta_i$, $\forall x \in R_i$ with vector $\alpha_i \in \mathbb{R}^n$ and scalar $\beta_i \in \mathbb{R}$.*
3. *The function f is continuous on any boundary between two regions.*

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

And finally, we can introduce (continuous) PWA systems:

Definition 2.16 ((Continuous) Piecewise Affine System)**.** Piecewise Affine Systems *are formulated as:*

$$\begin{aligned} x(k) &= \mathcal{P}_x(x(k-1), u(k)) \\ y(k) &= \mathcal{P}_y(x(k), u(k)) \end{aligned} \tag{2-35}$$

In which $\mathcal{P}_x$ and $\mathcal{P}_y$ are vector-valued PWA functions. Or equivalently, in a more general state space description:

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i, \\ y(k) &= C_i x(k) + D_i u(k) + g_i, \end{aligned} \tag{2-36}$$

for $\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i, i = 1, ..., n,$ *and Ω_i convex polyhedra in input/state space defined as in Definition 2.13. The vectors $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $y(k) \in \mathbb{R}^l$ represent the state, input and output respectively, with k representing the time. Furthermore the system matrices are defined as $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times m}$, $C_i \in \mathbb{R}^{l \times n}$ and $D_i \in \mathbb{R}^{l \times m}$. The values $f_i \in \mathbb{R}^l$ and $g_i \in \mathbb{R}^n$ represent scalar offsets $\forall i = 1, ..., N$, making the state and output equations affine. If $\mathcal{P}_x$ and $\mathcal{P}_y$ are continuous, or if the affine equations in Equation 2-36 are continuous, we speak of a* continuous PWA system.

2-5-3 Mixed Logical Dynamical Systems

The third and final hybrid system class we introduce, is the class of MLD systems based on [14, paragraph 2.2]. This class provides a framework to present hybrid systems described by linear dynamic equations subject to linear inequalities involving real and integer variables. Many applications include parts described by logic, such as on and off switches. Such parts will be presented as integer or binary variables. Before we introduce the MLD framework, we introduce some necessary tools to transform logical statements involving continuous variables into mixed-integer linear inequalities with the use of logical variable $\delta \in \{0, 1\}$ and linear function $f : \mathbb{R}^n \to \mathbb{R}$. Let $\mathcal{X}$ be a bounded with $x \in \mathcal{X}$ such that:

$$M = \max_{x \in \mathcal{X}} f(x) \qquad m = \min_{x \in \mathcal{X}} f(x) \tag{2-37}$$

We can define following statements concerning δ and f:

$$\begin{aligned} &[f(x) \leq 0] \wedge [\delta = 1] \text{ true iff } f(x) - \delta \leq -1 + m(1 - \delta) \\ &[f(x) \leq 0] \vee [\delta = 1] \text{ true iff } f(x) \leq M\delta \\ &\sim [f(x) \leq 0] \text{ true iff } f(x) \geq \epsilon \qquad \text{(with } \epsilon \text{ a machine error)} \\ &[f(x) \leq 0] \Rightarrow [\delta = 1] \text{ true iff } f(x) \geq \epsilon + (m - \epsilon)\delta \\ &[f(x) \leq 0] \Leftrightarrow [\delta = 1] \text{ true iff } \begin{cases} f(x) \leq M(1 - \delta) \\ f(x) \geq \epsilon + (m - \epsilon)\delta \end{cases} \end{aligned} \tag{2-38}$$

Now let us define $z = \delta f(x)$, which is equivalent to the following linear inequalities:

$$[z = \delta f(x)] \Leftrightarrow \begin{cases} z \leq M\delta \\ z \geq m\delta \\ z \leq f(x) - m(1 - \delta) \\ z \geq f(x) - M(1 - \delta) \end{cases} \tag{2-39}$$

We can now formally introduce the MLD framework:

Definition 2.17 (Mixed Logical Dynamic System)**.** *A MLD system is defined by the following state-space model:*

$$\begin{aligned} x(k+1) &= Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \\ y(k) &= Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \end{aligned} \tag{2-40}$$

Subject to the constraint:

$$E_1x(k) + E_2u(k) + E_3\delta(k) + E_4z(k) \leq g \tag{2-41}$$

With state vector $x(k) \in \begin{bmatrix} x_r^\top(k) & x_b^\top(k) \end{bmatrix}^\top$ *with* $x_r(k) \in \mathbb{R}^{n_r}$ *real states,* $x_b(k) \in \{0,1\}^{n_b}$ *binary states and* $n = n_r + n_b$*. Output* $y(k) \in \mathbb{R}^{n_y}$ *and input* $u(k) \in \mathbb{R}^{n_u}$ *are defined similarly, and* $z(k) \in \mathbb{R}^{n_r}$ *and* $\delta(k) \in \{0,1\}^{n_b}$ *auxiliary variables. The system matrices are defined as* $A \in \mathbb{R}^{n\times n}$*,* $B_1 \in \mathbb{R}^{n\times n_u}$*,* $B_2 \in \mathbb{R}^{n\times n_b}$*,* $B_3 \in \mathbb{R}^{n\times n_r}$*,* $C \in \mathbb{R}^{n_y\times n}$*,* $D_1 \in \mathbb{R}^{n_y\times n_u}$*,* $D_2 \in \mathbb{R}^{n_y\times n_b}$ *and* $D_3 \in \mathbb{R}^{n_y\times n_r}$*. Furthermore we have* $E_1 \in \mathbb{R}^{n_g\times n}$*,* $E_2 \in \mathbb{R}^{n_g\times n_u}$*,* $E_3 \in \mathbb{R}^{n_g\times n_b}$*,* $E_4 \in \mathbb{R}^{n_g\times n_r}$ *and vector* $g \in \mathbb{R}^{n_g}$*.*

2-5-4 Class Equivalences

In this subsection, we will present the equivalences between the presented hybrid system classes based on [14, paragraph 2.7]. Some equivalences go along with mild assumptions. Figure 2-3 gives a graphical representation of the equivalences, in which the $*$ notations indicate that an assumption is necessary. Let us present these equivalences by means of the propositions stated in the figure.

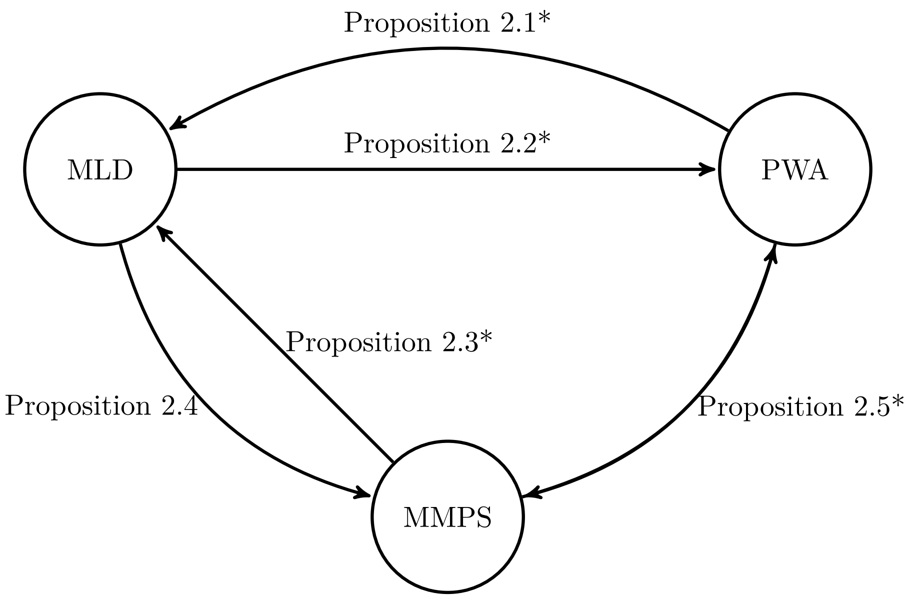


Figure 2-3: Graphical representation of the equivalences between the three hybrid system classes presented in the previous subsections. The $*$ notation indicates that a (mild) assumption is necessary for the concerning equivalence.

Proposition 2.1. *A PWA system can be written intro MLD format, provided that the PWA system is well-posed. A PWA system is considered well-posed if, given* $x(k)$ *and* $u(k)$*, a unique solution* $x(k+1)$ *and* $y(k)$ *exists such that the dynamics of the state space description hold.*

Proposition 2.2. *An MLD system can be written into an PWA system, given that the MLD system is completely well-posed. An MLD system is completely well-posed, if $x(k+1)$, $y(k)$, $\delta(k)$ and $z(k)$ are uniquely defined in their domain, given $x(k)$ and $u(k)$.*

Proposition 2.3. *MMPS systems can be written as an MLD system, given that $g - E_1x(k) - E_2u(k) - E_3\delta(k) - E_4z(k)$ is bounded component wise.*

Proposition 2.4. *MLD systems can be written into MMPS format.*

Proposition 2.5. *PWA and MMPS systems are equivalent, provided that the PWA system is continuous.*

Details of the proofs of Proposition 2.5 until Proposition 2.3 can be found in [14, paragraph 2.7].

2-6 Case Study I: An Urban Railway System as Max-Plus Linear System

In this section, we introduce an Urban Railway System (URS) as an MPL system. This URS will be the running case study and will be expanded in every chapter throughout the thesis. A similar description for a URS was firstly introduced by [19], and thereafter adopted in [10] and [8]. The URS considered in this section will be a highly simplified version of a real life URS such as metro systems or tram systems, in order to fit it into the MPL system framework.

2-6-1 System Description and Assumptions of the Urban Railway System

To fit the URS into the MPL framework, we will consider only two stations:

Assumption 2.1. The route of the URS consists out of two stations.

Note that the system can easily be expanded to more stations, but for simplicity and readability of the expressions, we only consider such a small system. Figure 2-4 visualizes the railway route that is considered.

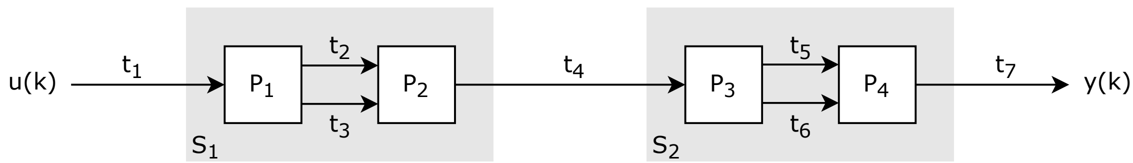


Figure 2-4: Case Study: two-station URS as an MPL system.

Station 1 is represented as S_1 and station 2 equivalently as S_2. P_i with $i = 1, ..., 4$ represent the execution of events which can be defined in the following manner:

- $P_1 = a_1(k)$: arrival time of the k-th train at station 1;

- $P_2 = d_1(k)$: departure time of the k-th train from station 1;
- $P_3 = a_2(k)$: arrival time of the k-th train at station 2;
- $P_4 = d_2(k)$: departure time of the k-th train from station 2.

The time units t_1, t_2, ..., t_7 represent the time it takes for a train to go from one event to another. Let us specify these time units:

- t_1: number of time units it takes for a new train to enter the route, or the *entering running time*;
- t_2: number of time units a train must wait on the platform of S_1 before it can depart, or the *minimum dwell time* S_1;
- $t_3 = \tau_{\text{dwell},1}$: number of time units it takes for passengers to disembark and board, or *actual dwell time* S_1. The actual dwell time at S_1 can vary per train, and is therefore not specified exactly;
- t_4: number of time units it takes for a train to travel from S_1 to S_2, or the *running time*;
- t_5: number of time units a train must wait on the platform of S_2 before it can depart, or the *minimum dwell time* S_2;
- $t_6 = \tau_{\text{dwell},2}$: number of time units it takes for passengers to disembark and board, or *actual dwell time* S_2. The actual dwell time at S_2 can vary per train, and is therefore not specified exactly;
- t_7: number of time units it takes for a train to leave the route, or the *leaving running time.*

Let us furthermore define input $u(k)$ as the time instant a k-th train is 'fed' into the two-station route. Finally we define output $y(k)$ the time instant a k-th train completes and leaves the train route. We will consider the following assumptions for the two-station URS route:

Assumption 2.2. The URS is assumed to be uni-directional.

Assumption 2.3. The URS operates without a timetable which is usually the realistic case. Trains should come at a high frequency (every few minutes), but will not be scheduled to arrive or depart on the minute precise.

Assumption 2.4. Each train is assumed to have an infinite capacity, in order to prevent any situation in which the train is full and passengers cannot board.

Assumption 2.5. Each train leaves as soon as possible, resulting in the highest frequency possible.

Assumption 2.6. Trains cannot overtake one another.

Assumption 2.7. Each station only has one platform, ensuring that there can only be one train present at the time.

Assumption 2.8. For safety, a headway time τ_H is included, representing the fixed distance in time units required between consecutive trains.

Assumption 2.9. The actual dwell times $\tau_{\text{dwell},i}$ with $i = 1, 2$ are assumed to be measurable.

With the above assumptions, we are ready to obtain the state-space description of the simplified URS.

2-6-2 MPL State-Space Description of the Urban Railway System

Let us assign values for the time units $t_1, t_2, ..., t_7$ and t_H:

Time unit	Definition	Value [min]
t_1	Entering running time	18
t_2	Minimum dwell time S_1	3
t_3	Actual dwell time S_1	measurable
t_4	Running time	20
t_5	Minimum dwell time S_2	4
t_6	Actual dwell time S_2	measurable
t_7	Leaving running time	16
t_H	Headway time	2

Table 2-1: Time unit values in minutes for the two-station URS as presented in Figure 2-4.

If we collect the events P_i with $i = 1, ..., 4$ in a state vector, we obtain:

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} = \begin{bmatrix} a_1(k) \\ d_1(k) \\ a_2(k) \\ d_2(k) \end{bmatrix} \tag{2-42}$$

Given the assumptions, we can state the following inequalities for $x_1(k)$:

$$\begin{aligned} x_1(k) &\leq x_2(k-1) + \tau_H \\ x_1(k) &\leq u(k) + t_1 \end{aligned} \tag{2-43}$$

Or equivalently, we can write:

$$x_1(k) \leq \max(x_2(k-1) + \tau_h, u(k) + t_1) \tag{2-44}$$

From the fourth assumption that each train leaves as soon as possible, we can replace the inequality sign with an equality sign, resulting in:

$$x_1(k) = \max(x_2(k-1) + \tau_h, u(k) + t_1) \tag{2-45}$$

Similarly, we can obtain the following for the departure time of train k from S_1:

$$\begin{aligned} x_2(k) &\leq x_1(k) + t_2 \\ x_2(k) &\leq x_1(k) + t_3 \end{aligned} \tag{2-46}$$

And thus we have:

$$x_2(k) \leq \max(x_1(k) + t_2, x_1(k) + t_3) \tag{2-47}$$

And again by the fourth assumption we have:

$$x_2(k) = \max(x_1(k) + t_2, x_1(k) + t_3) \tag{2-48}$$

Following the same steps for $x_3(k)$ and $x_4(k)$ results in the following:

$$\begin{aligned} x_3(k) &= \max(x_2(k) + t_4, x_4(k-1) + \tau_H) \\ x_4(k) &= \max(x_3(k) + t_5, x_3(k) + t_6) \end{aligned} \tag{2-49}$$

If we combine the above expressions for the states, we can write:

$$\begin{aligned} x(k) &= \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ t_2 \otimes t_3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & t_4 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_5 \otimes t_6 & \varepsilon \end{bmatrix} \otimes x(k) \\ &\quad \oplus \begin{bmatrix} \varepsilon & \tau_H & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \tau_H \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \otimes x(k-1) \oplus \begin{bmatrix} t_1 \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix} \otimes u(k) \\ y(k) &= \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & t_7 \end{bmatrix} \otimes x(k) \end{aligned} \tag{2-50}$$

And with the fixed numerical values as defined in Table 2-1, we can write:

$$\begin{aligned} x(k) &= \underbrace{\begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 3 \otimes \tau_{\text{dwell},1} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 20 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 4 \otimes \tau_{\text{dwell},2} & \varepsilon \end{bmatrix}}_{A_0} \otimes x(k) \\ &\quad \oplus \underbrace{\begin{bmatrix} \varepsilon & 2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}}_{A_1} \otimes x(k-1) \oplus \underbrace{\begin{bmatrix} 18 \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix}}_{B} \otimes u(k) \\ y(k) &= \underbrace{\begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & 16 \end{bmatrix}}_{C} \otimes x(k) \end{aligned} \tag{2-51}$$

For simplicity, let us write:

$$\begin{aligned} x(k) &= A_0 \otimes x(k) \oplus A_1 \otimes x(k-1) \oplus B \otimes u(k) \\ y(k) &= C \otimes x(k) \end{aligned} \tag{2-52}$$

We can recognize the general implicit form of an MPL system as earlier defined in Equation 2-24. Note that we are dealing with implicitness as the A_0 matrix is not empty. Let us investigate if we can resolve this implicitness using the Kleene star. We have seen that if the maximal average circuit weight is non-positive, we can obtain the Kleene star A_0^*. However, for the two-station URS, the A_0 matrix is constructed in such a manner that it does not contain any circuits, and we therefore have that the maximal average circuit weight is equal to ε and is therefore non-positive. We can therefore conclude that the Kleene star exists, and we can rewrite Equation 2-52 in the following manner:

$$\begin{aligned} x(k) &= A_0^* \otimes A_1 \otimes x(k-1) \oplus B \otimes u(k) \\ y(k) &= C \otimes x(k) \end{aligned} \tag{2-53}$$

In which A_0^* is the Kleene star of matrix A_0, which can be obtained in the following manner using Definition 2.3:

$$\begin{aligned} A_0^* &= \bigoplus_{k\geq 0} A_0^{\otimes k} \\ &= E \oplus A_0 \oplus A_0^{\otimes 2} \oplus A_0^{\otimes 3} \oplus \dots \end{aligned} \tag{2-54}$$

With:

$$\begin{aligned} A_0^{\otimes 1} &= A_0 \\ A_0^{\otimes 2} &= A_0 \otimes A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 23 \otimes \tau_{\text{dwell},1} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 24 \otimes \tau_{\text{dwell},2} & \varepsilon \end{bmatrix} \\ A_0^{\otimes 3} &= A_0 \otimes A_0 \otimes A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 26 \otimes \tau_{\text{dwell},1} \otimes \tau_{\text{dwell},1} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 44 \otimes \tau_{\text{dwell},2} & \varepsilon & \varepsilon \end{bmatrix} \\ A_0^{\otimes 4} &= A_0 \otimes A_0 \otimes A_0 \otimes A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 47 \otimes \tau_{\text{dwell},1} \otimes \tau_{\text{dwell},2} & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \\ A_0^{\otimes 5} &= A_0 \otimes A_0 \otimes A_0 \otimes A_0 \otimes A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \end{aligned} \tag{2-55}$$

Clearly, for $A_0^{\otimes k}$ with $k \geq 5$, we have $A_0^{\otimes k} = \mathcal{E}$. And we therefore have:

$$A_0^* = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ 3 \otimes \tau_{\text{dwell},1} & e & \varepsilon & \varepsilon \\ 26 \otimes \tau_{\text{dwell},1} \otimes \tau_{\text{dwell},1} & 23 \otimes \tau_{\text{dwell},1} & e & \varepsilon \\ 47 \otimes \tau_{\text{dwell},1} \otimes \tau_{\text{dwell},2} & 44 \otimes \tau_{\text{dwell},2} & 24 \otimes \tau_{\text{dwell},2} & e \end{bmatrix} \tag{2-56}$$

Chapter 3

Linear Parameter Varying Systems

This chapter is devoted to the Linear Parameter Varying (LPV) framework. This framework is useful in conventional algebra, as it offers a theoretical approach to model nonlinear systems. Recently, this framework is extended to its max-plus analogue and is therefore promising for system modeling and control in the field of max-plus algebra. Today's control field is challenged as the need to optimize efficiency and improve reliability of automatization is increasing. To cope with these challenges and meet the objectives, multiple control approaches have been introduced, such as robust and nonlinear control. However, in order to implement these control approaches, an accurate and reliable mathematical description of the dynamics of the physical phenomena is necessary. Especially for physical and chemical processes in which nonlinear behavior or dependence on external variables often occurs, modeling these dynamics can be challenging. Take for example the upcoming field of sustainable energy; when generating energy from sources such as wind, waves and tides, one deals with highly nonlinear dynamics [23]. Based on the first principle laws of for example physics, one can construct models that capture these dynamics. This procedure is however time consuming and only experts of that field can obtain a reliable model. These challenges give rise to the introduction of the LPV framework and the extension to the Max-Plus Linear Parameter Varying (MP-LPV) framework. With the use of these frameworks, we are able to capture the nonlinear system dynamics in a parameterized (max-plus) linear structure.
This chapter is organized as follows. We will first present the conventional LPV framework in section 3-1, which we will extend to the MP-LPV framework in section 3-2 thereafter. In section 3-3 we will extend the Urban Railway System (URS) presented in section 2-6 as a first MP-LPV system, and finally in section 3-4, we present the URS as an Max-Min-Plus-Scaling (MMPS) system.

3-1 Linear Parameter Varying Systems

In this section, we introduce the LPV structure for conventional systems. This framework provides the possibility to model nonlinear dynamics in a parameterized linear system structure.

The LPV framework is firstly introduced in [24] as a result of investigating gain-scheduling control approaches. Gain-scheduling is a commonly used nonlinear control method that linearizes the nonlinear system model around different "frozen" operating points, resulting in a set of local Linear Time Invariant (LTI) systems. The set of operating points is chosen such that the range of the system's dynamics is covered. For each of the resulting LTI systems, a local linear controller is designed. The gains of the local controllers at each operating point are interpolated or *scheduled*, resulting in a global nonlinear controller. There are however some important drawbacks in the gain-scheduling approach [25]. One of these drawbacks is that this method only makes a discrete selection of operating points for which a local controller is designed. As a result, performance properties such as stability or robustness can only be guaranteed at those operating points. Thus, nothing can be guaranteed globally a priori. These drawbacks were resolved by the introduction of the LPV framework, as it provides a method to contain all (nonlinear) dynamics in one model presenting a complete overview of the system. The framework therefore gives rise to the possibility to design global gain-scheduling based controllers instead of local controllers.
We will describe the LPV framework following [9]; the system dynamics are captured in a linear state-space model, in which the system matrices depend on an exogenous *scheduling variable* p that describes the changes of the operating point. This time-varying scheduling parameter p can itself be nonlinear, but the signal relations of the system remain linear. A simple example for a scheduling parameter p is $p(k) = x(k-1) + x(k)$; the presence of $x(k)$ in $p(k)$ makes the system matrices state-dependent, resulting in a nonlinear system. The system is however still linear ín the parameter, due to the structure. It is assumed that the time variation of the parameter p is not (necessarily) known, but that it is measurable in real time. The result is the LPV system class, that can describe both time-varying and nonlinear phenomena because of this varying parameter dependence. The difference with the classical gain-scheduling approach is therefore that instead of obtaining multiple controllers for a set of frozen operation points, one global controller is designed for the complete range of system dynamics. The LPV framework is therefore interesting for control purposes, and it can be seen as the "middle ground" between LTI systems and nonlinear or time-varying systems [26]. The LPV framework is described by the following discrete-time state-space model [27] in conventional algebra:

Definition 3.1 (General LPV System)**.**

$$\begin{aligned} x(k+1) &= A(p(k))x(k) + B(p(k))u(k), \\ y(k) &= C(p(k))x(k) + D(p(k))u(k) \end{aligned} \tag{3-1}$$

In which $u(k) \in \mathbb{R}^m$ is the input, $y(k) \in \mathbb{R}^n$ the output, $x(k) \in \mathbb{R}^n$ the state vector and the system matrices $A(\cdot) \in \mathbb{R}^{n \times n}$, $B(\cdot) \in \mathbb{R}^{n \times m}$, $C(\cdot) \in \mathbb{R}^{l \times n}$ and $D(\cdot) \in \mathbb{R}^{l \times m}$ depend on the exogenous parameter vector $p(k) \in P \subset \mathbb{R}^r$ with P the set of parameter values. Furthermore the index k is considered to be a time variable.

We can define the set of parameter values P as the following:

Definition 3.2 (Parameter Set)**.** *Let P be a closed, bounded set in $\mathbb{R}^r$. The system matrices of the LPV system depend on parameter p in P^r, with r representing the size of p.*

It remains to show that nonlinear systems actually can be recasted into the LPV framework. There exist many approaches for this LPV identification process, of which multiple can be found in [28], and furthermore in [29] and [30]. We can furthermore find multiple applications that use the LPV framework; [31] uses the LPV framework for control of anesthesia delivery during surgeries, [32] uses LPV for preserving vehicle stability in extreme situations and [33] uses it for the control of wind energy conversion systems in wind turbines.

3-2 Max-Plus Linear Parameter Varying Systems

In this section the conventional LPV framework is extended to its max-plus analogue, resulting in MP-LPV systems. The notion of MP-LPV systems is firstly introduced in [7] as an extended description on Max-Plus-Linear (MPL) systems. In the theses [19] and [10] elaborate research is done on MP-LPV systems and therefore form the foundation of this thesis. Research in MP-LPV systems is very limited; the named resources are one of the few that have researched this type of systems. We define a general model for an MP-LPV system in the following manner:

Definition 3.3. [General MP-LPV System]

$$\begin{aligned} x(k) &= A(p(k)) \otimes x(k-1) \oplus B(p(k)) \otimes u(k) \\ y(k) &= C(p(k)) \otimes x(k), \end{aligned} \tag{3-2}$$

With $\{A(p(k)), B(p(k)), C(p(k))\}$ the parameter dependent system matrices in $\mathbb{R}_\varepsilon^{n\times n}$, $\mathbb{R}_\varepsilon^{n\times m}$ and $\mathbb{R}_\varepsilon^{l\times n}$ respectively, the state vector $x(k) \in \mathbb{R}_\varepsilon^n$, the input vector $u(k) \in \mathbb{R}_\varepsilon^m$, the output vector $y(k) \in \mathbb{R}_\varepsilon^l$ and exogenous parameter vector $p(k) \in P_\varepsilon \in \mathbb{R}_\varepsilon^r$.

The index k is now considered to be an event counter instead of a time variable as in conventional LPV systems. Therefore equivalently as in MPL systems, the state $x(k)$ and output $y(k)$ represent the k-th occurrence of events x and y, and $u(k)$ the k-th time that input u is implemented. Let us formulate the definition of the parameter set P_ε, now in the max-plus sense:

Definition 3.4 (Max-Plus Algebraic Parameter Set)**.** *Let P_ε be a closed, bounded set in $\mathbb{R}_\varepsilon^r$. The system matrices of the MP-LPV system depend on parameter p in P_ε^r, with r representing the size of p.*

3-2-1 Canonical Form of State-Space Description

The general state-space description of an MP-LPV system defined in Definition 3.3 can be rewritten into a standard implicit form, similarly as for MPL systems in Equation 2-24. This description is given by:

$$\begin{aligned} x(k) &= \bigoplus_{\mu=0}^{M} \left(A_\mu(p(\cdot)) \otimes x(k-\mu)\right) \oplus B(p(\cdot)) \otimes u(k) \\ y(k) &= C(p(\cdot)) \otimes x(k) \end{aligned} \tag{3-3}$$

With the parameter vector $p(\cdot)$ defined as:

$$p(k, x(k), u(k), z(k)) = [x^T(k), x^T(k-1), \ldots, x^T(k-M), u^T(k), z^T(k)] \in P \tag{3-4}$$

$M \geq 1$ is the recurrence order, representing the number of past cycles that are contained in the state vector, $z(k) \in \mathbb{R}^{n_z}$ is an exogenous input that is assumed to be independent of the state and the system vectors and matrices defined as in Definition 3.3. This general implicit form can be rewritten into a canonical form, which will be useful for studying system characteristics such as solvability. However, before we can translate the MP-LPV description, the system first needs to meet the following assumption, presented in [8]:

Assumption 3.1. *The finite entries of the system matrices $A_\mu(\cdot)$ with $\mu = \{1, ..., M\}$, $B(\cdot)$ and $C(\cdot)$ in Equation 3-3 are continuous piecewise affine in the parameter $p(\cdot) \in P$ defined as in Equation 3-4. We assume furthermore that the system description in Equation 3-3 is structurally finite. The latter means that the following matrix is row finite:*

$$F(\cdot) = \begin{bmatrix} A_0(\cdot) & A_1(\cdot) & \dots & A_M(\cdot) & B(\cdot) & \mathcal{E} \\ \mathcal{E} & \mathcal{E} & \dots & \mathcal{E} & \mathcal{E} & C(\cdot) \end{bmatrix} \tag{3-5}$$

Mathematically this means that F is row finite[1]. *The structure of matrix $F(\cdot)$ is assumed to be independent of the parameter $p(\cdot)$.*

If Assumption 3.1 holds for the MP-LPV system in Equation 3-3, we can rewrite that description into the following canonical form:

Lemma 3.1. *Consider the implicit MP-LPV system as defined in Equation 3-3. Under Assumption 3.1 the dynamics of the MP-LPV system can be rewritten into the following canonical form:*

$$\begin{aligned} x(k) &= \bigoplus_{\mu=0}^{M} \Big(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu) \Big) \oplus B(p^{(1)}(k)) \otimes u(k) \\ y(k) &= C(p(k)) \otimes x(k) \\ p^{(\mu)}(k) &= [x^T(k-\mu), \dots, x^T(k-M), u^T(k), z^T(k)]^T \in P^{(\mu)} \end{aligned} \tag{3-6}$$

The proof for the above lemma can be found in [8]. This internal document introduces furthermore that under Assumption 3.1, MP-LPV systems and MMPS systems are equivalent. Let us formally state the latter theorem:

Theorem 3.1. *Under Assumption 3.1, the class of MMPS systems as defined in Definition 2.10 and the class of MP-LPV systems as defined in Lemma 3.1 coincide.*

For details of the proof of this theorem we again refer to [8]. We can therefore update Figure 2-3 to Figure 3-1. The propositions referred to in this figure can be found in subsection 2-5-4.

[1] A matrix is row finite if every row has at least one finite entry.

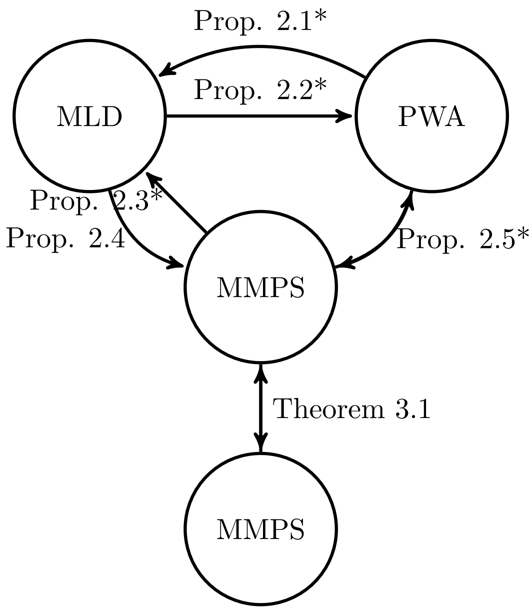


Figure 3-1: Updated graphical representation of the equivalences between the hybrid system classes presented in subsection 2-5-4. The $*$ notation indicates that a (mild) assumption is necessary for the concerning equivalence. Proposition 2.1 until 2.5 can be found in subsection 2-5-4.

3-2-2 Levels of Implicitness

The canonical representation of the MP-LPV system is given in an implicit format. We can however identify different levels of implicitness by making small adjustments in the representation. We will extract systems in four levels of implicitness, namely explicit systems, single implicit systems version 1, single implicit systems version 2 and doubly implicit systems.

Explicit MP-LPV System

An *explicit* system occurs when having explicit signal relations and furthermore an explicit parameter set. One can observe that such a system occurs when $A_0(\cdot) = \mathcal{E}_{n\times n}$, because then $A_0(p^{(0)}(k)) \otimes x(k) = \mathcal{E}_{n\times n} \otimes x(k) = \mathcal{E}_{n\times n}$. Let us write the following for such an explicit system:

$$\begin{aligned}
x(k) &= \bigoplus_{\mu=0}^{M} \Big(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\Big) \oplus B(p^{(1)}(k)) \otimes u(k) \\
&= A_0(p^{(0)}(k)) \otimes x(k) \oplus \bigoplus_{\mu=1}^{M} \Big(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\Big) \oplus B(p^{(1)}(k)) \otimes u(k) \\
&= \mathcal{E}_{n\times n} \oplus \bigoplus_{\mu=1}^{M} \Big(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\Big) \oplus B(p^{(1)}(k)) \otimes u(k) \\
&= \bigoplus_{\mu=1}^{M} \Big(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\Big) \oplus B(p^{(1)}(k)) \otimes u(k)
\end{aligned} \tag{3-7}$$

We can see that the state $x(k)$ only depends on $A_1(p^{(1)}), ..., A_M(p^{(M)})$ and $x(k-1), ..., x(k-M)$. The resulting state equation is therefore explicit, as it only depends on previous state values.

Single Implicit MP-LPV System - Version 1

We define version 1 of a *single implicit* system if we combine implicit signal relations with an explicit parameter vector. Mathematically this happens when $A_0(p(\cdot)) = A_0(p^{(\eta)}(\cdot))$ for $\eta \geq 1$. We can then write the state equation in the following manner:

$$x(k) = A_0\left(p^{(1)}(k)\right) \otimes x(k) \oplus \bigoplus_{\mu=1}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) \oplus B(p^{(1)}(k)) \otimes u(k) \tag{3-8}$$

The implicitness is thus only due to the presence of $x(k)$ on both the left and right side. This level of implicitness is similar to the implicitness we have observed in Equation 2-24.

Single Implicit MP-LPV System - Version 2

Version 2 of the single implicit MP-LPV system occurs if we switch version 1; combining explicit signal relations with an implicit parameter vector. Such a system occurs if $A_0(\cdot) = \mathcal{E}_{n\times n}$ and $A_1(p^{(0)}(k))$. The state vector is then represented as:

$$\begin{aligned} x(k) = A_0\left(p^{(0)}(k)\right) \otimes x(k) \oplus A_1\left(p^{(0)}(k)\right) \otimes x(k-1) \oplus& \\ \bigoplus_{\mu=2}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) &\oplus B(p^{(1)}(k)) \otimes u(k) \\ = \mathcal{E}_{n\times n} \oplus A_1\left(p^{(0)}(k)\right) \otimes x(k-1) \oplus& \\ \bigoplus_{\mu=2}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) &\oplus B(p^{(1)}(k)) \otimes u(k) \\ = A_1\left(p^{(0)}(k)\right) \otimes x(k-1) \oplus& \\ \bigoplus_{\mu=2}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) &\oplus B(p^{(1)}(k)) \otimes u(k) \end{aligned} \tag{3-9}$$

This form however cannot be obtained directly from the canonical form defined in Lemma 3.1; in order to obtain such a form, we need to investigate whether the following form can be obtained from the standard implicit form defined in Equation 3-3:

$$x(k) = A_0(p^{(0)}(k)) \otimes x(k) \oplus \bigoplus_{\mu=1}^{M} (A_\mu(p^{(\mu-1)}(k)) \otimes x(k-\mu)) \tag{3-10}$$

In this thesis, we will only consider version 1 of the single implicit MP-LPV system, as further research is necessary for this level of implicitness. From this moment on, we will refer to the single implicit MP-LPV systems version 1 as single implicit MP-LPV systems.

Doubly Implicit MP-LPV System

Finally we obtain a *doubly implicit* system if both the signal relations and the parameter vector are implicit. This happens when $A_0(p^{(0)}(k)) = A_0(x(k); p^{(1)}(k))$, thus A_0 having a parametric dependence on the current state $x(k)$. This is equivalent to the formal canonical representation of the MP-LPV system as defined in Equation 3-6, in which the state equation is defined as:

$$x(k) = \bigoplus_{\mu=0}^{M} \Big(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu) \Big) \oplus B(p^{(1)}(k)) \otimes u(k) \tag{3-11}$$

3-3 Case Study II: An Urban Railway System as Max-Plus Linear Parameter Varying System

Let us recall the URS introduced in the previous chapter. To model the URS as an MPL system, multiple assumptions have been made simplifying the system dynamics. By extending the system description to an MP-LPV format, we can omit several assumptions. The resulting URS is therefore a more realistic representation.

3-3-1 System Description and Relaxed Assumptions of the Urban Railway System

The URS is no longer restricted to two stations, but contains J number of stations with K the number of trains. Assumption 2.1 is therefore adjusted to:

Assumption 3.2. The route of the URS is formed by $j = 1, ..., J$ stations and $k = 1, ..., K$ trains.

For the URS in MPL format it was furthermore assumed in Assumption 2.9 that the variable actual dwell times $\tau_{\text{dwell},i}$ with $i = 1, 2$ are measurable. We will now approach the dwell time in a more realistic sense, by describing the disembarking and boarding time. Based on these relaxations, we therefore need to consider more states to describe the URS than only the arrival and departure time. For a more realistic and complete description of the URS, we need to obtain elaborate expressions for the following system characteristics:

- $a_j(k)$: arrival time of the k-th train at station j;
- $d_j(k)$: departure time of the k-th train from station j;
- $\tau_{\text{dwell},j}(k)$: dwell time of the k-th train at station j

Arrival Time

A train k can arrive at a station j after the platform at station j is cleared (due to Assumption 2.7). The arrival time $a_j(k)$ of train k at station j furthermore depends on the time it takes for the train k to travel from the previous station $j-1$ to j. The following values therefore need to be taken into account:

- Departure time of train k from previous station $j-1$, $d_{j-1}(k)$;
- Running time $\tau_{r,j}(k)$ between station $j-1$ and j for train k;
- Time between two consecutive trains for safety (*minimum headway time* $\tau_{h,\min}$);
- Departure time of previous train $k-1$ from station j, $d_j(k-1)$.

Let us add the following assumptions to the system description:

Assumption 3.3. The minimum headway time $\tau_{h,\min}$ is assumed to be a fixed value in time units for all consecutive trains.

Assumption 3.4. The running time $\tau_{r,j}(k)$ of train k between station $j-1$ and j can vary per route part, but is assumed to be known in advance.

Based on these factors, we can obtain the following expression for the arrival time:

$$a_j(k) = \max(d_j(k-1) + \tau_{h,\min}, d_{j-1}(k) + \tau_{r,j}(k)) \tag{3-12}$$

Departure Time

A train k can depart from station j as soon as the dwell time has past. A minimum dwell time $\tau_{\text{dwell,min}}$ is considered at each station, to prevent that a train can skip a stop. The departure time $d_j(k)$ of train k at station j therefore depends on:

- Arrival time of train k at station j, $a_j(k)$;
- Dwell time of train k at station j, $\tau_{\text{dwell},j}(k)$;
- Minimum dwell time $\tau_{\text{dwell,min}}$; a fixed value for all stations.

We can therefore obtain the following expression for the departure time:

$$d_j(k) = \max(a_j(k) + \tau_{\text{dwell,min}}, a_j(k) + \tau_{\text{dwell},j}(k)) \tag{3-13}$$

Dwell time

The dwell time of a train represents the time it spends on a platform. During this dwell time, passengers can disembark and board the train. Let us add the following assumptions:

Assumption 3.5. The passengers taking part in the URS are assumed to follow the common courtesy that passengers first disembark, and thereafter board. This is to overcome overlap in the disembarking and boarding time.

For now, we still consider Assumption 2.4 which considers an unlimited capacity of the trains for simplicity purposes. The section hereafter however we will include a maximum capacity. The dwell time $\tau_{\text{dwell},j}(k)$ of train k at station j thus depends on:

- Time it takes for passengers to disembark train k at station j, or the disembarking time $\tau_{d,j}(k)$;
- Time it takes for passengers to board train k at station j, or boarding time $\tau_{b,j}(k)$.

Based on this, we obtain:

$$\tau_{\mathrm{dwell},j}(k) = \tau_{d,j}(k) + \tau_{b,j}(k) \tag{3-14}$$

Disembarking time

The disembarking time represents the time it takes for passengers to disembark. Let us add the following assumptions:

Assumption 3.6. At every station, a fixed fraction β_j of passengers disembarks. It is assumed that this is known in advance. Furthermore, it is assumed that when a passenger disembarks a train, he/she immediately leaves the station.

Assumption 3.7. The number of passengers that can disembark per time unit, or disembarking speed f is fixed and known.

The dwell time $\tau_{d,j}(k)$ of train k at station j therefore also depends on the number of passengers present in the train. For this, let us introduce $\rho_j(k)$; the number of passengers in train k at moment of departure from station j. The dwell time thus depends on:

- The fixed fraction of disembarking passengers β_j;
- The disembarking speed f;
- Number of passengers present in train k on moment of arrival at station j, $\rho_{j-1}(k)$.

We can define this disembarking time $\tau_{d,j}(k)$ as:

$$\tau_{d,j}(k) = \frac{\beta_j}{f}\rho_{j-1}(k) \tag{3-15}$$

Number of Passengers in the Train at Moment of Departure

The variable $\rho_j(k)$ represents the number of passengers present in train k on the moment of departure from station j (i.e. the number of passengers in train k on the moment of arrival at station $j+1$). We assume the following:

Assumption 3.8. Every train k arriving at the initial station 1 arrives with an initial, known number of passengers present in the train, $\rho_0(k)$.

Assumption 3.9. Passengers arrive at a constant and known rate e_j at each station (continuously).

Finally, let us for now assume the following:

Assumption 3.10. The number of passengers in the train at moment of departure from the previous station $\rho_{j-1}(k)$ is assumed to be measurable. Note that this again is a realistic assumption, as there exist many URS in which passengers check-in in the particular vehicle. An example for this is the Dutch tram system.

As the disembarking time $\tau_{d,j}(k)$ only depends on $\rho_{j-1}(k)$ and not on $\rho_j(k)$, we do not need an expression for $\rho_j(k)$.

Boarding time

For the boarding time $\tau_{b,j}(k)$ of train k at station j we assume the following:

Assumption 3.11. The number of passengers that can disembark per time unit, or boarding speed b, is fixed and known.

We furthermore introduce $q_j(k)$; the number of passengers boarding train k at station j, which will be expressed in more detail later on. The boarding time therefore depends on:

- Number of passengers boarding $q_j(k)$;
- Boarding speed b.

We obtain therefore the following expression:

$$\tau_{b,j}(k) = \frac{1}{b} q_j(k) \tag{3-16}$$

Number of Passengers Boarding

As we are still considering an unlimited capacity in the trains, all passengers that are waiting on the platform to board fit in the train (and no passengers are left behind). Let us assume the following:

Assumption 3.12. Passengers arrive at each station at a constant, known rate e. We furthermore assume, equivalently as for disembarking, that passengers arriving at the station are directly ready to board.

Under Assumption 2.4, the number $q_j(k)$ of passengers boarding train k at station j depends only on the number of passengers arriving between departures of consecutive trains. We can therefore express $q_j(k)$ as:

$$q_j(k) = e_j(d_j(k) - d_j(k-1)) \tag{3-17}$$

3-3-2 MP-LPV State-Space Description of the Urban Railway System

We have defined all the necessary variables of the URS system to define a more complete system description. For this, let us collect the previous expressions and after substitutions we can obtain:

$$\begin{aligned} a_j(k) &= \max\Big(d_j(k-1) + \tau_{h,\min}, d_{j-1}(k) + \tau_{r,j}(k)\Big) \\ d_j(k) &= \max\Big(a_j(k) + \tau_{\text{dwell},\min}, a_j(k) + \tau_{\text{dwell},j}(k)\Big) \\ \tau_{\text{dwell},j}(k) &= \frac{\beta_j}{f}\rho_{j-1}(k) + \frac{e_j}{b}\left(d_j(k) - d_j(k-1)\right) \end{aligned} \tag{3-18}$$

It can be observed that the expression for the departure time is implicit, as it depends on $\tau_{\text{dwell},j}(k)$ which itself also depends on $d_j(k)$. We will later on observe that the implicitness level is doubly implicit. Let us first define the following state vector for the URS:

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{2J-1}(k) \\ x_{2J}(k) \end{bmatrix} = \begin{bmatrix} a_1(k) \\ d_1(k) \\ \vdots \\ a_J(k) \\ d_J(k) \end{bmatrix} \tag{3-19}$$

By defining the parameter vector correctly, we can obtain the max-plus parameter varying structure, such that the (implicit) relations can be illustrated more clearly. We define:

$$p^{(0)}(k) = \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix} \tag{3-20}$$

We can then define the dwell time $\tau_{\text{dwell}}(k)$ as a plus-scaling function (which is a subclass of MMPS functions) of the parameter vector $p^{(0)}(k)$. Let us illustrate this for station j and train k:

$$\tau_{\text{dwell},j}(k) = \begin{bmatrix} 0 & \frac{e_j}{b} & 0 & -\frac{e_j}{b} \end{bmatrix} \underbrace{\begin{bmatrix} x_{2j-1}(k) \\ x_{2j}(k) \\ x_{2j-1}(k-1) \\ x_{2j}(k-1) \end{bmatrix}}_{p_j^{(0)}(k)} + \frac{\beta_j}{f}\rho_{j-1}(k) \tag{3-21}$$

Let us finally add the following assumption in addition to Assumption 3.8, which is necessary to obtain the state-space description:

Assumption 3.13. We assume that, in addition to $\rho_0(k)$, the following initial values are known: $\tau_{\text{dwell},1}(k)$, $d_0(k)$ and $d_j(0)$.

We furthermore consider $d_j(0)$ as an input for the URS. We can now write the state-space description of the URS, with the state vector as defined in Equation 3-19:

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{2J-1}(k) \\ x_{2J}(k) \end{bmatrix} = \underbrace{\begin{bmatrix} \varepsilon & \dots & \dots & \dots & \dots & \varepsilon \\ \begin{array}{c}\tau_{\text{dwell,min}}(k) \\ \oplus\tau_{\text{dwell},1}(k)\end{array} & \varepsilon & \dots & \dots & \dots & \varepsilon \\ \varepsilon & \tau_{r,2}(k) & \varepsilon & \dots & \dots & \varepsilon \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \varepsilon & \varepsilon & \dots & \ddots & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \dots & \dots & \begin{array}{c}\tau_{\text{dwell,min}}(k) \\ \oplus\tau_{\text{dwell},J}(k)\end{array} & \varepsilon \end{bmatrix}}_{A_0(p^{(0)}(k))} \otimes \underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{2J-1}(k) \\ x_{2J}(k) \end{bmatrix}}_{x(k)}$$
$$\oplus \underbrace{\begin{bmatrix} \varepsilon & \tau_{h,\min} & \varepsilon & \dots & \dots & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \dots & \dots & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \dots & \dots & \varepsilon \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \varepsilon & \varepsilon & \varepsilon & \ddots & \ddots & \tau_{h,\min} \\ \varepsilon & \varepsilon & \varepsilon & \dots & \dots & \varepsilon \end{bmatrix}}_{A_1} \otimes \underbrace{\begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ \vdots \\ x_{2J-1}(k-1) \\ x_{2J}(k-1) \end{bmatrix}}_{x(k-1)}$$
$$\oplus \underbrace{\begin{bmatrix} \tau_{r,1}(k) & \varepsilon & \varepsilon & \dots & \varepsilon \end{bmatrix}}_{B} \otimes \underbrace{\begin{bmatrix} d_0(k) \\ \varepsilon \\ \varepsilon \\ \vdots \\ \varepsilon \end{bmatrix}}_{u(k)} \tag{3-22}$$

It can be observed that we are dealing with a doubly implicit MP-LPV structure, due to the presence of the parameter vector $p_j^{(0)}(k)$ in the $\tau_{\text{dwell},j}(k)$ expressions as defined in Equation 3-21 in the $A_0(\cdot)$ matrix, and the presence of the $x(k)$ vector on both the left and the right side of the equation. Let us investigate if we can resolve part of this implicitness and obtain a single implicit MP-LPV system. For this, let us first define the number of passengers actually boarding:

$$q_{\text{actual},j}(k) = b\left(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)\right) \tag{3-23}$$

Note that $d_j(k) - a_j(k)$ is the actual dwell time, and $\frac{\beta_j}{f}\rho_{j-1}(k)$ the disembarking time as defined in Equation 3-15. Therefore, the value in between the brackets is another expression for the boarding time. As b represents the number of passengers boarding per time index, we have indeed obtained the actual number of passengers boarding. We can therefore say that the following must hold:

$$q_j(k) = e_j(d_j(k) - d_j(k-1)) = b\left(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)\right) \tag{3-24}$$

We can rewrite this to obtain the following:

$$d_j(k) = \underbrace{-\frac{b}{e_j - b}}_{\lambda_1} a_j(k) \underbrace{-\frac{b\beta_j}{e_j f - fb}}_{\lambda_2} \rho_{j-1}(k) + \underbrace{\frac{e_j}{e_j - b}}_{\lambda_3} d_j(k-1) \tag{3-25}$$

If we now include the minimum dwell time, we obtain:

$$d_j(k) = \max\left(a_j(k) + \tau_{\text{dwell,min}}, \lambda_1 a_j(k) + \lambda_2 \rho_{j-1}(k) + \lambda_3 d_j(k-1)\right) \tag{3-26}$$

Thus we then have the following expressions describing the URS system:

$$\begin{aligned} a_j(k) &= \max\left(d_j(k-1) + \tau_{h,\min}, d_{j-1}(k) + \tau_{r,j}(k)\right) \\ d_j(k) &= \max\left(a_j(k) + \tau_{\text{dwell,min}}, \lambda_1 a_j(k) + \lambda_2 \rho_{j-1}(k) + \lambda_3 d_j(k-1)\right) \end{aligned} \tag{3-27}$$

It can be observed that the expressions now are only single implicit. To obtain this MP-LPV state-space description, we consider the same state vector as in Equation 3-19 and the same parameter vector as in Equation 3-20. Let us define the departure time as a function of this parameter vector. For train k at station j, we have:

$$\begin{aligned} d_j(k) &= \max\left(\tau_{\text{dwell,min}}, (\lambda_1 - 1)a_j(k) + \lambda_2 \rho_{j-1}(k) + \lambda_3 d_j(k-1)\right) + a_j(k) \\ &= \max\left(\tau_{\text{dwell,min}}, \underbrace{\begin{bmatrix} \lambda_1 - 1 & 0 & 0 & \lambda_3 \end{bmatrix} \underbrace{\begin{bmatrix} x_{2j-1}(k) \\ x_{2j}(k) \\ x_{2j-1}(k-1) \\ x_{2j}(k-1) \end{bmatrix}}_{p_j^{(0)}(k)} + \lambda_2 \rho_{j-1}(k)}_{f_{\text{MMPS}}\left(p_j^{(0)}(k)\right)}\right) + a_j(k) \\ &= \max\left(\tau_{\text{dwell,min}}, f_{\text{MMPS}}\left(p_j^{(0)}(k)\right)\right) + a_j(k) \end{aligned} \tag{3-28}$$

The resulting maximization is again an MMPS function, and we therefore have:

$$d_j(k) = f_{\text{MMPS}}\left(p_j^{(0)}(k)\right) + a_j(k) \tag{3-29}$$

Note that we are allowed to consider MMPS functions $f_{\text{MMPS}}(\cdot)$ instead of continuous Piecewise Affine (PWA) functions $f_{\text{cPWA}}(\cdot)$ as assumed in Assumption 3.1, as we have seen in subsection 2-5-4 that these classes of systems are equivalent. We can now obtain an MP-LPV system described by:

$$
\begin{aligned}
x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{2J-1}(k) \\ x_{2J}(k) \end{bmatrix} = \begin{bmatrix} a_1(k) \\ d_1(k) \\ a_2(k) \\ \vdots \\ a_J(k) \\ d_J(k) \end{bmatrix} = \\
\underbrace{\begin{bmatrix} \varepsilon & \dots & \dots & \dots & \dots & \varepsilon \\ f_{\text{MMPS}}\left(p_1^{(0)}(k)\right) & \varepsilon & \dots & \dots & \dots & \varepsilon \\ \varepsilon & \tau_{r,2}(k) & \varepsilon & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \varepsilon & \varepsilon & \dots & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \dots & \varepsilon & f_{\text{MMPS}}\left(p_J^{(0)}(k)\right) & \varepsilon \end{bmatrix}}_{A_0(p^{(0)}(k))} \otimes \underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{2J-1}(k) \\ x_{2J}(k) \end{bmatrix}}_{x(k)} \\
\oplus \underbrace{\begin{bmatrix} \varepsilon & \tau_{h,\min} & \varepsilon & \dots & \dots & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \dots & \dots & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \varepsilon & \varepsilon & \varepsilon & \ddots & \ddots & \tau_{h,\min} \\ \varepsilon & \varepsilon & \varepsilon & \dots & \dots & \varepsilon \end{bmatrix}}_{A_1(p^{(1)}(k))} \otimes \underbrace{\begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ \vdots \\ x_{2J-1}(k-1) \\ x_{2J}(k-1) \end{bmatrix}}_{x(k-1)} \\
\oplus \underbrace{\begin{bmatrix} \tau_{r,1}(k) & \varepsilon & \varepsilon & \dots & \varepsilon \end{bmatrix}}_{B} \otimes \underbrace{\begin{bmatrix} d_0(k) \\ \varepsilon \\ \varepsilon \\ \vdots \\ \varepsilon \end{bmatrix}}_{u(k)}
\end{aligned}
\tag{3-30}
$$

On first hand, the resulting MP-LPV system seems doubly implicit due to the presence of the parameter vector $p^{(0)}(k)$ in the A_0 matrix and the presence of the current state vector $x(k)$ on both the left and the right side of the expression. Let us however investigate this MMPS function. As can be observed in Equation 3-28, we have:

$$
f_{\text{MMPS}}\left(p_j^{(0)}(k)\right) = f_{\text{MMPS}}(a_j(k), d_j(k-1), \rho_{j-1}(k)) \tag{3-31}
$$

Furthermore, we have that only the departure time $d_j(k)$ depends on $f_{\text{MMPS}}(p_j^{(0)}(k))$. Clearly, the arrival time $a_j(k)$ is known on the moment the train k departs from station j. Therefore, if we correctly maintain the order of the state vector, we can conclude that the resulting MP-LPV system is single implicit. The URS that is considered now is still subject to several assumptions that can be relaxed. In the next section we will relax these assumptions and we will observe that the resulting system initially is derived as an MMPS system.

3-4 Case Study III: Urban Railway System as MMPS System

In this case study, we will relax the assumptions for the URS system even more, resulting in an MMPS system description. First, let us remove Assumption 2.4, which considers an unlimited capacity of the trains. Removing this assumption results in the possibility that not all passengers fit in the train and therefore are left behind. Let us thus add a new assumption:

Assumption 3.14. The trains have a limited number of seats available, defined as the maximum capacity $\rho_{\max}$.

Because of this assumption, we are now dealing with two possible scenarios:

1. The passengers waiting on the platform to board all fit in the train. The train is then either filled up until the maximum capacity ($\rho_j(k) = \rho_{\max}$), or there are still empty seats available ($\rho_j(k) < \rho_{\max}$).
2. There are more passengers waiting on the platform than empty seats available. We then have $\rho_j(k) = \rho_{\max}$ and passengers are left behind on the platform.

In the following subsection, we will adjust the URS states according to these two scenarios.

3-4-1 System Description and Relaxed Assumptions of the Urban Railway System

To describe scenario 1 and 2, we need to adjust the variables $\rho_j(k)$ and $q_j(k)$, and we need to introduce another variable $\sigma_j(k)$; the number of passengers left behind by train k on station j. All assumptions that are defined so far, still hold, unless stated otherwise.

Updated: Number of Passengers in the Train at Moment of Departure

Because of the addition of Assumption 3.14, we need to adjust the variable $\rho_j(k)$. In Assumption 3.10, we have assumed that $\rho_{j-1}(k)$ is measurable. However, let us now correctly describe $\rho_j(k)$ such that we can include its influence. The number of passengers $\rho_j(k)$ depends on the following:

- Number of passengers that remain seated (and thus not disembark), $(1 - \beta_j)\rho_{j-1}(k)$;
- Number of passengers boarding, $q_j(k)$.

We can define the expression for $\rho_j(k)$ by:

$$\rho_j(k) = (1 - \beta_j)\rho_{j-1}(k) + q_j(k) \tag{3-32}$$

Updated: Number of Passengers Boarding

Because of the introduction of scenario 2 due to Assumption 3.14 the expression for $q_j(k)$ also needs to be updated. In Equation 3-17, we have defined the value $q_j(k)$ in case scenario 1 occurs. However, this expression needs to be updated as we are now also considering $\sigma_j(k)$, the number of passengers left behind by train k on platform j. The number of passengers boarding in scenario 1 now depends on the following factors:

- The number of passengers arriving between departures of consecutive trains, $e_j(d_j(k)-d_j(k-1)$;
- The number of passengers left behind by the previous train, $\sigma_j(k-1)$.

We can therefore define the new expression $q_{j,1}(k)$ for scenario 1:

$$q_{j,1}(k) = e_j(d_j(k) - d_j(k-1)) + \sigma_j(k-1) \tag{3-33}$$

In case of the second scenario, not all passengers waiting on the platform can actually board the train. Therefore, $q_{j,2}(k)$ depends on:

- Maximum capacity $\rho_{\max}$;
- Number of passengers present in train k at arrival at station j, $\rho_{j-1}(k)$;
- Number of passengers disembarking, $\beta_j\rho_{j-1}(k)$ (and therefore also the number of passengers remaining seated $(1-\beta_j)\rho_{j-1}(k)$).

The number of passengers boarding in scenario 2 can be defined as:

$$q_{j,2}(k) = \rho_{\max} - (1-\beta_j)\rho_{j-1}(k) \tag{3-34}$$

We can therefore define the actual number of passengers boarding train k at station j as:

$$q_j(k) = \min\left(e_j\left(d_j(k) - d_j(k-1)\right) + \sigma_j(k-1), \rho_{\max} - (1-\beta_j)\rho_{j-1}(k)\right) \tag{3-35}$$

Passengers Left Behind

It remains to describe $\sigma_j(k)$, the number of passengers left behind by train k at station j. This variable depends on the following:

- Number of passengers on the platform at beginning of boarding time, or equivalently the number of passengers that want to board;
- Number of passengers actually boarding (this value depends on whether scenario 1 or 2 occurs).

We can therefore express $\sigma_j(k)$ as:

$$\sigma_j(k) = e_j(d_j(k) - d_j(k-1)) + \sigma_j(k-1) - q_j(k) \tag{3-36}$$

Note that $\sigma_j(k)$ will always be zero in case of scenario 1.

3-4-2 MMPS Description of the Urban Railway System

Combining all previous definitions of the system variables and substituting where logical, we obtain the following expressions that describe the complete URS:

$$\begin{aligned} a_j(k) &= \max(d_j(k-1) + \tau_{h,\min}, d_{j-1}(k) + \tau_{r,j}(k)) \\ d_j(k) &= \max\left(a_j(k) + \tau_{\text{dwell,min}}, a_j(k) + \frac{\beta_j}{f}\rho_{j-1}(k) + \frac{1}{b}q_j(k)\right) \\ q_j(k) &= \min\left(e_j\left(d_j(k) - d_j(k-1)\right) + \sigma_j(k-1), \rho_{\max} - (1-\beta_j)\rho_{j-1}(k)\right) \\ \rho_j(k) &= (1-\beta_j)\rho_{j-1}(k) + q_j(k) \\ \sigma_j(k) &= e_j(d_j(k) - d_j(k-1)) + \sigma_j(k-1) - q_j(k) \end{aligned} \tag{3-37}$$

Just as for the previous case we can observe that we are dealing with doubly implicitness, due to the expression of $q_j(k)$. Let us resolve this implicitness in the following manner. For this, let us recall the number of passengers actually boarding:

$$q_{\text{actual},j}(k) = b\left(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)\right) \tag{3-38}$$

If scenario 1 holds, we have:

$$q_{\text{actual},j}(k) = b\left(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)\right) = e_j\left(d_j(k) - d_j(k-1)\right) + \sigma_j(k-1) \tag{3-39}$$

Rewriting this and taking all $d_j(k)$ to the left side results in:

$$d_j(k) = \underbrace{\frac{b}{b-e_j}}_{\alpha_1} a_j(k) + \underbrace{\frac{b}{b-e_j}\frac{\beta_j}{f}}_{\alpha_2} \rho_{j-1}(k) + \underbrace{\frac{1}{b-e_j}}_{\alpha_3} \sigma_j(k-1) \underbrace{-\frac{e_j}{b-e_j}}_{\alpha_4} d_j(k-1) \tag{3-40}$$

If scenario 2 holds, we can write:

$$q_{\text{actual},j}(k) = b\left(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)\right) = \rho_{\max} - (1-\beta_j)\rho_{j-1}(k) \tag{3-41}$$

If we rewrite this again, we obtain:

$$d_j(k) = \underbrace{\frac{1}{b}\rho_{\max}}_{\gamma_0} \underbrace{+}_{\gamma_1} a_j(k) + \underbrace{(\frac{\beta_j}{f} - \frac{1-\beta_j}{b})}_{\gamma_2} \rho_{j-1}(k) \tag{3-42}$$

Therefore, we can write the following for the departure time:

$$\begin{aligned} d_j(k) = \min(&\alpha_1 a_j(k) + \alpha_2\rho_{j-1}(k) + \alpha_3\sigma_j(k-1) + \alpha_4 d_j(k-1), \\ &\gamma_0 + \gamma_1 a_j(k) + \gamma_2\rho_{j-1}(k)) \end{aligned} \tag{3-43}$$

If we now add the minimum dwell time $\tau_{\text{dwell},j,\min}$ to the departure time, we end up with:

$$\begin{aligned} d_j(k) =& \max(a_j(k) + \tau_{\text{dwell},j,\min}, \min(\alpha_1 a_j(k) + \alpha_2\rho_{j-1}(k) + \alpha_3\sigma_j(k-1) + \alpha_4 d_j(k-1), \\ &\gamma_0 + \gamma_1 a_j(k) + \gamma_2\rho_{j-1}(k))) \end{aligned} \tag{3-44}$$

We finally add the following assumption, as an update of Assumption 3.13:

Assumption 3.15. We assume that the following initial values are known: $d_0(k)$, $d_j(0)$, $\rho_0(k)$ and $\sigma_j(0)$.

Let us now define the state vector that includes all the states that describe the URS system completely:

$$x(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_{5J}(k) \end{bmatrix} = \begin{bmatrix} a_1(k) \\ d_1(k) \\ q_1(k) \\ \rho_1(k) \\ \sigma_1(k) \\ \vdots \\ a_J(k) \\ d_J(k) \\ q_J(k) \\ \rho_J(k) \\ \sigma_J(k) \end{bmatrix} \tag{3-45}$$

With therefore, for station j and train k:

$$\begin{bmatrix} x_j(k) \\ \hline x_{j+1}(k) \\ \\ \\ \hline x_{j+2}(k) \\ \\ \hline x_{j+3}(k) \\ \hline x_{j+4}(k) \end{bmatrix} = \begin{bmatrix} \max(x_{j+1}(k-1) + \tau_{h,\min}, x_{j-4}(k) + \tau_{r,j}(k)) \\ \hline \max(x_j(k) + \tau_{\text{dwell},j,\min}, \\ \min(\alpha_1 x_j(k) + \alpha_2 x_{j-2}(k) + \alpha_3 x_{j+4}(k-1) + \alpha_4 x_{j+1}(k-1), \\ \gamma_0 + \gamma_1 x_j(k) + \gamma_2 x_{j-2}(k))) \\ \hline \min(e_j\,(x_{j+1}(k) - x_{j+1}(k-1)) + x_{j+4}(k-1), \\ \rho_{\max} - (1-\beta_j) x_{j-2}(k)) \\ \hline (1-\beta_j) x_{j-2}(k) + x_{j+2}(k) \\ \hline e_j(x_{j+1}(k) - x_{j+1}(k-1)) + x_{j+4}(k-1) - x_{j+2}(k) \end{bmatrix} \tag{3-46}$$

Note that the expressions contain both maximization as well as minimization. Furthermore, all above expressions contain the multiplication operation. Therefore, the expressions that we obtained now are actually MMPS functions. We have seen however in subsection 3-2-1 that under Assumption 3.1 the class of MP-LPV systems and MMPS systems coincide. Let us therefore show how we can rewrite the above state equations into expressions that fit into the MP-LPV framework, for the initial station $j = 1$. Let us consider the following parameter vector for $j = 1$:

$$p_1^{(0)}(k) = \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix} = \begin{bmatrix} a_1(k) \\ d_1(k) \\ q_1(k) \\ \rho_1(k) \\ \sigma_1(k) \\ a_1(k-1) \\ d_1(k-1) \\ q_1(k-1) \\ \rho_1(k-1) \\ \sigma_1(k-1) \end{bmatrix} \tag{3-47}$$

We can then write the following for the departure time $d_1(k)$:

$$\begin{aligned} d_1(k) = \max\Big(&\tau_{\text{dwell},1,\min}, \min((\alpha_1-1)a_1(k)+\alpha_2\rho_0(k)+\alpha_3\sigma_1(k-1)+\alpha_4 d_1(k-1),\\ &\gamma_0+(\gamma_1-1)a_1(k)+\gamma_2\rho_0(k))\Big)+a_1(k)\\ &= f_{\text{MMPS}}\Big(a_1(k),\rho_0,\sigma_1(k-1),d_1(k-1)\Big)+a_1(k)\\ &= f_{\text{MMPS},1}\left(p_1^{(0)}(k)\right)+a_1(k) \end{aligned} \tag{3-48}$$

Similarly, we have for number of passengers boarding:

$$\begin{aligned} q_1(k) =& \min\Big((e_1-1)d_1(k)-e_1 d_1(k-1)+\sigma_1(k-1),\\ &\rho_{\max}-(1-\beta_1)\rho_0(k)-d_1(k)\Big)+d_1(k)\\ =&f_{\text{MMPS}}\Big(d_1(k),d_1(k-1),\sigma_1(k-1),\rho_0\Big)+d_1(k)\\ =&f_{\text{MMPS},2}\left(p_1^{(0)}(k)\right)+d_1(k) \end{aligned} \tag{3-49}$$

And for the number of passengers in the train at moment of departure $\rho_1(k)$:

$$\begin{aligned} \rho_1(k) &= (1-\beta_1)\rho_0(k)+q_1(k)\\ &= f_{\text{MMPS}}\Big(\rho_0\Big)+q_1(k)\\ &= f_{\text{MMPS},3}\left(p_1^{(0)}(k)\right)+q_1(k) \end{aligned} \tag{3-50}$$

And finally, for the number of passengers left behind $\sigma_1(k)$:

$$\begin{aligned} \sigma_1(k) &= e_1(d_1(k)-d_1(k-1))+\sigma_1(k-1)-2q_1(k)+q_1(k)\\ &= f_{\text{MMPS}}\Big(d_1(k),d_1(k-1),\sigma_1(k-1),q_1(k)\Big)+q_1(k)\\ &= f_{\text{MMPS},4}\left(p_1^{(0)}(k)\right)+q_1(k) \end{aligned} \tag{3-51}$$

Recall that we have assumed in Assumption 3.15 that the initial values $d_0(k)$ and $\rho_0(k)$ are known. We can therefore obtain the following MP-LPV system:

$$\begin{aligned} \begin{bmatrix} a_1(k)\\ d_1(k)\\ q_1(k)\\ \rho_1(k)\\ \sigma_1(k)\end{bmatrix} =& \underbrace{\begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon\\ f_{\text{MMPS},1}\left(p_1^{(0)}(k)\right) & \varepsilon & \varepsilon & \varepsilon & \varepsilon\\ \varepsilon & f_{\text{MMPS},2}\left(p_1^{(0)}(k)\right) & \varepsilon & \varepsilon & \varepsilon\\ \varepsilon & \varepsilon & f_{\text{MMPS},3}\left(p_1^{(0)}(k)\right) & \varepsilon & \varepsilon\\ \varepsilon & \varepsilon & f_{\text{MMPS},4}\left(p_1^{(0)}(k)\right) & \varepsilon & \varepsilon \end{bmatrix}}_{A_0(p_1^{(0)}(k))} \otimes \underbrace{\begin{bmatrix} a_1(k)\\ d_1(k)\\ q_1(k)\\ \rho_1(k)\\ \sigma_1(k)\end{bmatrix}}_{x(k)}\\ &\oplus \underbrace{\begin{bmatrix} \varepsilon & \tau_{h,\min} & \varepsilon & \varepsilon & \varepsilon\\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon\\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon\\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon\\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}}_{A_1(p^{(1)}(k))} \otimes \underbrace{\begin{bmatrix} a_1(k-1)\\ d_1(k-1)\\ q_1(k-1)\\ \rho_1(k-1)\\ \sigma_1(k-1)\end{bmatrix}}_{x(k-1)} \oplus \underbrace{\begin{bmatrix}\tau_{r,1}(k) & \varepsilon & \varepsilon & \dots & \varepsilon\end{bmatrix}}_{B} \otimes \underbrace{\begin{bmatrix} d_0(k)\\ \varepsilon\\ \varepsilon\\ \vdots\\ \varepsilon\end{bmatrix}}_{u(k)} \end{aligned} \tag{3-52}$$

Again, as continuous PWA functions and MMPS functions are equivalent, we are allowed to use MMPS functions in the A_0 matrix. Similarly as for the MP-LPV state-space description defined in Equation 3-30 for the URS of Case Study II, the description seems doubly implicit on first sight. However, due to the construction of the MMPS functions in Equation 3-48, 3-49, 3-50 and 3-51 we can conclude that the URS is single implicit. Let us elaborate on this, by observing $f_{\text{MMPS},1}(p^{(0)}(k))$. This function is located at entry $[A_0(\cdot)]_{21}$ and therefore is only present at the state equation for the departure time $d_1(k)$. The only implicit value present in this entry is the arrival time $a_1(k)$. But if we maintain the order of the states as defined when calculating the states, we can conclude that we first calculate $a_1(k)$, and thereafter $d_1(k)$. This is confirmed by the logical reasoning that the arrival time of train k at station j is known at the departure time of train k from station j. Therefore, when calculating $d_1(k)$, this value $a_1(k)$ can be considered as a known value. The same reasoning can be implemented on the remaining MMPS functions $f_{\text{MMPS},i}(p^{(0)}(k))$ with $i = 2, ..., 4$, confirming that the resulting URS is indeed single implicit.

Chapter 4

Solvability of Max-Plus Linear Parameter Varying Systems

In this chapter, we discuss the solvability of Max-Plus Linear Parameter Varying (MP-LPV) systems, which were introduced in the previous chapter. We have seen that MP-LPV systems can present themselves with different levels of implicitness. This implicitness occurs as a result of the presence of the state $x(k)$ on both the left as the right side of the state equation and the presence of the maximization operation. Due to the absence of an inverse of the maximization operation we cannot easily solve the state equation for $x(k)$. It gives rise to the solvability problem; we can not at all times guarantee the existence of a solution to the state equation of the MP-LPV system. We therefore aim to find a set of conditions that does guarantee this existence of this state solution. With the different levels of implicitness, we will construct different sets of conditions to ensure solvability for the concerning system. This chapter is organized as follows. In the first section, we present an updated definition of the solvability framework for MP-LPV systems. Thereafter we will discuss the solvability of explicit MP-LPV systems in section 4-2. We will present the problems for solvability for single implicit MP-LPV systems in section 4-3, after which we will suggest a solvability framework in section 4-4. Subsequently, we will approach doubly implicit MP-LPV systems in the same manner in section 4-5 and section 4-6. Finally in section 4-7 we discuss the solvability of Case Study I and II of the previous chapter.

4-1 Solvability Framework

In this section, we present the definition of solvability of MP-LPV systems. There has been done some research in solvability for MP-LPV systems in [10] and [8], but the resulting framework is incomplete and subject to several strict assumptions. Let us first present the latter solvability framework in subsection 4-1-1, after which we can present the updated solvability definitions in subsection 4-1-2.

4-1-1 Background

In this subsection, we present the solvability framework as proposed in [8], which has shown to be incomplete and subject to assumptions that can be relaxed. This internal document defines solvability of an MP-LPV system in the following manner:

Definition 4.1 (Solvability of a general MP-LPV System). *The MP-LPV system as defined in Lemma 3.1 is considered* solvable *if for every $x(k-j) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $z(k) \in \mathbb{R}^z$ with $k \geq M$ and $j \in \{1, ..., M\}$, at least one state $x(k)$ exists such that the dynamics of the state-space description hold.*

In the previous chapter we have seen three levels of implicitness for MP-LPV systems, namely explicit, single implicit and doubly implicit systems. Let us introduce the solvability framework per level of implicitness as proposed by [8]. We have seen that an *explicit* MP-LPV system occurs when $A_0(\cdot) = \mathcal{E}_{n\times n}$. We have furthermore seen that the state equation in such an explicit format is only depending on previous states $x(k-1), ..., x(k-M)$. Obtaining a solvability framework for explicit MP-LPV systems will therefore not give rise to any problems.

A *single implicit* system occurs when $A_0(p(\cdot)) = A_0(p^{(\eta)}(\cdot))$ for $\eta \geq 1$. This level of implicitness is comparable to the Max-Plus-Linear (MPL) system introduced in Equation 2-24 in Chapter 2. The paper [8] proposes that the implicitness of a single implicit system can be resolved using the Kleene star operator as defined in Definition 2.3 resulting in an explicit system. The paper states that under Assumption 3.1, solvability of such a system is ensured if the Kleene star $A_0^*(p^{(1)})$ exists. Furthermore, [8] states that the uniqueness of the solution is guaranteed if the communication graph $\mathcal{G}(A_0(\cdot))$ has negative circuit weights. We will investigate, expand and improve this idea of solvability of single implicit MP-LPV systems later in this chapter.

Let us finally present the solvability problem of a *doubly implicit* MP-LPV system as proposed by [8]. Unfortunately, we cannot translate this doubly implicit MP-LPV into an explicit form using the Kleene Star operator. This is due to the presence of the current state $x(k)$ in the parameter vector and therefore in the entries of matrix $A_0(\cdot)$; the absence of an inverse of the maximization property imprisons this $x(k)$. [8] provides an algebraic approach to this solvability problem, and presents a sufficient condition that ensures the existence and uniqueness of the state trajectories under the following assumptions:

Assumption 4.1. *Consider the doubly implicit MP-LPV system as defined in subsection 3-2-2. We assume the following:*

- *Matrix $A_0(\cdot)$ is reducible and in Frobenius normal form*;
- *The finite components of matrix $A_0(\cdot)$ are Max-Plus-Scaling (MPS) functions*[1] *of the parameter $p(k)$*;
- *The finite entries of the matrix $A_0(\cdot)$ are defined as: $[A_0(\cdot)]_{ij} = [A_0(\cdot)]_{ij}(x^{(-i)}(k), x_i(k), p^{(1)})$ for $i \leq j$, $i, j \in \{1, ..., n\}$, with $x^{(-i)}(k)$ defined as $x^{(-i)}(k) = [x_1(k) \ \dots \ x_{i-1}(k)]^T$.*

[1] A MPS function is a recursive function that can only contain the operations maximization, addition and multiplication.

Under this assumption, we can express the state x at event step k as:

$$x_i(k) = \left(f^{(i)}_{\mathrm{MPS}}(x^{(-i)}(k), x_i(k), p^{(1)}(k)), r_i(k)\right) \tag{4-1}$$

In which $r(k) = \bigoplus_{\mu=1}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) \oplus B(p^{(1)}(k)) \otimes u(k)$ and f_{MPS} is a MPS function defined as:

$$f^{(i)}_{\mathrm{MPS}}(\cdot) = \max_{i \in \{1,\dots,N\}} \left(\alpha^{(i)}_l x_i(k) + \beta^{(i)}_l(x^{(-i)}(k), p^{(1)}(k))\right) + \delta^{(i)}_0 \tag{4-2}$$

In which $\delta^{(i)}_0 = \varepsilon$ if $[A_0(\cdot)]_{ij} = \varepsilon$ for all $j \in \{1, \dots, n\}$ and $\delta^{(i)}_0 = 0$ otherwise. Furthermore, all known terms are collected in $\beta^{(i)}_l(\cdot)$, and $\alpha^{(i)}_l$ are real valued coefficients. The condition for solvability as defined in Definition 4.1 of a doubly implicit MP-LPV that [8] presents as a result from these assumptions, is:

Lemma 4.1. *Under Assumption 3.1 and 4.1, we have that for every $i \in \{1, \dots, n\}$ there exists a unique solution $x_i(k)$ to Equation 4-1 if for every $l \in \{1, \dots, N\}$ we have that $\alpha^{(i)}_l < 1$ in $f^{(i)}_{MPS}(\cdot)$ in Equation 4-2.*

More details of the proof of this lemma can be found in [8]. This proof however lacks the concrete proof of existence of a solution to the state equation, and only provides the proof on uniqueness of the solution. This lack therefore gives rise to the need for a more complete solvability framework, separating existence and uniqueness of a state solution. Furthermore, with Assumption 4.1 we restrict the systems to a doubly implicit MP-LPV format for which there is no interdependence between current and future states. For the simplified Urban Railway System (URS) this can make sense; the departure time from a station i is independent of departure times of upcoming stations $j > i$. However, if we would no longer restrict the URS to being uni-directional, this assumption no longer would hold. Therefore, to obtain a solvability framework that is more general and can therefore fit a wider set of possibly more complex MP-LPV systems, there is a need to relax these assumptions. In the remainder of this chapter, we will therefore omit all assumptions in Assumption 4.1, and retain only Assumption 3.1, which is necessary to obtain the canonical form of the MP-LPV system.

4-1-2 Updated Solvability Framework

We have seen the definition for solvability of MP-LPV systems as proposed by [8] in Definition 4.1. We have seen however that this definition is incomplete. Let us therefore extend this definition and introduce a solvability framework consisting of four (separate) levels, namely existence of a solution, unique existence of a solution, solvability of the system and unique solvability of the system. Let us present these definitions for a general MP-LPV system in the format as presented in Lemma 3.1:

Definition 4.2 (Existence of a solution of an MP-LPV system)**.** *The MP-LPV system as defined in Lemma 3.1 is considered to have a solution $x(k)$ to the state equation if for $x(k-1) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $z(k) \in \mathbb{R}^z$ at least one state $x(k)$ exists such that the dynamics of the state-space description hold.*

Definition 4.3 (Unique Existence of a solution of an MP-LPV system)**.** *The MP-LPV system as defined in Lemma 3.1 is considered to have a unique solution $x(k)$ to the state equation if for $x(k-1) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $z(k) \in \mathbb{R}^z$ the solution $x(k)$ exists and is unique such that the dynamics of the state-space description hold.*

Definition 4.4 (Solvability of an MP-LPV System)**.** *The MP-LPV system as defined in Lemma 3.1 is considered* solvable *if, given that existence of a solution as defined in Definition 4.2 is guaranteed, a solution exists for all future states $x(k+j) \in \mathbb{R}^n$ with $j \geq 1$, such that the dynamics of the state-space description hold.*

And finally we can define:

Definition 4.5 (Unique Solvability of an MP-LPV System)**.** *The MP-LPV system as defined in Lemma 3.1 is considered* uniquely solvable *if it is solvable and all solutions $x(k+j)$ for $j \geq 0$ are unique.*

These four levels forming the solvability framework all present different degrees, which is visualized by Figure 4-1. The arrows represent the degrees of solvability; if an MP-LPV system is considered solvable, it is considered automatically that existence holds as well. Furthermore, if we can confirm unique existence, we can confirm existence. Equivalently, if an MP-LPV system is considered uniquely solvable, we can confirm unique existence as well as solvability. If we compare these definitions with the definition for solvability as proposed by [8], we can conclude that Definition 4.1 is actually comparable with the definition of existence as proposed in Definition 4.2.

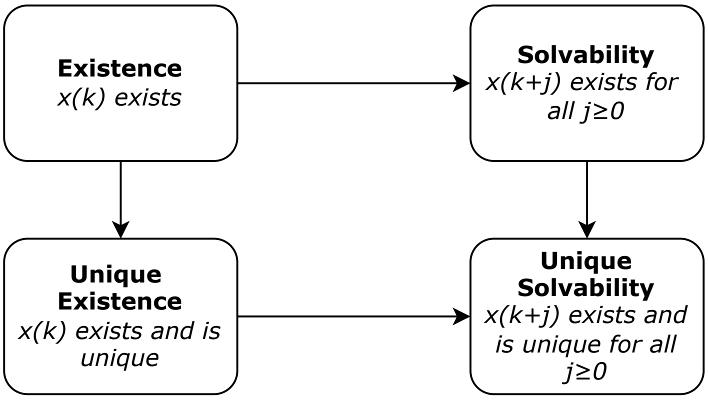


Figure 4-1: The four levels of the solvability framework for MP-LPV systems, with $x(k)$ the solution to the state equation given by $x(k) = \bigoplus_{\mu=0}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) \oplus B(p^{(1)}(k)) \otimes u(k)$. More details of the levels can be found in Definition 4.2, 4.3, 4.4 and 4.5.

In the previous section we have seen how [8] analyzes the solvability property as defined in Definition 4.1 for the different levels of implicitness for MP-LPV systems. Based on this, let us propose an approach to investigate the updated solvability framework. We have seen that analyzing solvability as proposed in [8] for explicit MP-LPV systems does not give rise to any problems and nor will the updated solvability framework, as we will present in the following section. For single implicit MP-LPV systems, we will extend the Kleene star approach in section 4-3 to investigate the four levels of the updated framework separately. Finally, we will disregard all three assumptions made in Assumption 4.1, and we will define a new and more general condition necessary for analyzing the updated solvability framework for doubly implicit MP-LPV systems.

4-2 Solvability of Explicit MP-LPV Systems

In Equation 3-7 we have introduced to canonical form of an MP-LPV system with explicit system dynamics. For simplicity, let us repeat the state equation of such a system:

$$x(k) = \bigoplus_{\mu=1}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) \oplus B(p^{(1)}(k)) \otimes u(k) \tag{4-3}$$

Let us formally propose the condition for unique solvability of explicit MP-LPV systems:

Proposition 4.1. *The state equation defined in Equation 4-3 of the explicit MP-LPV system is considered uniquely solvable if Assumption 3.1 holds.*

Proof. Let us prove this using direct proof. In order to obtain the (explicit) canonical form of the MP-LPV system, we have seen that Assumption 3.1 needs to hold. In this canonical form, we have that the right hand side only contains known variables. The state equation therefore always has a solution $x(k)$ which can be calculated directly and this solution will automatically be unique. Likewise, if $x(k)$ exists, the state thereafter $(x(k+1))$ will also exist, as this will be built from $x(k)$ and so on. □

4-3 The Solvability Problem of Single Implicit MP-LPV Systems

To investigate how we can extend the method to resolve the implicitness in MPL systems to a solvability framework for single implicit systems, let us recall Lemma 2.1:

Recall: Lemma 2.1. *Let $A \in \mathbb{R}_\varepsilon^{n\times n}$ be a matrix of which the communication graph $\mathcal{G}(A)$ only has circuits of non-positive weight. Then the solution x to $x = A \otimes x \oplus b$ is given by:*

$$x = A^* \otimes b \tag{4-4}$$

In which A^* is the Kleene star and vector $b \in \mathbb{R}_\varepsilon^n$. Furthermore, if the circuit weights are negative instead of non-positive, this solution x is unique.

Let us furthermore recall the state-space description of the single implicit MP-LPV system based on Lemma 3.1:

$$\begin{aligned} x(k) &= A_0\left(p^{(1)}(k)\right) \otimes x(k) \oplus \underbrace{\bigoplus_{\mu=1}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) \oplus B(p^{(1)}(k)) \otimes u(k)}_{b(k)} \\ &= A_0\left(p^{(1)}(k)\right) \otimes x(k) \oplus b(k) \\ y(k) &= C(p(k)) \otimes x(k) \\ p^{(\mu)}(k) &= [x^T(k-\mu), \ldots, x^T(k-M), u^T(k), z^T(k)]^T \in \mathbf{P}^{(\mu)} \end{aligned} \tag{4-5}$$

With $z(k) \in \mathbb{R}^z$ an exogenous input that is assumed to be independent of the state $x(\cdot) \in \mathbb{R}^n$ and $u(\cdot) \in \mathbb{R}^m$ the control input. Furthermore, μ is a variable for which holds that $\mu \in \mathbb{Z}_+$ and m is the maximum number of past cycles that are included. Clearly, the parameter vector $p^{(1)}(k)$ is defined such that it contains only previous and therefore known variables. The same holds for $b(k)$. It can be observed that the format of the state equation is equal to Equation 4-4 considered in Lemma 2.1, and we can therefore propose the following:

Proposition 4.2. *The state equation of the single implicit MP-LPV system as defined in Equation 4-5 has a solution $x(k)$ (and thus Definition 4.2 holds) if the Kleene star $A_0^*\left(p^{(1)}(k)\right)$ exists.*

Proof. The proof is based on [12, Theorem 3.17] combined with Lemma 2.1. The proof of [12, Theorem 3.17] provides that for existence of $A_0^*(p^{(1)}(k))$, it is necessary and sufficient that the elementary circuit weights of the communication graph $\mathcal{G}(A_0(p^{(1)}(k))$ are non-positive. Lemma 2.1 states that if the circuit weights are non-positive, the solution to the state equation in Equation 4-5 exists and is given by:

$$x(k) = A_0^*\left(p^{(1)}(k)\right) \otimes \bigoplus_{\mu=1}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) \oplus B(p^{(1)}(k)) \otimes u(k) \tag{4-6}$$

□

This proposition provides the condition necessary for the existence of the solution $x(k)$, given $x(k-1)$. It does not provide the necessary and sufficient proof; it is only necessary and sufficient that the elementary circuit weights have to be non-positive for existence of the Kleene star. However, the existence of the Kleene star is only a necessary condition for existence of the solution as given by Lemma 2.1. We thus aim to find the set $\mathcal{P}$ such that if $p^{(1)}(k) \in \mathcal{P}$, the Kleene star $A_0^*(p^{(1)}(k))$ exists; or equivalently, the set $\mathcal{P}$ such that if $p^{(1)}(k) \in \mathcal{P}$, the elementary circuits are non-positive. If the elementary circuit weights are strictly negative, we can confirm unique existence as defined in Definition 4.3 based on Lemma 2.1. For the property of (unique) solvability, we are interested in conditions that guarantee that a solution $x(k+j)$ remains to exist for all future event steps $k+j$ with $j \in \mathbb{Z}_+$. Equivalently, we aim to find an invariant set $\mathcal{P}_{\text{inv}} \subseteq \mathcal{P}$ for which holds that if $p^{(1)}(k) \in \mathcal{P}_{\text{inv}}$ with $p^{(1)}(k)$ resulting in existence of a solution $x(k)$, then $p^{(1)}(k+j) \in \mathcal{P}_{\text{inv}}$ for all $j \in \mathbb{Z}_+$ resulting in a solution $x(k+j)$. In addition to Proposition 4.2, we propose the following:

Proposition 4.3. *The following conditions are equivalent to the existence of the Kleene star $A_0^*(p^{(1)}(k))$:*

1. *Circuit weights $|p|_w = \bigotimes_{k=1}^{C} a_{i_{k+1}i_k} \leq 0 \ \forall m$, with C the number of nodes that constitute the circuits and $a_{(\cdot)}$ entries of the matrix $A_0(p^{(1)}(k))$;*

2. $\max(\lambda_m(A_0(p^{(1)}(k)))) \leq 0 \ \forall m$;

3. $\lambda_m(A_0(p^{(1)}(k))) \leq 0 \ \forall m$.

In which m all elementary circuits of $A_0(p^{(1)}(k))$. For existence of a unique solution $x(k)$, we require all inequality signs in the above conditions to be strict inequality signs ($<$).

Proof. We have seen in the proof of Proposition 4.2 that a necessary and sufficient condition for existence of the Kleene star $A_0^*(p^{(1)}(k))$ is that the elementary circuit weights of the communication graph must be non-positive. We can therefore confirm the equivalence of the first condition. By [13, Lemma 2.5] we can say that average circuit weights are candidates for eigenvalues. Equivalently we can say that every finite eigenvalue is equal to some average circuit weight. Thus, if all circuit weights are non-positive, then all eigenvalues are non-positive and vice versa, or equivalently the maximum of all eigenvalues is non-positive and vice versa. This confirms the equivalence of the second and third condition.

For uniqueness of the solution, we refer to Lemma 2.1, which states that if the circuit weights are negative instead of non-positive, the solution $x(k)$ is unique. As the three conditions above are equivalent to one another, it is easily confirmed that they are still equivalent for $<$ instead of $\leq$. It remains to show this equivalence extends to Proposition 4.2. This is automatically confirmed by Lemma 2.1, as the unique solution is then defined by $x = A^* \otimes b$ and therefore A^* must be unique. □

In conclusion, in Proposition 4.3 we have seen that in order to investigate the existence of solution $x(k)$ to the state equation of the single implicit MP-LPV system, we can investigate the values of $\lambda(A_0(p^{(1)}(k)))$. Let us start with analyzing the possible $\mathcal{E}$-structures of $A_0(p^{(1)}(k))$, to investigate if we can already make a few statements:

1. A strictly lower triangular, reducible structure with the diagonal entries equal to ε:

$$A_0(p^{(1)}(k)) = \begin{bmatrix} \varepsilon & \dots & \dots & \varepsilon \\ * & \varepsilon & \dots & \varepsilon \\ \vdots & \ddots & \ddots & \vdots \\ * & \dots & * & \varepsilon \end{bmatrix} \tag{4-7}$$

   With the entries $* \in \mathbb{R}_\varepsilon$. Let us propose the following:

   Proposition 4.4. *A single implicit MP-LPV system with an $A_0(p^{(1)}(k))$-structure as in Equation 4-7 for all $k+j$ with $j \geq 0$ is considered to be uniquely solvable.*

   Proof. In [13, Chapter 2] it is stated that in the case where the set of elementary circuits $C(A_0(\cdot))$ is empty, we define the eigenvalue to be $\lambda = \varepsilon$. In this strictly lower triangular structure there are no circuits present and we can therefore define $\lambda(A_0(\cdot)) = \varepsilon < 0$, regardless of the values of the finite entries $* \in \mathbb{R}_\varepsilon$. By Proposition 4.2 and 4.3, we can conclude that a solution $x(k)$ therefore exists and is unique. If this structure is maintained for all future steps $k+j$ with $j \geq 1$, we can conclude that $\lambda(A_0(\cdot)) = \varepsilon$ for all $k+j$, and therefore the Kleene star exists for all $k+j$, such that the single implicit MP-LPV system is uniquely solvable for all $k+j$. □

2. A lower triangular, reducible structure with the diagonal entries possibly finite:

$$A_0(p^{(1)}(k)) = \begin{bmatrix} *_d & \varepsilon & \dots & \varepsilon \\ * & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \varepsilon \\ * & \dots & * & *_d \end{bmatrix} \tag{4-8}$$

With $* \in \mathbb{R}_\varepsilon$ and $*_d \in \mathbb{R}_\varepsilon$. In this structure, the set of elementary circuits $C(A_0(\cdot))$ consists out of all diagonal entries $*_d$ representing the self-loops.

Proposition 4.5. *A single implicit MP-LPV system with an $A_0(p^{(1)}(k))$-structure as in Equation 4-8 has a solution $x(k)$ (and thus Definition 4.2 holds) if $p^{(1)}(k)$ is such that $*_d \leq 0$. If furthermore $*_d < 0$, this solution $x(k)$ is unique (and thus Definition 4.3 holds).*

Proof. If all diagonal entries are non-positive (or negative), the weights of the elementary circuits are non-positive (or negative) and by Proposition 4.2 and 4.3 we can ensure that solution $x(k)$ exists (and is unique). □

If we want to ensure (unique) solvability for this reducible structure with diagonals possibly finite, we again have to find the set $\mathcal{P}_{\text{inv}} \subseteq \mathcal{P}$ such that if $p^{(1)}(k) \in \mathcal{P}_{\text{inv}}$, then $*_d \leq 0$ (or $*_d < 0$ for uniqueness), and furthermore $p^{(1)}(k+j) \in \mathcal{P}_{\text{inv}}$ for all $j \in \mathbb{Z}_+$. The values of the entries $* \in \mathbb{R}_\varepsilon$ can again be disregarded, as they will not be a part of any (elementary) circuit.

3. An irreducible structure:

$$A_0(p^{(1)}(k)) = \begin{bmatrix} * & \dots & * \\ \vdots & \ddots & \vdots \\ * & \dots & * \end{bmatrix} \tag{4-9}$$

With $*$ representing any continuous Piecewise Affine (PWA) function of $p^{(1)}(k)$ as assumed in Assumption 3.1. From [12, Paragraph 3.2.4] we state the following:

Theorem 4.1. *If A is irreducible (and thus communication graph $\mathcal{G}(A)$ is strongly connected), there exists one and only one eigenvalue λ and at least one associated eigenvector v. This eigenvalue λ is equal to the maximum average weight over all elementary circuits.*

From this theorem we know that if $A_0(\cdot)$ has the structure as in Equation 4-9, we have one and only one eigenvalue which is equal to the maximum average weight over all elementary circuits. Thus, in order for this eigenvalue to be non-positive (negative) for existence of a (unique) solution $x(k)$, we again must ensure that the maximum average weight over all elementary circuits is non-positive (negative). We can ensure this by having all entries $* \leq 0$ (or $* < 0$ for uniqueness), but this condition is stricter than necessary. We will therefore approach the solvability problem for the irreducible structure in a elaborate manner in the following section.

These different structures show that depending on the structure of the system, different conditions are necessary to guarantee existence of the Kleene star of $A_0(\cdot)$ which ensures existence of solution $x(k)$. However, for both the reducible as the irreducible case, we need to investigate which parameter vectors ensure (unique) solvability is guaranteed. In the following section, we will investigate how to obtain such a parameter set.

4-4 (Unique) Solvability of Single Implicit MP-LPV Systems

In this section, we will approach the (unique) existence and (unique) solvability problem of the single implicit MP-LPV system in a set theoretical manner. This approach will be suitable for as well the reducible structure, as the irreducible structure. The goal is to find the maximal positive invariant set $\mathcal{P}_{\text{inv}}$ for which is ensured that if $p^{(1)}(k) \in \mathcal{P}_{\text{inv}}$ results in the existence of solution $x(k)$, then also $p^{(1)}(k+j) \in \mathcal{P}_{\text{inv}}$ ensures the existence of solution $x(k+j)$ for all $j \in \mathbb{Z}_+$. Let us first define a (positive) invariant set:

Definition 4.6 ((Positive) Invariant Set)**.** *A set $\mathcal{O}$ is an* invariant set *with respect to the (autonomous) system $x(k+1) = f(x(k))$ with $x(k) \in \mathcal{X}$ if the following holds:*

$$\mathcal{O} \subseteq \mathcal{X}; \quad x(0) \in \mathcal{O} \Rightarrow x(k) \in \mathcal{O} \quad \forall\, k \in \mathbb{R} \tag{4-10}$$

And therefore $x(k) \in \mathcal{X}$ for all $k \in \mathbb{R}$. If the above holds solely for all $k \in \mathbb{Z}_+$, the set $\mathcal{O}$ is a positive invariant set.

In other words, the positive invariant set $\mathcal{O}$ contains all solutions for which is ensured that all future solutions also are contained in $\mathcal{O}$.

Definition 4.7 (Maximal Positive Invariant Set)**.** *The set $\mathcal{O}_\infty$ is the* maximal positive invariant set *of the system $x(k+1) = f(x(k))$ with $x(k) \in \mathcal{X}$ if $\mathcal{O}_\infty$ is positive invariant and $\mathcal{O}_\infty$ contains all positive invariant sets $\mathcal{O} \subseteq \mathcal{X}$. Or equivalently:*

$$\mathcal{O}_\infty = \bigcup \mathcal{O} \tag{4-11}$$

We can find this maximal positive invariant set $\mathcal{P}_{\text{inv}}$ in three steps. First, we need to find the set $\mathcal{P}$, defined as:

$$\mathcal{P} = \left\{ p^{(1)}(k) \mid \lambda_m \left(A_0 \left(p^{(1)}(k) \right) \right) \leq 0 \right\} \tag{4-12}$$

Or equivalently, set $\mathcal{P}$ contains all parameter vectors that ensure that the eigenvalues of the $A_0(\cdot)$ are non-positive. And thus, for all vectors $p^{(1)}(k) \in \mathcal{P}$ existence as defined in Definition 4.2 is guaranteed. If we furthermore have $\lambda_m < 0$, we can confirm unique existence as defined in Definition 4.3. We will first focus on existence and extend this afterwards to unique existence. The next step after obtaining the set $\mathcal{P}$ is to obtain the precursor set $\text{Pre}(\mathcal{P})$, defined as:

$$\text{Pre}(\mathcal{P}) = \left\{ p^{(1)}(k-1) \mid p^{(1)}(k) = f_{\text{cPWA}} \left(p^{(1)}(k-1) \right) \in \mathcal{P} \right\} \tag{4-13}$$

In which f_{cPWA} is a continuous PWA function. Note that because of Assumption 3.1 in which is assumed that the entries of $A_0(\cdot)$ are continuous PWA functions of the parameter vectors, we require $p^{(1)}(k)$ to be a continuous PWA function of $p^{(1)}(k-1)$. This precursor set $\text{Pre}(\mathcal{P})$ represents the set of parameter vectors that evolve to the target set $\mathcal{P}$ in one event step. Clearly, we have $\text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$. The idea of the target set $\mathcal{P}$ and precursor set $\text{Pre}(\mathcal{P})$ is visualized in Figure 4-2.

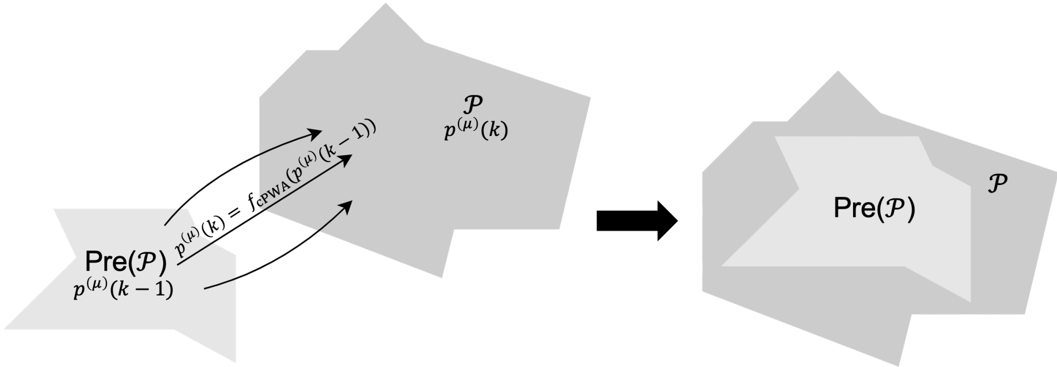


Figure 4-2: We compute the precursor set Pre($\mathcal{P}$) of the target set $\mathcal{P}$ for which holds that if $p^{(\mu)}(k-1)$ ∈Pre($\mathcal{P}$), then $p^{(\mu)}(k) \in \mathcal{P}$, as shown on the left. The resulting precursor set Pre($\mathcal{P}$) will then be a subset of the target set $\mathcal{P}$, as shown on the right.

Finally, we want to find the maximal positive invariant set $\mathcal{P}_{\text{inv}}$ with respect to the set $\mathcal{P}$ and therefore sequentially of $\text{Pre}(\mathcal{P})$, for which holds that:

$$\mathcal{P}_{\text{inv}} = \left\{ p^{(1)}(k) \mid p^{(1)}(k+j) \in \mathcal{P}_{\text{inv}} \ \forall \ j \in \mathbb{Z}_+ \right\} \tag{4-14}$$

And by definition of a maximal positive invariant set we then also know that $p^{(1)}(k) \in \mathcal{P}$, and therefore also $p^{(1)}(k+j) \in \mathcal{P}$ for all $j \in \mathbb{Z}_+$. We then have:

$$\mathcal{P}_{\text{inv}} \subseteq \text{Pre}(\mathcal{P}) \subseteq \mathcal{P} \tag{4-15}$$

The concept of the invariant set $\mathcal{P}_{\text{inv}}$ is shown in Figure 4-3.

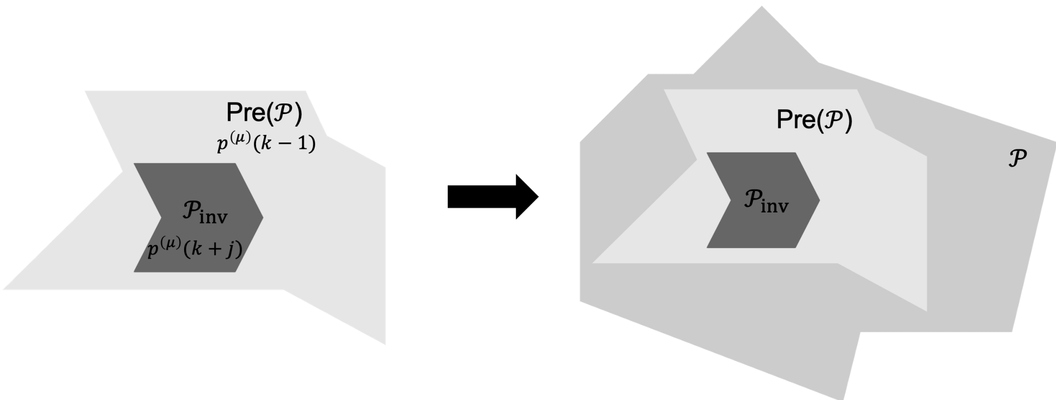


Figure 4-3: We compute the invariant set $\mathcal{P}_{\text{inv}}$ with respect to Pre($\mathcal{P}$) on the left. As the precursor set Pre($\mathcal{P}$) is a subset of the target set $\mathcal{P}$, $\mathcal{P}_{\text{inv}}$ is also invariant with respect to $\mathcal{P}$, as shown on the right.

Let us combine all this in a formal proposition:

Proposition 4.6. *Existence of a solution to the state equation of the single implicit MP-LPV system (with any A_0-structure) can be confirmed if there exist nonempty sets Pre($\mathcal{P}$) and*

$\mathcal{P}$ with Pre($\mathcal{P}$) $\subseteq \mathcal{P}$ as defined in Equation 4-13 and 4-12 respectively. The single implicit MP-LPV system (with any A_0-structure) is considered solvable if we can also find a nonempty maximal positive invariant subset $\mathcal{P}_{inv}$ with $\mathcal{P}_{inv} \subseteq$ Pre($\mathcal{P}$) $\subseteq \mathcal{P}$, defined in Equation 4-14.

Proof. Let us prove this using direct proof. For this, let us assume that $\mathcal{P}_{\text{inv}} \subseteq \text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$ exist and are nonempty. By definition of a subset $\subseteq$, we know that if $p^{(1)}(k) \in \mathcal{P}_{\text{inv}}$, then $p^{(1)}(k) \in \text{Pre}(\mathcal{P})$ and also $p^{(1)}(k) \in \mathcal{P}$. Furthermore, by definition of the set $\mathcal{P}_{\text{inv}}$, we know that if $p^{(1)}(k) \in \mathcal{P}_{\text{inv}}$, then all future $p^{(1)}(k+j) \in \mathcal{P}_{\text{inv}}$ for all $j > 0$. And, by definition of set $\mathcal{P}$ we know that if $p^{(1)}(k) \in \mathcal{P}$, we have that $\lambda_m(A_0(p^{(1)}(k))) \leq 0$ for all elementary circuits m. By Proposition 4.2 and 4.3, we know that if all eigenvalues of the elementary circuits of the $A_0(\cdot)$ matrix are non-negative, existence of a solution is ensured, regardless of the structure of $A_0(\cdot)$. By the definition of a maximal invariant set, we know that therefore existence of all future solutions $x(k+j)$ is ensured. By the definition of solvability in Definition 4.4, we can therefore confirm solvability of the single implicit MP-LPV system with any A_0-structure. □

In the following subsections, we show how we can obtain these sets for the single implicit MP-LPV system. We will also see that these results can easily be extended to the properties of unique existence and unique solvability.

4-4-1 Computing $\mathcal{P}$

In Equation 4-12, we have seen how the target set $\mathcal{P}$ is defined. This target set $\mathcal{P}$ is introduced as the set that contains all parameter vectors $p^{(1)}(k)$ that ensure that the Kleene star $A_0^*(p^{(1)}(k))$ and therefore the solution $x(k)$ to the state equation of the single implicit MP-LPV system exists, ensuring that Definition 4.2 holds. In this subsection, we will define an approach to obtain this set. Let us first focus on existence only. These results will thereafter be extended to unique existence.

We have seen in Proposition 4.2 and 4.3 that the Kleene star exists if all elementary circuit weights are non-positive, or equivalently if $\lambda(A_0(p^{(1)}(k))) \leq 0$. Let us define the following matrix:

$$\Gamma\left(A_0(p^{(1)}(k)\right) = \bigoplus_{i=1}^{n-1} A_0^{\otimes i}\left(p^{(1)}(k)\right) \tag{4-16}$$

The entries of this matrix are the maximum weights of all the circuits with a length less than or equal to n (and thus the elementary circuits). As a path is only a circuit if it starts and ends in the same node, we know that the diagonal entries of $\Gamma(A_0(p^{(1)}(k))$ represent the maximum weights of all the circuits. Therefore, by Proposition 4.3 we can say that if $[\Gamma(A_0(p^{(1)}(k))]_{ii} \leq 0$ for all i, $\lambda(A_0(p^{(1)}(k))) \leq 0$. By [13, Lemma 2.2], we then have:

$$A_0^*\left(p^{(1)}(k)\right) = E \oplus \Gamma\left(A_0(p^{(1)}(k)\right) \tag{4-17}$$

Let us vectorize matrix $A_0(\cdot)$ and $\Gamma(\cdot)$ in the following manner:

$$a\left(A_0(\cdot)\right) = \begin{bmatrix} [A_0(\cdot)]_{11} \\ [A_0(\cdot)]_{12} \\ \vdots \\ [A_0(\cdot)]_{nn} \end{bmatrix} \qquad \gamma\left(\Gamma(\cdot)\right) = \begin{bmatrix} [\Gamma(\cdot)]_{11} \\ [\Gamma(\cdot)]_{12} \\ \vdots \\ [\Gamma(\cdot)]_{nn} \end{bmatrix} \tag{4-18}$$

We want to compute the set $\mathcal{P}$ that contains all $p^{(1)}(k)$ that ensures the following:

$$\gamma_{ii}\left(\Gamma(\cdot)\right) \leq 0 \qquad \forall i = 1, ..., n \tag{4-19}$$

In which $\gamma_{ii}(\Gamma(\cdot))$ represent the diagonal entries $[\Gamma(\cdot)]_{ii}$. By the definition of $\Gamma(\cdot)$ in Equation 4-16, the entries of $\Gamma(\cdot)$ are max-plus equations of the entries of $A_0(\cdot)$. In other words, the entries of vector $\gamma(\cdot)$ are max-plus equations of the entries of vector $a(\cdot)$, as well as $\gamma_{ii}(\cdot)$. Let us visualize this by a 3×3 example:

$$\begin{aligned}
\Gamma\left(A_0(p^{(1)}(k)\right) &= \bigoplus_{i=1}^{2} A_0^{\otimes i}\left(p^{(1)}(k)\right) = A_0(\cdot)^{\otimes 1} \oplus A_0(\cdot)^{\otimes 2} \\
&= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \oplus \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \otimes \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \\
&= \begin{bmatrix} \max(a_{11}, a_{11} + a_{11}, a_{12} + a_{21}, a_{13} + a_{31}) & \dots & \dots \\ \max(a_{21}, a_{21} + a_{11}, a_{22} + a_{21}, a_{23} + a_{31}) & \dots & \dots \\ \max(a_{31}, a_{31} + a_{11}, a_{32} + a_{21}, a_{33} + a_{31}) & \dots & \dots \end{bmatrix}
\end{aligned} \tag{4-20}$$

These entries are clearly max-plus equations of the vector $a(\cdot)$. We can furthermore write these entries as sets of linear inequalities. Let us write this out for the $[\Gamma(A_0(p^{(1)}(k))]_{11}$ entry of the above example. From Equation 4-19 we know that we must have $[\Gamma(A_0(p^{(1)}(k))]_{11} \leq 0$. We can therefore write:

$$[\Gamma(A_0(p^{(1)}(k))]_{11} = \max(a_{11}, a_{11} + a_{11}, a_{12} + a_{21}, a_{13} + a_{31}) \leq 0 \tag{4-21}$$

This can be rewritten as a set of linear inequalities in the following manner:

$$\begin{aligned}
a_{11} &\leq 0 \\
a_{11} + a_{11} &\leq 0 \\
a_{12} + a_{21} &\leq 0 \\
a_{13} + a_{31} &\leq 0
\end{aligned} \tag{4-22}$$

From this example, we have therefore observed that we can rewrite the condition in Equation 4-19 as a set of linear inequalities:[2]

$$\gamma_i(\cdot) \leq 0 \quad \forall i = 1, ..., n \;\Leftrightarrow\; \mathcal{A}_a = \{a(\cdot) \mid H_a a(\cdot) \leq 0\} \tag{4-23}$$

In which $H_a \in \mathbb{R}^{n_{H_a} \times n^2}$ a tall matrix. The set $\mathcal{A}_a$ contains all vectors a such that the entries of vector γ are non-positive and therefore all vectors a such that the Kleene star $A_0^*(\cdot)$ exists and $\lambda(A_0(\cdot)) \leq 0$. In Assumption 3.1 we have assumed all finite entries of $A_0(\cdot)$ to be continuous PWA functions of the parameter vector. We can therefore write the entries of vector $a(\cdot)$ as continuous PWA functions in the following manner:

$$a\left(p^{(1)}(k)\right) = G_j p^{(1)}(k) + g_j, \quad \forall p^{(1)}(k) \in \mathcal{T}_j \tag{4-24}$$

[2]The resulting set $\mathcal{A}_a$ is a polyhedral cone.

With $\mathcal{T}_j = \{p^{(1)}(k) \mid T_j p^{(1)}(k) \leq t_j\}$ non-overlapping convex polyhedra, $G_j \in \mathbb{R}^{n^2 \times n_{p^{(1)}}}$, $g_j \in \mathbb{R}^{n^2}$, $T_j \in \mathbb{R}^{n_{\mathcal{T}} \times n_{p^{(1)}}}$, $t_j \in \mathbb{R}^{n_{\mathcal{T}}}$ for $j = 1, ..., n_{\mathcal{T}}$. Equivalently, after substituting $a(\cdot)$ in Equation 4-23, we can construct the following sets:

$$\mathcal{P}_j = \left\{ p^{(1)}(k) \;\middle|\; \begin{bmatrix} H_a G_j \\ T_j \end{bmatrix} p^{(1)}(k) \leq \begin{bmatrix} -H_a g_j \\ t_j \end{bmatrix} \right\}, \quad j = 1, ..., n_{\mathcal{T}} \tag{4-25}$$

With $\mathcal{P}_j$ convex for all j. Computing the union of these sets results in the set that contains all parameter vectors $p^{(1)}(k)$ that ensure that $\lambda(A_0(\cdot)) \leq 0$ and therefore existence of a solution $x(k)$:

$$\mathcal{P} = \bigcup_{j=1}^{n_{\mathcal{T}}} \mathcal{P}_j \tag{4-26}$$

The resulting set $\mathcal{P}$ will in general not be convex. We can extend these results to obtaining a set $\mathcal{P}$ that ensures existence of a *unique* solution. Let us formulate a formal proposition:

Proposition 4.7. *The set $\mathcal{P}$ defined in Equation 4-26 contains all parameter vectors $p^{(1)}(k)$ that ensure existence of a unique solution $x(k)$ of the single implicit MP-LPV system, if the inequality signs in Equation 4-25, 4-23 and 4-19 are replaced with strictly inequality signs $<$.*

Proof. Let us prove this using direct proof. For this, we assume that we indeed have strict inequality signs in the concerning equations. By Equation 4-19 we then know that all diagonal entries of $\Gamma(\cdot)$ are strictly negative, and thus the elementary circuit weights are strictly negative. By Proposition 4.2 we can then confirm that the single implicit MP-LPV system indeed has a unique solution $x(k)$. □

4-4-2 Computing Pre $(\mathcal{P}) \subseteq \mathcal{P}$

Now that we have obtained the target set $\mathcal{P}$, the next step is to investigate how we can obtain the precursor set Pre($\mathcal{P}$), as defined in Equation 4-13. This precursor set will be constructed in such a manner that it contains all parameter vectors that evolve to the target set $\mathcal{P}$ in one event step, or equivalently $p^{(1)}(k-1) \in \text{Pre}(\mathcal{P})$ ensures that $p^{(1)}(k) \in \mathcal{P}$. In Equation 4-13 we have defined the parameter vector $p^{(1)}(k)$ as a continuous PWA function of $p^{(1)}(k-1)$. This can be confirmed by the following reasoning. Recall that the parameter vector $p^{(1)}(k)$ is defined such that it contains $x(k-1)$. Furthermore, we know that $x(k-1)$ is a max-plus equation of the entries of $A_0(p^{(1)}(k-1))$, which we can conclude from the 3×3 example in the previous subsection. As the entries of $A_0(p^{(1)}(k-1))$ are continuous PWA functions of $p^{(1)}(k-1)$, we therefore know that $p^{(1)}(k)$ is a continuous PWA function of $p^{(1)}(k-1)$. We can therefore write $f_{\text{cPWA}}(p^{(1)}(k-1))$ in the following manner:

$$f_{\text{cPWA}}\left(p^{(1)}(k-1)\right) = p^{(1)}(k) = D_l p^{(1)}(k-1) + d_l \quad \forall p^{(1)}(k-1) \in \mathcal{S}_l \tag{4-27}$$

With non-overlapping convex polyhedra $\mathcal{S}_l = \{p^{(1)}(k-1) \mid S_l p^{(1)}(k-1) \leq s_l\}$, matrices $D_l \in \mathbb{R}^{n_{p^{(1)}} \times n_{p^{(1)}}}$ and $S_l \in \mathbb{R}^{n_{\mathcal{S}} \times n_{p^{(1)}}}$ and vectors $d_l \in \mathbb{R}^{n_{p^{(1)}}}$ and $s_l \in \mathbb{R}^{n_{p^{(1)}}}$ for $l = 1, ..., n_{\mathcal{S}}$. We can follow a similar procedure as in the previous subsection, and after substitution of Equation 4-27 in Equation 4-25, we can construct the following sets:

$$\chi_{j,l} = \left\{ p^{(1)}(k-1) \;\middle|\; \begin{bmatrix} H_a G_j D_l \\ T_j D_l \\ S_l \end{bmatrix} p^{(1)}(k-1) \leq \begin{bmatrix} -H_a g_j - H_a G_j d_l \\ t_j - T_j d_l \\ s_l \end{bmatrix} \right\}, \quad \forall j, l \tag{4-28}$$

With $j = 1, ..., n_{\mathcal{T}}$ and $l = 1, ..., n_{\mathcal{S}}$. Thus in summary, if $p^{(1)}(k-1) \in \chi_{j,l}$, then we know that $p^{(1)}(k-1) \in \mathcal{S}_l$ and therefore $f_{\text{cPWA}}(p^{(1)}(k-1)) = p^{(1)}(k)$ which composes the precursor set $\text{Pre}(\mathcal{P})_{j,l}$. We then have found all $p^{(1)}(k-1) \in \text{Pre}(\mathcal{P})_{j,l}$ such that $f_{\text{cPWA}}(p^{(1)}(k-1)) = p^{(1)}(k) \in \mathcal{P}_j$. As $\chi_{j,l}$ represent $(n_{\mathcal{T}} \cdot n_{\mathcal{S}})$ number of sets, we want to find the union of all these sets to find $\text{Pre}(\mathcal{P})$:

$$\text{Pre}(\mathcal{P}) = \bigcup_{l=1}^{n_{\mathcal{S}}} \bigcup_{j=1}^{n_{\mathcal{T}}} \chi_{j,l} \subseteq \mathcal{P} \tag{4-29}$$

Again, the resulting precursor set $\text{Pre}(\mathcal{P})$ will in general be not convex. Note that we again can extend the above result to ensure existence of a unique solution $x(k)$:

Proposition 4.8. *The set Pre($\mathcal{P}$) contains all parameter vectors $p^{(1)}(k-1)$ that ensure that $p^{(1)}(k) \in \mathcal{P}$, with $\mathcal{P}$ the set of parameter vectors, such that existence of a unique solution as defined in Definition 4.3 of the single implicit MP-LPV system is ensured, if the inequality signs discussed in Proposition 4.7 and in Equation 4-28 are replaced with $<$.*

Proof. As Equation 4-28 is obtained through substitution, replacing the inequality sign in this equation with $<$ is an inevitable result of replacing the inequality signs with $<$ in the equations discussed in Proposition 4.7. □

4-4-3 Computing $\mathcal{P}_{\text{inv}} \subseteq \text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$

Now that we have found the sets $\mathcal{P}$ and $\text{Pre}(\mathcal{P})$ with $\text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$, it remains to find the maximal positive invariant set $\mathcal{P}_{\text{inv}} \subseteq \mathcal{P}$, formally defined in Equation 4-14. Recall that this set will ensure that if $p^{(1)}(k)$ results in a solution $x(k)$, all future parameter vectors $p^{(1)}(k+j)$ will result in a solution $x(k+j)$ with $j \in \mathbb{Z}_+$. In finding $\mathcal{P}_{\text{inv}}$, let us first state the following theorem based on [34, Theorem 10.1]:

Theorem 4.2 (Geometric Condition for Invariance)**.** *A set $\mathcal{O} \subseteq \mathcal{X}$ is a positive invariant set with respect to the system $x(k+1) = f(x(k))$ with $x(k) \in \mathcal{X}$ if and only if the following holds:*

$$\mathcal{O} \subseteq \text{Pre}(\mathcal{O}) \tag{4-30}$$

Or equivalently:

$$\text{Pre}(\mathcal{O}) \cap \mathcal{O} = \mathcal{O} \tag{4-31}$$

We have already seen that both the target set $\mathcal{P}$ and $\text{Pre}(\mathcal{P})$ are both the unions of multiple sets $\mathcal{P}_j$ and $\chi_{j,l}$ respectively, and that the resulting unions generally are nonconvex. For understanding purposes, we have included an example situation in Figure 4-4, when computing the sets. It can be observed that in this example case we still have $\mathcal{P}_{\text{inv}} \subseteq \text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$.

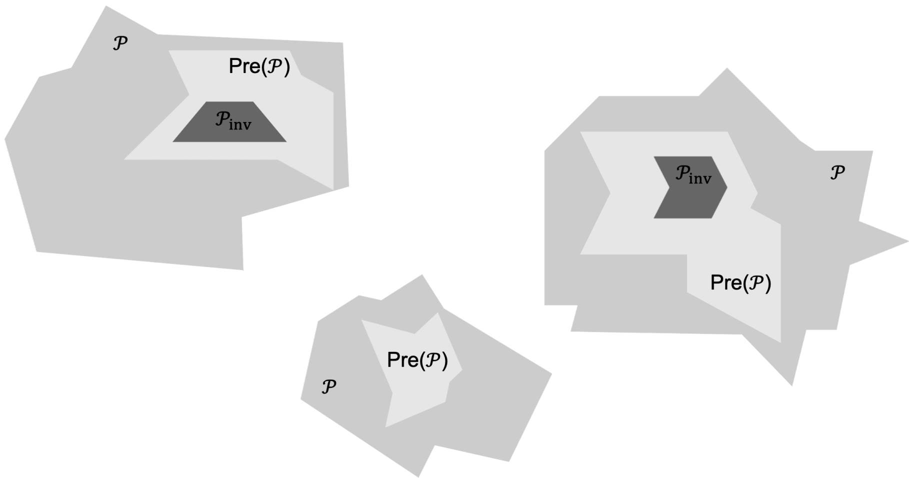


Figure 4-4: The union sets $\mathcal{P}$, Pre($\mathcal{P}$ and $\mathcal{P}_{\text{inv}}$ are generally nonconvex. This figure shows an example situation of the resulting sets.

Based on Theorem 4.2, the definition of an invariant set as defined in Definition 4.6 and observing Figure 4-4, we can state the following equivalences:

$$\begin{aligned} &\mathcal{P}_{\text{inv}} \cap \mathcal{P} = \mathcal{P}_{\text{inv}} \\ &\mathcal{P}_{\text{inv}} \cap \text{Pre}(\mathcal{P}) = \mathcal{P}_{\text{inv}} \\ &\mathcal{P}_{\text{inv}} \cap \text{Pre}(\mathcal{P}) \cap \mathcal{P} = \mathcal{P}_{\text{inv}} \end{aligned} \tag{4-32}$$

These equivalences make sense as the definition of an invariant set states that if the initial solution is in $\mathcal{P}_{\text{inv}}$, then all subsequent solutions are in $\mathcal{P}_{\text{inv}}$. Furthermore, based on Theorem 4.2, [34] provides an algorithm for finding the maximal positive invariant set. Adjusting this algorithm for the single implicit MP-LPV system, we obtain:

Algorithm 1 Computation of $\mathcal{P}_{\text{inv}}$

Input: $f_{\text{cPWA}}(\cdot)$, $\mathcal{P}$
Output: $\mathcal{P}_{\text{inv}}$
    $\Omega_0 \leftarrow \mathcal{P}$, $k \leftarrow -1$
    Repeat
        $k \leftarrow k + 1$
        $\Omega_{k+1} \leftarrow \text{Pre}(\Omega_k) \cap \Omega_k$
    Until $\Omega_{k+1} = \Omega_k$
    $\mathcal{P}_{\text{inv}} \leftarrow \Omega_k$

The book [34] furthermore introduces the concept of control invariant sets. Let us properly define such a set as:

Definition 4.8 (Control Invariant Set)**.** *A set $\mathcal{C} \subseteq \mathcal{X}$ is said to be a control invariant set for a non-autonomous system $x(k+1) = f(x(k), u(k))$ with $x(k) \in \mathcal{X}$ and $u(k) \in \mathcal{U}$ if the following holds:*

$$x(k) \in \mathcal{C} \quad \Rightarrow \quad \exists u(k) \in \mathcal{U} \text{ such that } f(x(k), u(k)) \in \mathcal{C}, \quad \forall k \in \mathbb{Z}_+ \tag{4-33}$$

Such a set can be interesting to obtain for MP-LPV systems in the following manner; if we can find a nonempty control invariant set, we can find the control effort that steers the state trajectories of the MP-LPV system such that the states will remain in the target set $\mathcal{P}$. If we for example have an empty invariant set $\mathcal{P}_{\text{inv}}$, the MP-LPV system is not solvable. But if there does exist a nonempty control invariant set, we can steer the system such that the non-autonomous MP-LPV system ís solvable. Finding such a set however can be a complex procedure. The aim is to find the hull of the projection of the target set $\mathcal{P}$, which will contain the parameter vectors that form the control invariant set. For this, future research is necessary.

In conclusion, if we can find the sets $\mathcal{P}_{\text{inv}} \subseteq \text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$ following the above procedures and it is provided that these sets are all nonempty, we can ensure solvability of the single implicit MP-LPV system. Furthermore, if we want to ensure unique solvability, we just need to ensure that the inequality signs $\leq$ in the necessary equations as discussed in Proposition 4.7 and 4.8 are all replaced with strict inequality signs $<$. If the target set $\mathcal{P}$ is empty, the precursor set $\text{Pre}(\mathcal{P})$ and invariant set $\mathcal{P}_{\text{inv}}$ are automatically empty as well, and we cannot guarantee (unique) solvability nor (unique) existence. If $\mathcal{P}$ and $\text{Pre}(\mathcal{P})$ are nonempty, but the invariant set $\mathcal{P}_{\text{inv}}$ is empty, only (unique) existence can be guaranteed. Note that finding nonempty sets $\mathcal{P}_{\text{inv}} \subseteq \text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$ only provide a necessary condition for (unique) existence and solvability, not a necessary and sufficient condition. This can be explained by the following; the condition in the target set $\mathcal{P}$ is based on Lemma 2.1, but this Lemma however provides only a necessary condition for existence of a (unique) solution.

4-5 The Solvability Problem of Doubly Implicit MP-LPV Systems

In section 4-1 and the previous chapter, we have seen that we cannot use the same condition in the target set on existence of the Kleene star as we used for single implicit MP-LPV systems in order to construct a (unique) existence and solvability framework. The main reason for this is that Proposition 4.2 does not hold for doubly implicit MP-LPV systems, and therefore neither the subsequent propositions 4.3, 4.4 and 4.5. Let us formally introduce the following proposition to demonstrate this:

Proposition 4.9. *Proposition 4.2 until 4.5 do not hold for doubly implicit MP-LPV systems.*

Proof. Let us prove this proposition using proof by counterexample to disprove Proposition 4.2 using a one-by-one case of the doubly implicit MP-LPV system. Subsequently it will follow that Proposition 4.3 then also does not hold. Thereafter, we will show with the use of an example that Proposition 4.4 and 4.5 do not hold. For the one-by-one case of the doubly implicit MP-LPV system, the state equation is defined as:

$$\begin{aligned} x(k) &= A_0\left(p^{(0)}(k)\right) \otimes x(k) \oplus \underbrace{\bigoplus_{\mu=1}^{M}\left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) \oplus B(p^{(1)}(k)) \otimes u(k)}_{b(k)} \\ &= A_0\left(p^{(0)}(k)\right) \otimes x(k) \oplus b(k) \\ &= \max\left(a_0\left(p^{(0)}(k)\right) + x(k), b(k)\right) \end{aligned} \tag{4-34}$$

We have assumed in Assumption 3.1 that the entries of the $A_0(p^{(0)}(k))$ matrix are continuous PWA functions of the parameter vector $p^{(0)}(k)$. We have furthermore seen in subsection 2-5-4 that continuous PWA functions and Max-Min-Plus-Scaling (MMPS) functions are equivalent. We can therefore write the following for $a_0(\cdot)$:

$$a_0\left(p^{(0)}(k)\right) = \max_{l\in\underline{L}}\left(\min_{m\in\underline{n}_l}\left(\alpha_{(l,m)}x(k) + \beta_{(l,m)}\left(p^{(1)}(k)\right)\right)\right) \tag{4-35}$$

For simplicity, let us take $\underline{L} = \underline{n}_1 = 1$. We can then write:

$$x(k) = \max\left(\alpha x(k) + \beta + x(k), b(k)\right) \tag{4-36}$$

From Proposition 4.2 and 4.3 we know that in order for the Kleene star $A_0^*(\cdot)$ to exist, it is required that the weights of the elementary circuits of $A_0(\cdot)$ are non-positive. For the one-by-one case, this means that the following must hold:

$$a_0\left(p^{(0)}(k)\right) = \alpha x(k) + \beta \leq 0 \tag{4-37}$$

If this is not the case, we can already confirm that no solution exists because of the following:

$$\alpha x(k) + \beta \geq 0 \quad \Rightarrow \quad \alpha x(k) + \beta + x(k) \geq x(k) \tag{4-38}$$

In this case clearly Equation 4-36 cannot have a solution. Therefore, it is required that $\alpha x(k) + \beta \leq 0$. This is the case for:

$$\begin{aligned} x(k) &\leq -\frac{\beta}{\alpha} \quad \text{for } \alpha \geq 0 \qquad (1) \\ x(k) &\geq -\frac{\beta}{\alpha} \quad \text{for } \alpha \leq 0 \qquad (2) \end{aligned} \tag{4-39}$$

Therefore, the solution set is bounded by conditions (1) and (2). But then automatically $b(k)$ is also bounded to those conditions. Let us prove this by contradiction. We will first look into case (1) where $\alpha \geq 0$. Let us assume that $b(k) > -\frac{\beta}{\alpha}$ (in contradiction to condition (1)):

$$\begin{aligned} x(k) &= \max\Big(\underbrace{\underbrace{\alpha x(k) + \beta}_{\leq 0} + \underbrace{x(k)}_{\leq -\frac{\beta}{\alpha}}}_{\leq -\frac{\beta}{\alpha}}, \underbrace{b(k)}_{> -\frac{\beta}{\alpha}}\Big) \\ &= b(k) > -\frac{\beta}{\alpha} \end{aligned} \tag{4-40}$$

Clearly this is different than our first assumption that $x(k) \leq -\frac{\beta}{\alpha}$ and it is therefore not possible that $b(k) > -\frac{\beta}{\alpha}$. The same reasoning can be implemented for case (2) where $\alpha \leq 0$. In conclusion, for the one-by-one case we require the following:

$$\begin{aligned} x(k) \wedge b(k) &\leq -\frac{\beta}{\alpha} \quad \text{for } \alpha \geq 0 \qquad (1) \\ x(k) \wedge b(k) &\geq -\frac{\beta}{\alpha} \quad \text{for } \alpha \leq 0 \qquad (2) \end{aligned} \tag{4-41}$$

We can therefore conclude that for doubly implicit MP-LPV systems, more conditions are necessary than solely the existence of the Kleene star. The number of these conditions will increase for increasing order n of the system. As Proposition 4.3 is based on Proposition 4.2, Proposition 4.3 automatically does not hold for doubly implicit MP-LPV systems. Therefore, propositions 4.2 and 4.3 both do not hold for doubly implicit MP-LPV systems.

The above result shows that we cannot use the condition of the existence of the Kleene star $A_0(p^{(0)}(k))$ as a requirement for the target set that contains all parameter vectors $p^{(0)}(k)$ that ensure that the solution $x(k)$ to the state equation of the doubly implicit MP-LPV system exists. It remains to prove that Proposition 4.4 and 4.5 do not hold for doubly implicit MP-LPV systems, and thus that analyzing different structures for the A_0 matrix (as we did for the single implicit MP-LPV system) does not provide us with any simplifications or interim results. Let us prove this using proof by counter example again, observing a doubly implicit MP-LPV system with a strictly reducible A_0 matrix:

Example 4.1. Consider a doubly implicit MP-LPV system with a strictly reducible A_0 matrix:

$$x(k) = \begin{bmatrix} \varepsilon & \dots & \dots & \varepsilon \\ a_{21}(\cdot) & \varepsilon & \dots & \varepsilon \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1}(\cdot) & \dots & a_{n(n-1)}(\cdot) & \varepsilon \end{bmatrix} \otimes x(k) \oplus b(k) \tag{4-42}$$

In which $b(k) = \bigoplus_{\mu=1}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu) \right) \oplus B(p^{(1)}(k)) \otimes u(k)$. By Assumption 3.1 and the canonical form of doubly implicit MP-LPV systems, we know that the finite entries are continuous PWA in the parameter vector $p^{(0)}(k)$. Let us observe state $x_2(k)$. We can write the following:

$$\begin{aligned} x_2(k) &= a_{21}(\cdot) \otimes x_1(k) \oplus b(k) \\ &= f_{\text{cPWA}} \left(p^{(0)}(k) \right) \otimes x_1(k) \oplus b(k) \\ &= \max \left(f_{\text{cPWA}} \left(p^{(0)}(k) \right) + x_1(k), b(k) \right) \end{aligned} \tag{4-43}$$

By definition of the parameter vector $p^{(0)}(k)$, $f_{\text{cPWA}}(p^{(0)}(k))$ can therefore contain the states $x_2(k), x_3(k)$ and so on. Therefore, the state equation still contains implicitness that cannot be removed because of the absence of the maximization operation.

The same reasoning can be extended to doubly implicit systems with a (not strictly) reducible A_0 matrix. We can therefore conclude that Propositions 4.2 until 4.5 do not hold for doubly implicit MP-LPV systems. □

We therefore aim to find conditions suitable for the target set which suffice for existence (and uniqueness) of a solution, regardless of the structure of the A_0 matrix. We will obtain these conditions in the following sections.

4-6 (Unique) Solvability of Doubly Implicit MP-LPV Systems

Let us approach the solvability problem of the doubly implicit MP-LPV system in a similar manner as we did for the single implicit MP-LPV system. We will again compute a target set for the doubly implicit MP-LPV system, here referred to as Λ. We will introduce a condition different from the Kleene star condition used for the target set of single implicit

MP-LPV systems, as defined in Equation 4-12. Thereafter we will follow the same procedure to construct the precursor set Pre(Λ)$\subseteq \Lambda$ and the invariant set $\Lambda_{\text{inv}} \subseteq \text{Pre}(\Lambda) \subseteq \Lambda$. Recall Figures 4-2 and 4-3 for visualization of the procedure. We will again provide conditions to ensure unique existence and unique solvability.

4-6-1 Computing Λ

Let us recall this state equation:

$$\begin{aligned} x(k) &= A_0\left(p^{(0)}(k)\right) \otimes x(k) \oplus \underbrace{\bigoplus_{\mu=1}^{M} \left(A_\mu(p^{(\mu)}(k)) \otimes x(k-\mu)\right) \oplus B(p^{(1)}(k)) \otimes u(k)}_{b(k)} \\ &= A_0\left(p^{(0)}(k)\right) \otimes x(k) \oplus b(k) \end{aligned} \tag{4-44}$$

As we have assumed in Assumption 3.1 that the (finite) entries of the $A_0(p^{(0)}(k))$ matrix are continuous PWA functions of the parameter vector $p^{(0)}(k)$ and as the state equation itself is a max-plus function, we can write the following:

$$x(k) = f_{\text{cPWA}}\left(p^{(0)}(k)\right) \tag{4-45}$$

By definition of the parameter vector, we can write $p^{(0)}(k)$ as:

$$p^{(0)}(k) = \begin{bmatrix} x(k) \\ x(k-1) \\ \vdots \\ x(k-M) \end{bmatrix} = \begin{bmatrix} x(k) \\ p^{(1)}(k) \end{bmatrix} \in \mathbb{R}^{n_{p^{(0)}}},\ n_{p^{(0)}} \geq n \tag{4-46}$$

We can ensure existence of the solution $x(k)$ if we introduce the function $F(p^{(0)}(k))$ and can ensure the following:

$$F\left(p^{(0)}(k)\right) = f_{\text{cPWA}}\left(p^{(0)}(k)\right) - x(k) = 0 \tag{4-47}$$

Clearly, $F(p^{(0)}(k))$ is again a continuous PWA function. We can therefore write:

$$F\left(p^{(0)}(k)\right) = E_i p^{(0)}(k) + e_i, \quad \forall p^{(0)}(k) \in \mathcal{W}_i \tag{4-48}$$

With $\mathcal{W}_i = \{p^{(0)}(k) \mid W_i p^{(0)}(k) \leq w_i\}$ non-overlapping convex polyhedra, $E_i \in \mathbb{R}^{n \times n_{p^{(0)}}}$, $e_i \in \mathbb{R}^n$, $W_i \in \mathbb{R}^{n_\mathcal{W} \times n_{p^{(0)}}}$ and $w_i \in \mathbb{R}^{n_\mathcal{W}}$ for $i = 1, ..., n_\mathcal{W}$. As we can write $p^{(0)}(k)$ as in Equation 4-46, we can split up matrices E_i and W_i in the following manner:

$$F\left(p^{(0)}(k)\right) = E_{i,0} x(k) + E_{i,1} p^{(1)}(k) + e_i, \quad \forall p^{(0)}(k) \in \mathcal{W}_i \tag{4-49}$$

With $\mathcal{W}_i$ written as:

$$\mathcal{W}_i = \left\{p^{(0)}(k) \mid W_{i,0} x(k) + W_{i,1} p^{(1)}(k) \leq w_i\right\} \tag{4-50}$$

With $E_{i,0} \in \mathbb{R}^{n \times n}$, $E_{i,1} \in \mathbb{R}^{n \times \left(n_{p^{(0)}} - n\right)}$, $W_{i,0} \in \mathbb{R}^{n_{\mathcal{W}} \times n}$ and $W_{i,1} \in \mathbb{R}^{n_{\mathcal{W}} \times \left(n_{p^{(0)}} - n\right)}$. As we require $F(p^{(0)}(k)) = 0$, we can write:

$$\begin{aligned} F\left(p^{(0)}(k)\right) &= E_{i,0}x(k) + E_{i,1}p^{(1)}(k) + e_i = 0 \\ E_{i,0}x(k) &= -E_{i,1}p^{(1)}(k) - e_i \end{aligned} \tag{4-51}$$

Matrix $E_{i,0}$ can either be singular or nonsingular. Let us first consider the case where $E_{i,0}$ is singular. Then Equation 4-51 has infinitely many solutions $x(k)$ if the right-hand side of the equation $(-E_{i,1}p^{(1)}(k) - e_i)$ is contained in the vector space spanned by matrix $E_{i,0}$. Thus in that case existence of the solution $x(k)$ is guaranteed, but uniqueness not. If the right-hand side is not contained in the vector space spanned by matrix $E_{i,0}$, no solution $x(k)$ exists. In this thesis, we will only consider the case where $E_{i,0}$ is nonsingular for the target set Λ. For this, let $\mathcal{M}$ with $m \in \{1, ..., n_{\mathcal{M}}\}$ be the set for which $E_{m,0}$ is invertible. Clearly, if $\mathcal{M} = \emptyset$ the matrix $E_{m,0}$ is singular. Therefore, let us assume $\mathcal{M} \neq \emptyset$. We can then find the following solution to Equation 4-51:

$$x(k) = -E_{m,0}^{-1}E_{m,1}p^{(1)}(k) - E_{m,0}^{-1}e_m, \quad m \in \mathcal{M}, \;\; \forall p^{(0)}(k) \in \mathcal{W}_m \tag{4-52}$$

If we substitute Equation 4-52 in $\mathcal{W}_m$ with $m \in \mathcal{M}$, we obtain the target sets Λ_m:

$$\Lambda_m = \left\{p^{(1)}(k) \mid \left(W_{m,1} - W_{m,0}E_{m,0}^{-1}E_{m,1}\right)p^{(1)}(k) \leq w_m + W_{m,0}E_{m,0}^{-1}e_m\right\}, \quad m \in \mathcal{M} \tag{4-53}$$

Computing the union of these sets Λ_m results in the set Λ that contains all parameter vectors $p^{(1)}(k)$ that ensure that Equation 4-51 and 4-52 hold, and therefore existence of the solution $x(k)$:

$$\Lambda = \bigcup_{m=1}^{n_{\mathcal{M}}} \Lambda_m \tag{4-54}$$

In general this set Λ will not be convex. Note furthermore that the sets Λ_m might be overlapping. If that is the case, we cannot confirm uniqueness of the solution. Figure 4-5 shows an example situation of overlapping Λ_m.

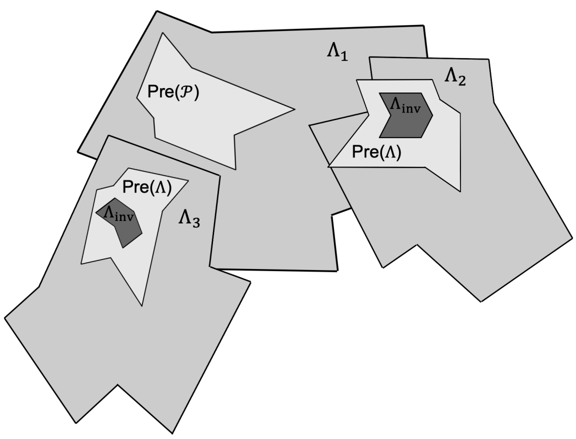


Figure 4-5: An example situation of overlapping sets Λ_m with $m = 1, 2, 3$. In this case, we cannot confirm uniqueness of the state solution.

4-6-2 Computing Pre(Λ) $\subseteq \Lambda$

We define the precursor set Pre(Λ) as the following:

$$\text{Pre}(\Lambda) = \left\{ p^{(1)}(k-1) \mid p^{(1)}(k) = f_{\text{MMPS}}\left(p^{(1)}(k-1)\right) \in \Lambda \right\} \tag{4-55}$$

Similarly as for the single implicit MP-LPV system we can say that $p^{(1)}(k) = f_{\text{cPWA}}(p^{(1)}(k-1))$ as the finite entries of $A_1(p^{(1)}(k-1))$ are continuous PWA functions of $p^{(1)}(k-1)$. We can write $f_{\text{cPWA}}(p^{(1)}(k-1))$ as:

$$f_{\text{cPWA}}\left(p^{(1)}(k-1)\right) = p^{(1)}(k) = C_l p^{(1)}(k-1) + c_l \quad \forall p^{(1)}(k-1) \in \mathcal{O}_l \tag{4-56}$$

With non-overlapping convex polyhedra $\mathcal{O}_l = \{p^{(1)}(k-1) \mid O_l p^{(1)}(k-1) \leq o_l\}$, matrices $C_l \in \mathbb{R}^{n_{p^{(1)}} \times n_{p^{(1)}}}$ and $O_l \in \mathbb{R}^{n_\mathcal{O} \times n_{p^{(1)}}}$ and vectors $c_l \in \mathbb{R}^{n_{p^{(1)}}}$ and $o_l \in \mathbb{R}^{n_{p^{(1)}}}$ for $l = 1, ..., n_\mathcal{O}$. Again, after substituting Equation 4-56 into set Λ_m in Equation 4-53, we obtain:

$$\begin{aligned}\Psi_{m,l} = \Bigg\{ p^{(1)}(k-1) \;\Bigg|\; &\begin{bmatrix} W_{m,1}C_l - W_{m,0}E_{m,0}^{-1}C_l \\ O_l \end{bmatrix} p^{(1)}(k-1) \leq \\ &\begin{bmatrix} w_m + W_{m,0}E_{m,0}^{-1}e_m - W_{m,1}c_l - W_{m,0}E_{m,0}^{-1}c_l \\ o_l \end{bmatrix} \Bigg\}, \quad \forall m, l\end{aligned} \tag{4-57}$$

With $m = 1, ..., n_\mathcal{M}$ and $l = 1, ..., n_\mathcal{O}$. Thus in summary, if $p^{(1)}(k-1) \in \Psi_{m,l}$, then we know that $p^{(1)}(k-1) \in \mathcal{O}_l$ and therefore $f_{\text{cPWA}}(p^{(1)}(k-1)) = p^{(1)}(k)$ which composes the precursor set $\text{Pre}(\Lambda)_{m,l}$. We then have found all $p^{(1)}(k-1) \in \text{Pre}(\Lambda)_{m,l}$ such that $f_{\text{cPWA}}(p^{(1)}(k-1)) = p^{(1)}(k) \in \Lambda_m$. As $\Psi_{m,l}$ represent $(n_\mathcal{M} \cdot n_\mathcal{O})$ number of sets, we want to find the union of all these sets to find Pre(Λ):

$$\text{Pre}(\Lambda) = \bigcup_{l=1}^{n_\mathcal{O}} \bigcup_{m=1}^{n_\mathcal{M}} \Psi_{m,l} \subseteq \mathcal{P} \tag{4-58}$$

Again, the resulting precursor set Pre(Λ) will in general not be convex.

4-6-3 Computing $\Lambda_{\text{inv}} \subseteq$ Pre(Λ) $\subseteq \Lambda$

Finally, it remains to obtain the maximal positive invariant set $\Lambda_{\text{inv}} \subseteq \Lambda$ that ensures that if $p^{(1)}(k)$ results in the unique solution $x(k)$, all future parameter vectors $p^{(1)}(k+j)$ will result in a solution $x(k+j)$ with $j \in \mathbb{Z}_+$. This can be done with the use of Algorithm 1, but then with input Λ:

Algorithm 2 Computation of Λ_{inv}

1: **Input:** $f_{\text{cPWA}}(\cdot)$, Λ
2: **Output:** Λ_{inv}
3: $\quad\Omega_0 \leftarrow \Lambda$, $k \leftarrow -1$
4: $\quad$**Repeat**
5: $\quad\quad k \leftarrow k+1$
6: $\quad\quad \Omega_{k+1} \leftarrow \text{Pre}(\Omega_k) \cap \Omega_k$
7: $\quad$**Until** $\Omega_{k+1} = \Omega_k$
8: $\quad\Lambda_{\text{inv}} \leftarrow \Omega_k$

Similarly as for single implicit systems, we can confirm the following statements, when observing Figure 4-5:

$$\begin{aligned}&\Lambda_{\text{inv}} \cap \Lambda = \Lambda_{\text{inv}}\\&\Lambda_{\text{inv}} \cap \text{Pre}(\Lambda) = \Lambda_{\text{inv}}\\&\Lambda_{\text{inv}} \cap \text{Pre}(\Lambda) \cap \Lambda = \Lambda_{\text{inv}}\end{aligned} \tag{4-59}$$

Equivalently as for single implicit MP-LPV systems, the next step can be to compute the control invariant set as defined in Definition 4.8. In conclusion, if we can find the sets $\Lambda_{\text{inv}} \subseteq \text{Pre}(\Lambda) \subseteq \Lambda$, we can ensure solvability of the doubly implicit MP-LPV system. Note that we can only ensure unique solvability, if we can confirm that the sets Λ_m are non-overlapping. We can however confirm that finding nonempty sets $\Lambda_{\text{inv}} \subseteq \text{Pre}(\Lambda) \subseteq \Lambda$ are a necessary and sufficient condition for existence and solvability of the doubly implicit MP-LPV system; the condition for the target set defined in Equation 4-47 can be considered as a necessary and sufficient condition for existence of solution $x(k)$. Let us present this in a formal proposition:

Proposition 4.10. *The existence of nonempty sets $\Lambda_{inv} \subseteq Pre(\Lambda) \subseteq \Lambda$ are a necessary and sufficient condition for existence and solvability of the doubly implicit MP-LPV system.*

Proof. The necessary proof is equivalent to the proof in Proposition 4.6. For the sufficient proof, we need to prove that if the doubly implicit MP-LPV system is solvable, we can find nonempty sets $\Lambda_{\text{inv}} \subseteq \text{Pre}(\Lambda) \subseteq \Lambda$. We can prove this using proof by contradiction. For this, let us assume that the sets $\Lambda_{\text{inv}} \subseteq \text{Pre}(\Lambda) \subseteq \Lambda$ are empty. We then cannot find a solution $x(k)$ such that Equation 4-47 holds. However, because of the definition of the state equation of the doubly implicit MP-LPV system, we then also cannot find a solution to Equation 4-45 and therefore cannot confirm existence of a solution as defined in Definition 4.2. If Definition 4.2 does not hold, we can also state that Definition 4.4 does not hold. Thus, by proof of contradiction, we can confirm the sufficient part. □

Note that the difference with single implicit MP-LPV systems is that the target set condition in Λ is necessary and sufficient for existence of doubly implicit MP-LPV systems, and the condition in $\mathcal{P}$ for single implicit MP-LPV systems is only necessary for existence. Finally, we can again make the following statements, similarly as for single implicit MP-LPV systems; if the target set Λ is empty, the precursor set $\text{Pre}(\Lambda)$ and invariant set Λ_{inv} are automatically empty as well, and we cannot guarantee (unique) solvability nor (unique) existence. If Λ and $\text{Pre}(\Lambda)$ are nonempty, but the invariant set Λ_{inv} is empty, only (unique) existence can be guaranteed.

4-7 Case Study IV: Solvability of the Urban Railway System

Let us investigate the solvability property of the URS described in Equation 3-30 in Case Study II in section 3-3. We have already seen that the concerning state-space description is in a single implicit format, given that the order of states is maintained. Let us recall the A_0

matrix of the system:

$$A_0\left(p^{(0)}(k)\right) = \begin{bmatrix} \varepsilon & \dots & \dots & \dots & \dots & \varepsilon \\ f_{\text{MMPS}}\left(p_1^{(0)}(k)\right) & \varepsilon & \dots & \dots & \dots & \varepsilon \\ \varepsilon & \tau_{r,2}(k) & \varepsilon & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \varepsilon & \varepsilon & \dots & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \dots & \varepsilon & f_{\text{MMPS}}\left(p_J^{(0)}(k)\right) & \varepsilon \end{bmatrix} \tag{4-60}$$

In which the MMPS functions are defined as:

$$f_{\text{MMPS}}\left(p_j^{(0)}(k)\right) = f_{\text{MMPS}}(a_j(k), d_j(k-1), \rho_{j-1}(k)) \tag{4-61}$$

Of which more details can be found in Equation 3-28. If the order of states indeed is retained, $a_j(k)$ in Equation 4-61 is a known value on the moment that $d_j(k)$ is being calculated, and this MMPS function therefore does not contain any implicit values. It can be observed that the A_0 matrix has a strictly lower triangular reducible structure. By Proposition 4.4, we can therefore conclude that the URS is uniquely solvable, given that this structure is maintained for all $k + j$ with $j \geq 0$. By construction of the expressions of states, we can indeed confirm that this structure will be maintained for all future states. We can therefore conclude that the URS presented in Case Study II in 3-3 is uniquely solvable as defined in Definition 4.5. Thus, due to the level of implicitness and the reducible structure of the A_0 matrix, there is no need to obtain the sets $\mathcal{P}_{\text{inv}} \subseteq \text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$, as the Kleene star of A_0 always exists.

4-8 Case Study V: Solvability of the Urban Railway System with Limited Capacity

In this section, we will analyze the solvability of the URS with limited capacity presented in Case Study III in section 3-4. Let us recall the A_0 matrix for the initial station $j = 1$:

$$A_0\left(p^{(0)}(k)\right) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ f_{\text{MMPS},1}\left(p_1^{(0)}(k)\right) & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & f_{\text{MMPS},2}\left(p_1^{(0)}(k)\right) & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & f_{\text{MMPS},3}\left(p_1^{(0)}(k)\right) & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & f_{\text{MMPS},4}\left(p_1^{(0)}(k)\right) & \varepsilon & \varepsilon \end{bmatrix} \tag{4-62}$$

In which the MMPS functions $f_{\text{MMPS},i}\left(p_1^{(0)}(k)\right)$ with $i = 1, ..., 4$ defined in Equation 3-48, 3-49, 3-50 and 3-51. We have seen in section 3-4 that similarly as for Case Study IV in the previous section, the URS is single implicit. Furthermore, as the A_0 matrix again is reducible with a strictly lower triangular structure, we can follow Proposition 4.4 and conclude that this URS is uniquely solvable, without obtaining the sets $\mathcal{P}_{\text{inv}} \subseteq \text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$.

Chapter 5

Model Predictive Control for Max-Plus Linear Parameter Varying Systems

This chapter is devoted to developing a Model Predictive Controller of the Urban Railway System (URS) system. Model Predictive Control (MPC) is introduced in 1978 by [35] and [36] and has gained popularity in the control community because of a wide set of useful properties; MPC is a model-based controller design method that can handle multi-input multi-output states. It can furthermore handle constraints and track reference signals and is an easy-to-tune method, making it easy to obtain the desired behaviour of the system. MPC has therefore proven itself to be useful for systems especially in the process industry. The conventional MPC approach was originally developed for discrete-time systems. However, for some classes of Discrete Event Systems (DES), MPC methods have also been constructed. The resulting methods still contain all the important building blocks of the conventional MPC framework, but these have been extended in such a manner that the resulting control approach is applicable to DES. Clearly, the difference between discrete-time systems and DES is system evolution over time versus the evolution over the occurrence of events. We will observe that this difference gives rise to different meanings of the MPC building blocks, such as the horizon and the objective. In this chapter, will investigate two developed MPC methods, namely for Max-Plus-Linear (MPL) systems and Mixed Logical Dynamical (MLD) systems. The MPC method for MPL systems is useful as MPL systems are DES systems. The MPC method for MLD systems is useful as MLD systems have the capability of handling real and integer variables, which will be useful for the URS. We will combine these methods and use the class equivalences with Max-Plus Linear Parameter Varying (MP-LPV) systems to construct an MPC approach applicable for the URS with limited capacity. The chapter is organized as follows. In section 5-1, we collect the important background information on the MPC approaches defined for MPL and MLD systems. We will introduce the format of Mixed Integer Linear Programming (MILP), an optimization problem formulation that will be of great importance when obtaining the optimization problem for the URS. In section 5-2, the URS with limited capacity will be rewritten into MLD format. Thereafter in section 5-3, we will define the optimization problem and section 5-4 introduces the modeling parameters that are considered. In section 5-5, we will present several simulations and include disturbances to visualize the performance of the MPC. Finally in 5-5, we will investigate whether we can conclude on closed-loop stability of the system.

5-1 Background Information

In this section, we discuss the MPC methods developed for MPL systems and MLD systems. The MPC method is an optimization method that calculates the optimal control input for each sample step k, while minimizing a certain objective function subject to constraints over a given *prediction horizon*. The prediction horizon indicates how far into the future the evolution of the system is estimated. The prediction horizon N_p is a design parameter must be long enough such that any unexpected future evolution of the system can still be covered by the input vector. The existing MPC approaches for MPL and MLD systems will be relevant for the control design for the URS system, as the latter will be based on those methods. In [37], there is also an MPC approach developed directly for Max-Min-Plus-Scaling (MMPS) systems. As the URS with limited capacity is initially derived as an MMPS system in section 4-7, the question arises on why we are not using this MPC method directly. However, these resulting MMPS-MPC optimization problems are nonlinear and nonconvex problems and therefore computationally hard to solve. The MPL- and MLD-MPC problems are linear and mixed-integer linear problems respectively, and therefore require less computational effort. The MLD format is furthermore useful as several states of the URS with limited capacity are already restricted to being integer variables (namely the states representing numbers of passengers: $q_j(k)$, $\rho_j(k)$ and $\sigma_j(k)$), and the MLD format is designed to handle such integer variables mixed with real variables. Finally, we have seen earlier in this thesis that MP-LPV systems are an extension of MPL systems. Furthermore, we have seen in section 3-2 that the classes of MLD systems, MMPS systems and MP-LPV systems are equivalent under mild assumptions. We will use these results in the following sections, to successfully obtain an MPC approach for the URS with limited capacity.

5-1-1 Model Predictive Control for Max-Plus Linear Systems

In this subsection, we introduce the MPC approach developed for MPL systems, based on the approach in [11]. For reading purposes, we repeat the state-space description of an MPL system as defined in Definition 2.8 in section 2-4, but now for the future evolution:

$$\begin{aligned} x(k+1) &= A \otimes x(k) \oplus B \otimes u(k) \\ y(k) &= C \otimes x(k) \end{aligned} \tag{5-1}$$

In which $A \in \mathbb{R}_\varepsilon^{n\times n}$, $B \in \mathbb{R}_\varepsilon^{n\times m}$ and $A \in \mathbb{R}_\varepsilon^{l\times n}$ the system matrices, and $x(k) \in \mathbb{R}_\varepsilon^{n}$, $u(k) \in \mathbb{R}_\varepsilon^{m}$ and $y(k) \in \mathbb{R}_\varepsilon^{l}$ the state, input and output vector respectively. Furthermore, as MPL systems describe DES, k is an event counter.

Let us first analyze the evolution of the system. For this, let us assume that the state vector $x(k)$ is measurable on event step k, or can be estimated based on previous measurements. Let us define the input sequence as the following:

$$\tilde{u}(k) = [u^T(k) \dots u^T(k+N_p-1)]^T \tag{5-2}$$

Using the state-space description described in Equation 5-1 as well as the input sequence, we can obtain an estimate for the evolution of the output $\tilde{y}(k)$. We have:

$$\tilde{y}(k) = [\hat{y}^T(k+1|k)\dots\hat{y}^T(k+N_p|k)]^T = H \otimes \tilde{u}(k) \oplus g(k) \tag{5-3}$$

In which:

$$H = \begin{bmatrix} C\otimes B & \varepsilon & \dots & \varepsilon \\ C\otimes A\otimes B & C\otimes B & \dots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ C\otimes A^{\otimes N_p-1}\otimes B & C\otimes A^{\otimes N_p-2}\otimes B & \dots & C\otimes B \end{bmatrix},$$
$$g(k) = \begin{bmatrix} C\otimes A \\ C\otimes A^{\otimes 2} \\ \vdots \\ C\otimes A^{\otimes N_p} \end{bmatrix} \otimes x(k) \tag{5-4}$$

Let us now investigate the possibilities for the cost criterion or objective function, which generally is defined as:

$$\begin{aligned} J(k) &= J_{\text{out}}(k) + \lambda J_{\text{in}}(k) \\ &= \underbrace{(\tilde{y}(k)-\tilde{r}(k))^T(\tilde{y}(k)-\tilde{r}(k))}_{J_{\text{out}}(k)} + \lambda \underbrace{\tilde{u}^T(k)\tilde{u}(k)}_{J_{\text{in}}(k)} \end{aligned} \tag{5-5}$$

In which $J_{\text{out}}(k)$ and $J_{\text{in}}(k)$ the reference tracking error and the control effort respectively. In MPC methods for conventional algebra, tracking the reference error is an often implemented optimization goal. Here, both the reference signal as the output then evolve over time. However, as we are dealing with DES, the output represents the occurrence of some event. Therefore, we must interpret a reference signal as due dates for the events. If we assume that these due dates are known, we can for example state that a penalty has to be paid for every delay. Then a well-suited choice for $J_{\text{out}(k)}$ is the tardiness, minimizing the penalty costs:

$$J_{\text{out},1} = \sum_{j=1}^{N_p}\sum_{i=1}^{l} \max(\hat{y}_i(k+1|k) - r_i(k+j), 0) \tag{5-6}$$

Another example of a situation is if we have perishable goods. To optimize the use of those goods, we would want to minimize the differences between the due dates and the actual output time instants:

$$J_{\text{out},2} = \sum_{j=1}^{N_p}\sum_{i=1}^{l} |\hat{y}_i(k+1|k) - r_i(k+j)| \tag{5-7}$$

A final example for $J_{\text{out}}(k)$ is the case of wanting to balance the output rates:

$$J_{\text{out},3} = \sum_{j=1}^{N_p}\sum_{i=1}^{l} |\Delta^2\hat{y}_i(k+1|k)| \tag{5-8}$$

In which the delta operator Δ is defined as:

$$\Delta^2 s(k) = \Delta s(k) - \Delta s(k-1) = s(k) - 2s(k-1) + s(k-2) \tag{5-9}$$

For the input cost criterion J_{in} for conventional MPC, it is very common to minimize the inputs. However, for DES this could result in input buffer overflows. It is therefore proposed that for DES, the goal should be to maximize the input time instants, keeping the buffer levels as low as possible. This can be achieved by using the following:

$$J_{\text{in},1} = -\sum_{j=1}^{N_p}\sum_{i=1}^{l} u_i(k+j-1) \tag{5-10}$$

Or:

$$J_{\text{in},2} = -\tilde{u}(k)^T\tilde{u}(k) \tag{5-11}$$

Another example for the input cost criterion is wanting to balance the input rates:

$$J_{\text{in},3} = \sum_{j=1}^{N_p}\sum_{i=1}^{l} |\Delta^2 u_i(ik+j)| \tag{5-12}$$

Again with Δ^2 defined as in Equation 5-9. We now have defined multiple options for the cost criterion with operators of conventional algebra. These operators can be replaced with their max-plus algebraic analogues, but that is not necessary per se. The optimization problem of MPC for MPL systems does not need to be in max-plus algebra completely; the goal of the MPC approach is to calculate the optimal input sequence while satisfying the constraints. This closed loop system is usually not an MPL system. This sequence can thereafter be fed into the MPL system, as this sequence will just contain event times.

We furthermore need to obtain a constraint equation for the MPC framework applied to MPL systems. Just as for conventional MPC, a typical constraint is to set upper and lower boundaries on the event times. This is usually represented in the following manner:

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) \tag{5-13}$$

In which $E(k) \in \mathbb{R}^{p\times mN_p}$, $F(k) \in \mathbb{R}^{p\times lN_p}$ and $h(k) \in \mathbb{R}^p$ with some integer p. It again does not matter that this equation is in conventional algebra, because of the same reasoning as for the cost criterion.

The MPL systems that we consider, correspond to DES systems with only synchronization and no choice. Due to the synchronization, the occurrence time of the events in such systems is consecutive; a new event can only occur after the previous event has finished. Therefore, as the input sequence represents the occurrence times of events, this sequence should always be non-decreasing and is furthermore often increasing. We therefore introduce the following constraint:

$$\Delta u(k+j) = u(k+j) - u(k+j-1) \geq 0 \qquad \text{for } j = 0,\ \ldots,\ N_p - 1 \tag{5-14}$$

We finally look at the max-plus algebraic interpretation of the receding control horizon principle as described in the beginning of this section. From this explanation, we would imply that the input should stay constant from event step $k + N_c$ on. However, as the input sequence normally is increasing with every next event step k, this principle is changed as follows; from event step $k + N_c$ on, the feeding rate should become constant:

$$\Delta u(k+j) = \Delta u(k+N_c-1), \qquad \text{for } j = N_c, N_c+1, ..., N_p - 1 \tag{5-15}$$

We can now finally state the MPC problem for MPL systems as:

Definition 5.1 (MPC Problem for MPL Systems)**.**

$$\min_{\{u(k),\ldots,u(k+N_p-1)\}} J(k) = \min_{\{u(k),\ldots,u(k+N_p-1)\}} J_{\text{out}}(k) + \lambda J_{\text{in}}(k) \tag{5-16}$$

Such that:

$$\begin{aligned}
&\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k) \\
&E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) \\
&x(k+1) = A \otimes x(k) \oplus B \otimes u(k) \\
&y(k) = C \otimes x(k) \\
&\Delta u(k+j) \geq 0, \quad j = N_c,\ N_c+1, \ldots,\ N_p - 1 \\
&\Delta u(k+j) = \Delta u(k+N_c-1), \quad j = N_c,\ N_c+1, \ldots,\ N_p - 1
\end{aligned} \tag{5-17}$$

With H and $g(k)$ defined as in Equation 5-4 and $\tilde{u}(k)$ defined as in Equation 5-2. Furthermore $E(k) \in \mathbb{R}^{p\times mN_p}$, $F(k) \in \mathbb{R}^{p\times lN_p}$, $h(k) \in \mathbb{R}^p$ for some integer p. Also $A \in \mathbb{R}_\varepsilon^{n\times n}$, $B \in \mathbb{R}_\varepsilon^{n\times m}$ and $A \in \mathbb{R}_\varepsilon^{l\times n}$ the system matrices, and $x(k) \in \mathbb{R}_\varepsilon^n$, $u(k) \in \mathbb{R}_\varepsilon^m$ and $y(k) \in \mathbb{R}_\varepsilon^l$ the state, input and output vector respectively. With finally N_p the prediction horizon and N_c the control horizon.

5-1-2 Model Predictive Control for Mixed Logical Dynamical Systems

In this subsection, we introduce the MPC approach developed for MLD systems, based on [38]. MLD systems have been introduced in subsection 2-5-3 and have shown to be useful as a modeling technique for systems with dynamics and constraints that are interdependent. We have seen the rules that are implemented to transform logical statements involving continuous variables into (mixed-integer) linear inequality constraints. Note that the MLD system considered in [38] is in discrete-time format. We can however use this format for discrete-event systems in a similar manner. For reading purposes, let us recall the MLD system description as defined in Definition 2.17:

$$\begin{aligned}
x(k+1) &= Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \\
y(k) &= Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k)
\end{aligned} \tag{5-18}$$

Subject to the constraint:

$$E_1x(k) + E_2u(k) + E_3\delta(k) + E_4z(k) \leq g \tag{5-19}$$

With state vector $x(k) \in \begin{bmatrix} x_r^\top(k) & x_b^\top(k) \end{bmatrix}^\top$ with $x_r(k) \in \mathbb{R}^{n_r}$ real states, $x_b(k) \in \{0,1\}^{n_b}$ binary states and $n = n_r + n_b$. Output $y(k) \in \mathbb{R}^{n_y}$ and input $u(k) \in \mathbb{R}^{n_u}$ are defined similarly, and $z(k) \in \mathbb{R}^{n_r}$ and $\delta(k) \in \{0,1\}^{n_b}$ auxiliary variables. The system matrices are defined as $A \in \mathbb{R}^{n\times n}$, $B_1 \in \mathbb{R}^{n\times n_u}$, $B_2 \in \mathbb{R}^{n\times n_b}$, $B_3 \in \mathbb{R}^{n\times n_r}$, $C \in \mathbb{R}^{n_y\times n}$, $D_1 \in \mathbb{R}^{n_y\times n_u}$, $D_2 \in \mathbb{R}^{n_y\times n_b}$ and $D_3 \in \mathbb{R}^{n_y\times n_r}$. Furthermore we have $E_1 \in \mathbb{R}^{n_g\times n}$, $E_2 \in \mathbb{R}^{n_g\times n_u}$, $E_3 \in \mathbb{R}^{n_g\times n_b}$, $E_4 \in \mathbb{R}^{n_g\times n_r}$ and vector $g \in \mathbb{R}^{n_g}$.

To describe the MPC approach for MLD systems, we can again define the input control sequence, similar to the input sequence defined for MPL systems in Equation 5-2:

$$u_0^{N_p-1} = [u^T(0)\ u^T(1) \ldots u^T(N_p-1)]^T \tag{5-20}$$

In which N_p is again the prediction horizon. [38] suggests a simple cost criterion, aiming to regulate an initial state x_0 to a target state, while minimizing the control effort. Such a cost criterion is equivalent to minimizing the reference tracking error as described for MPC for MPL systems. The value of the target state can vary over the receding horizon. Let us define this cost criterion in the following manner:

$$\begin{aligned} J\left(u_0^{N_p-1}, x_0\right) = \sum_{t=0}^{N_p-1} & ||u(t) - u_f||_{Q_1}^2 + ||\delta(t, x_0, u_0^t) - \delta_f||_{Q_2}^2 + ||z(t, x_0, u_0^t) - z_f||_{Q_3}^2 \\ & + ||x(t, x_0, u_0^{t-1}) - x_f||_{Q_4}^2 + ||y(t, x_0, u_0^{t-1}) - y_f||_{Q_4}^2 \end{aligned} \tag{5-21}$$

In which x_0 is the initial state, $\delta(t, x_0, u_0^t)$, $z(t, x_0, u_0^t)$, $x(t, x_0, u_0^{t-1})$ and $y(t, x_0, u_0^{t-1})$ following from the initial state and input vector and the system dynamics described as in Equation 5-18, and u_f, δ_f, z_f and y_f the final values for target state x_f. Furthermore we have Q_i with $i = 1, ..., 4$ as weight matrices. It remains to investigate which constraints can and need to be included. In the case for the above cost criterion, we have to add a constraint ensuring that the target state is reached, defined as:

$$x\left(N_p, x_0, u_0^{N_p-1}\right) = x_f \tag{5-22}$$

We can furthermore define upper and lower bounds on the both the state and the input in the following manner, based on [39]:

$$\begin{aligned} & u_{\min} \leq u(t+k) \leq u_{\max} && k = 0, \dots, N_p - 1 \\ & x_{\min} - \sigma \leq x(t+k|t) \leq x_{\min} - \sigma && k = 0, \dots, N_c \\ & 0 \leq \sigma \leq \sigma(t-1) \end{aligned} \tag{5-23}$$

In which N_c is the control horizon, $x(k|t)$ is the state predicted at time $(t+k)$ by applying the input $u(t+k)$, and σ is a slack variable introduced to make the constraint on $x(k+t|t)$ a soft constraint; the MPC approach in [38] distinguishes hard and soft constraints. Hard constraints can be considered as constraints that cannot be violated (such as voltage limits), while soft constraints are allowed to be violated (such as temperature bounds) to some extend. We are now ready to define the MPC problem for MLD systems:

Definition 5.2 (MPC Problem for MLD Systems)**.**

$$\begin{aligned} \min J\left(u_0^{N_p-1}, x_0\right) = \min \sum_{t=0}^{N_p-1} & ||u(t) - u_f||_{Q_1}^2 + ||\delta(t, x_0, u_0^t) - \delta_f||_{Q_2}^2 \\ & + ||z(t, x_0, u_0^t) - z_f||_{Q_3}^2 + ||x(t, x_0, u_0^{t-1}) - x_f||_{Q_4}^2 \\ & + ||y(t, x_0, u_0^{t-1}) - y_f||_{Q_4}^2 \end{aligned} \tag{5-24}$$

Such that:

$$\begin{aligned} x(k+1) &= Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \\ y(k) &= Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \\ & E_1x(k) + E_2u(k) + E_3\delta(k) + E_4z(k) \leq g \\ & u_{\min} \leq u(t+k) \leq u_{\max} && k = 0, \dots, N_p - 1 \\ & x_{\min} - \sigma \leq x(t+k|t) \leq x_{\min} - \sigma && k = 0, \dots, N_c \\ & 0 \leq \sigma \leq \sigma(t-1) \\ & x\left(N_p, x_0, u_0^{N_p-1}\right) = x_f \end{aligned} \tag{5-25}$$

With the variables all defined in previously in Equation 5-18, 5-19, 5-20, 5-21, 5-22 and 5-23.

5-1-3 Mixed Integer Linear Programming

Optimization problems for MLD systems can often be reformulated into MILP problems; such problems are described by a linear objective function and linear constraints, and a state vector divided in states that can take on real values and states that are restricted to be integer. Similarly as for MLD systems, nonlinear optimization problems can be recasted into MILP problems, and can then be solved by MILP solvers. MILP solving algorithms have shown to be more efficient compared to nonlinear optimization algorithms, and can therefore considered to be very useful for online MPC. A standard format for an MILP optimization problem can be defined in the following manner:

$$\begin{aligned} \min\ & c_I^T x_I + c_R^T x_R \\ \text{such that}\ & A_I x_I + A_R x_R \leq b \\ & A_{I,\text{eq}} x_I + A_{R,\text{eq}} x_R = b_{\text{eq}} \\ & x_{\min} \leq x \leq x_{\max} \\ & x_I \in \mathbb{Z}^{n_I},\ \ x_R \in \mathbb{R}^{n_R} \end{aligned} \tag{5-26}$$

Examples of common MILP solvers are the standard function `intlinprog` in MATLAB or the more powerful mathematical optimization solver GUROBI.

5-1-4 Mixed Integer Quadratic Programming

The optimization problem defined in Definition 5.2 can be solved as a Mixed Integer Quadratic Programming (MIQP), due to the presence of the 2-norm $||\cdot||^2$ in the objective function. Such a 2-norm is considered to be a quadratic term. We have seen that MILP problems are constructed by a linear objective and linear constraints. Its extension is therefore MIQP which allows quadratic terms in only the objective function. We can define a standard format for such MIQP optimization problems in the following manner:

$$\begin{aligned} \min\ & x_I^T C_I x_I + x_R^T C_R x_R + c_I^T x_I + c_R^T x_R \\ \text{such that}\ & A_I x_I + A_R x_R \leq b \\ & A_{I,\text{eq}} x_I + A_{R,\text{eq}} x_R = b_{\text{eq}} \\ & x_{\min} \leq x \leq x_{\max} \\ & x_I \in \mathbb{Z}^{n_I},\ \ x_R \in \mathbb{R}^{n_R} \end{aligned} \tag{5-27}$$

To solve this MIQP problem, one can again use GUROBI.

5-2 Urban Railway System as MLD System

Now that we have seen the MPC strategies developed for MPL and MLD systems, and furthermore discussed the efficient solving strategies MILP and MIQP suitable for MLD-MPC problems, we will write the URS with maximum capacity defined in section 3-4 into

MLD format. Let us recall the URS description defined in section 3-4. This expression is in MMPS format, but we have seen in subsection 2-5-4 that MMPS and MLD systems coincide. We can therefore rewrite the system description without any problems. In MMPS notation, recall that we have:

$$\begin{aligned} a_j(k) &= \max\Big(d_{j-1}(k) + \tau_{r,j}, d_j(k-1) + \tau_h\Big) \\ d_j(k) &= \max\Bigg(\tau_{\text{dwell},j,\min} + a_j(k), \min\Big(\alpha_1 a_j(k) + \alpha_2 \rho_{j-1}(k) + \alpha_3 \sigma_j(k-1) \\ &\qquad + \alpha_4 d_j(k-1), \gamma_0 + \gamma_1 a_j(k) + \gamma_2 \rho_{j-1}(k)\Big)\Bigg) \end{aligned} \tag{5-28}$$

In which $a_j(k) \in \mathbb{R}$ and $d_j(k) \in \mathbb{R}$ the arrival time and departure time of train k from and at station j. The variables $\alpha_{(\cdot)} \in \mathbb{R}$ and $\gamma_{(\cdot)} \in \mathbb{R}$ are scalar entries defined as in Equation 3-40 and 3-42. Let us furthermore define $q_j(k) \in \mathbb{Z}$ the number of passengers that board train k at station j, as:

$$q_j(k) = \min\left(e_j\left(d_j(k) - d_j(k-1)\right) + \sigma_j(k-1), \rho_{\max} - (1-\beta_j)\rho_{j-1}(k)\right) \tag{5-29}$$

With $e_j \in \mathbb{Z}$ the number of passengers arriving between departures of train $k-1$ and k at station j per time index, $\rho_{\max} \in \mathbb{Z}$ the maximum capacity of the trains (fixed), $\sigma_j(k) \in \mathbb{Z}$ the number of passengers left behind at station j by train k and $\rho_j(k) \in \mathbb{Z}$ is the number of passengers present in train k at departure from station j, defined as:

$$\rho_j(k) = (1-\beta_j)\rho_{j-1}(k) + q_j(k) \tag{5-30}$$

In which $\beta_j \in [0,1]$ is the fraction of passengers that disembarks at station j. Finally, $\sigma_j(k)$ is defined as:

$$\sigma_j(k) = e_j(d_j(k) - d_j(k-1)) + \sigma_j(k-1) - q_j(k) \tag{5-31}$$

Goal is to rewrite the above expressions such that the system fits in the MLD framework defined as in Definition 2.17. For this, we will use the auxiliary variables $\delta(k) \in \{0,1\}$ and $z(k) \in \mathbb{R}$ defined in subsection 2-5-3. Let us begin with the expression of the arrival time $a_j(k)$:

$$\begin{aligned} a_j(k) &= \max(\underbrace{d_{j-1}(k) + \tau_{r,j}}_{h_{1,j}(k)}, \underbrace{d_j(k-1) + \tau_h}_{h_{2,j}(k)}) \\ &= \max(h_{1,j}(k), h_{2,j}(k)) = -\min(-h_{1,j}(k), -h_{2,j}(k)) \end{aligned} \tag{5-32}$$

We introduce:

$$f_{1,j}(k) = -h_{2,j}(k) - (-h_{1,j}(k)) = -h_{2,j}(k) + h_{1,j}(k) \qquad \delta_{1,j}(k) = \begin{cases} 1 \text{ if } f_{1,j}(k) \leq 0 \\ 0 \text{ if } f_{1,j}(k) > 0 \end{cases} \tag{5-33}$$

We define the upper and lower bound of $f_{1,j}(k)$ as follows:

$$m_1 \leq f_{1,j}(k) \leq M_1 \tag{5-34}$$

We can then write:

$$\begin{aligned} a_j(k) &= -\Big(-h_{1,j}(k) + \Big(-h_{2,j}(k) + h_{1,j}(k)\Big)\delta_{1,j}(k)\Big) = h_{1,j}(k) - \underbrace{f_{1,j}(k)\delta_{1,j}(k)}_{z_{1,j}(k)} \\ &= h_{1,j}(k) - z_{1,j}(k) \end{aligned} \tag{5-35}$$

The upper and lower bound of $f_{1,j}(k)$ in Equation 5-34 and the expression $z_{1,j}(k) = f_{1,j}(k)\delta_{1,j}(k)$ can be expressed equivalently as the following constraints:

$$\begin{aligned}
&f_{1,j}(k) \leq M_1\Big(1-\delta_{1,j}(k)\Big)\\
&f_{1,j}(k) \geq \epsilon + \Big(m_1-\epsilon\Big)\delta_{1,j}(k)\\
&z_{1,j}(k) \leq M_1\delta_{1,j}(k)\\
&z_{1,j}(k) \geq m_1\delta_{1,j}(k)\\
&z_{1,j}(k) \leq f_{1,j}(k) - m_1\Big(1-\delta_{1,j}(k)\Big)\\
&z_{1,j}(k) \geq f_{1,j}(k) - M_1\Big(1-\delta_{1,j}(k)\Big)
\end{aligned} \tag{5-36}$$

In which ϵ represents the machine precision error. We can follow the same procedure for the remaining expressions that contain a maximization and/or minimization. For the departure time $d_j(k)$, we can write:

$$\begin{aligned}
d_j(k) =&\max(\underbrace{\tau_{\text{dwell},j,\min} + a_j(k)}_{h_{3,j}(k)}, \min(\underbrace{\alpha_1 a_j(k) + \alpha_2\rho_{j-1}(k) + \alpha_3\sigma_j(k-1) + \alpha_4 d_j(k-1)}_{g_{1,j}(k)},\\
&\underbrace{\gamma_0 + \gamma_1 a_j(k) + \gamma_2\rho_{j-1}(k)}_{g_{2,j}(k)}))\\
=&\max(h_{3,j}(k), \underbrace{\min(g_{1,j}(k), g_{2,j}(k))}_{h_{4,j}(k)})\\
=&\max(h_{3,j}(k), h_{4,j}(k)) = -\min(-h_{3,j}(k), -h_{4,j}(k))
\end{aligned} \tag{5-37}$$

Let us start with rewriting $h_{4,j}(k) = \min(g_{1,j}(k), g_{2,j}(k))$. For this, we introduce:

$$f_{2,j}(k) = g_{2,j}(k) - g_{1,j}(k) \tag{5-38}$$

We define the upper and lower bound of $f_{2,j}(k)$ as follows:

$$m_2 \leq f_{2,j}(k) \leq M_2 \tag{5-39}$$

Let us furthermore define the auxiliary variable $\delta_{2,j}(k)$:

$$\delta_{2,j}(k) = \begin{cases} 1 \text{ if } f_{2,j}(k) \leq 0 \\ 0 \text{ if } f_{2,j}(k) > 0 \end{cases} \tag{5-40}$$

We can then rewrite $h_{4,j}(k)$ in Equation 5-37 in the following manner:

$$h_{4,j}(k) = g_{1,j}(k) + \Big(g_{2,j}(k) - g_{1,j}(k)\Big)\delta_{2,j}(k) = g_{1,j}(k) + \underbrace{f_{2,j}(k)\delta_{2,j}(k)}_{z_{2,j}(k)} = g_{1,j}(k) + z_{2,j}(k) \tag{5-41}$$

Equivalently, we can rewrite $d_j(k)$. Let us introduce again the following:

$$f_{3,j}(k) = -h_{3,j}(k) + h_{4,j}(k) \qquad\qquad \delta_{3,j}(k) = \begin{cases} 1 \text{ if } f_{3,j}(k) \leq 0 \\ 0 \text{ if } f_{3,j}(k) > 0 \end{cases} \tag{5-42}$$

Again with $f_{3,j}(k)$ bounded as:

$$m_3 \leq f_{3,j}(k) \leq M_3 \tag{5-43}$$

We can then rewrite the equation for the departure time $d_j(k)$ as the following:

$$\begin{aligned} d_j(k) &= -\Big(- h_{4,j}(k) + \Big(- h_{3,j}(k) + h_{4,j}(k)\Big)\delta_{3,j}(k)\Big) = h_{4,j}(k) - \underbrace{f_{3,j}(k)\delta_{3,j}(k)}_{z_{3,j}(k)} \\ &= g_{1,j}(k) + z_{2,j}(k) - z_{3,j}(k) \end{aligned} \tag{5-44}$$

Finally, we can rewrite Equation 5-39 and 5-43 and $z_{2,j}(k) = f_{2,j}(k)\delta_{2,j}(k)$ and $z_{3,j}(k) = f_{3,j}(k)\delta_{3,j}(k)$ as the following constraints:

$$\begin{aligned} &f_{2,j}(k) \leq M_2\Big(1 - \delta_{2,j}(k)\Big) && f_{3,j}(k) \leq M_3\Big(1 - \delta_{3,j}(k)\Big) \\ &f_{2,j}(k) \geq \epsilon + \Big(m_2 - \epsilon\Big)\delta_{2,j}(k) && f_{3,j}(k) \geq \epsilon + \Big(m_3 - \epsilon\Big)\delta_{3,j}(k) \\ &z_{2,j}(k) \leq M_2\delta_{2,j}(k) && z_{3,j}(k) \leq M_3\delta_{3,j}(k) \\ &z_{2,j}(k) \geq m_2\delta_{2,j}(k) && z_{3,j}(k) \geq m_3\delta_{3,j}(k) \\ &z_{2,j}(k) \leq f_{2,j}(k) - m_2\Big(1 - \delta_{2,j}(k)\Big) && z_{3,j}(k) \leq f_{3,j}(k) - m_3\Big(1 - \delta_{3,j}(k)\Big) \\ &z_{2,j}(k) \geq f_{2,j}(k) - M_2\Big(1 - \delta_{2,j}(k)\Big) && z_{3,j}(k) \geq f_{3,j}(k) - M_3(1 - \delta_{3,j}(k)) \end{aligned} \tag{5-45}$$

Finally for the number of passengers boarding the train $q_j(k)$, we can write:

$$\begin{aligned} q_j(k) &= \min(\underbrace{e_j(d_j(k) - d_j(k-1)) + \sigma_j(k-1)}_{h_{5,j}(k)}, \underbrace{\rho_{\max} - (1 - \beta_j)\rho_{j-1}(k)}_{h_{6,j}(k)}) \\ &= \min(h_{5,j}(k), h_{6,j}(k)) \end{aligned} \tag{5-46}$$

We introduce:

$$f_{4,j}(k) = h_{6,j}(k) - h_{5,j}(k) \qquad\qquad \delta_{4,j}(k)(k) = \begin{cases} 1 \text{ if } f_{4,j}(k) \leq 0 \\ 0 \text{ if } f_{4,j}(k) > 0 \end{cases} \tag{5-47}$$

With:

$$m_4 \leq f_{4,j}(k) \leq M_4 \tag{5-48}$$

We then have:

$$q_j(k) = h_{5,j}(k) + (h_{6,j}(k) - h_{5,j}(k))\delta_{4,j}(k) = h_{5,j}(k) + \underbrace{f_{4,j}(k)\delta_{4,j}(k)}_{z_{4,j}(k)} = h_{5,j}(k) + z_{4,j}(k) \tag{5-49}$$

Subject to:

$$\begin{aligned} &f_{4,j}(k) \leq M_4\Big(1 - \delta_{4,j}(k)\Big) \\ &f_{4,j}(k) \geq \epsilon + \Big(m_4 - \epsilon\Big)\delta_{4,j}(k) \\ &z_{4,j}(k) \leq M_4\delta_{4,j}(k) \\ &z_{4,j}(k) \geq m_4\delta_{4,j}(k) \\ &z_{4,j}(k) \leq f_{4,j}(k) - m_4\Big(1 - \delta_{4,j}(k)\Big) \\ &z_{4,j}(k) \geq f_{4,j}(k) - M_4\Big(1 - \delta_{4,j}(k)\Big) \end{aligned} \tag{5-50}$$

So we now have rephrased the expressions that include maximization and minimization into MLD format, such that we can obtain an optimization problem that includes only linear (in)equality constraints. Combining the necessary expressions results in the following linear expression of the URS:

$$\text{URS: } \begin{cases} a_j(k) & = h_{1,j}(k) - z_{1,j}(k) \\ d_j(k) & = g_{1,j}(k) + z_{2,j}(k) - z_{3,j}(k) \\ q_j(k) & = h_{5,j}(k) + z_{4,j}(k) \\ \rho_j(k) & = (1-\beta_j)\rho_{j-1}(k) + q_j(k) \\ \sigma_j(k) & = e_j(d_j(k) - d_j(k-1)) + \sigma_j(k-1) - q_j(k) \end{cases} \tag{5-51}$$

If we take the same state vector as defined in Equation 3-45 in Case Study III, we can write the following for train k at station j:

$$\left[\begin{array}{c} x_j(k) \\ \hline x_{j+1}(k) \\ \\ \hline x_{j+2}(k) \\ \hline x_{j+3}(k) \\ \hline x_{j+4}(k) \end{array}\right] = \left[\begin{array}{c} x_{j-4}(k) + \tau_{r,j} - z_{1,j}(k) \\ \hline \alpha_1 x_j(k) + \alpha_2 x_{j-2}(k) + \alpha_3 x_{j+4}(k-1) + \\ \alpha_4 x_{j+1}(k-1) + z_{2,j}(k) - z_{3,j}(k) \\ \hline e_j(x_{j+1}(k) - x_{j+1}(k-1)) + x_{j+4}(k-1) + z_{4,j}(k) \\ \hline (1-\beta_j)x_{j-2}(k) + x_{j+2}(k) \\ \hline e_j(x_{j+1}(k) - x_{j+1}(k-1)) + x_{j+4}(k-1) - x_{j+2}(k) \end{array}\right] \tag{5-52}$$

Subject to:

$$\begin{aligned} & f_{i,j}(k) \leq M_i\Big(1 - \delta_{i,j}(k)\Big) \\ & f_{i,j}(k) \geq \epsilon + \Big(m_i - \epsilon\Big)\delta_{i,j}(k) \\ & z_{i,j}(k) \leq M_i \delta_{i,j}(k) \\ & z_{i,j}(k) \geq m_i \delta_{i,j}(k) \qquad\qquad \text{for } i = 1, ..., 4 \\ & z_{i,j}(k) \leq f_{i,j}(k) - m_i\Big(1 - \delta_{i,j}(k)\Big) \\ & z_{i,j}(k) \geq f_{i,j}(k) - M_i\Big(1 - \delta_{i,j}(k)\Big) \end{aligned} \tag{5-53}$$

Note that the above MLD system now describes a DES, as the URS is a DES. This is different from the MLD system used to describe the MPC approach in subsection 5-1-2, which describes a discrete-time system.

5-3 Optimization Problem

We have seen that MPC methods provide us with an efficient approach to solve optimization problems, by means of obtaining an optimal input sequence for the prediction horizon. In this section, we will define the optimization problem for the URS with limited capacity. First in subsection 5-3-1, we will define the control input that will be used for optimization of the URS with limited capacity. Thereafter in subsection 5-3-2, we will formulate the objective function that will be minimized in this optimization problem. We have seen in the earlier defined MPC methods for MPL and MLD systems that there are multiple possibilities for the objective functions as we can focus on different types of criteria. We will give a short introduction of few of the possibilities for this cost criterion, but will thereafter focus on one objective function. Finally in subsection 5-3-3, we will collect all the necessary constraints for the optimization problem of the URS.

5-3-1 Control Input

Before we define an objective to minimize, let us define the control effort that can be used to steer the system into its optimal state. As we are considering a scheduling problem of a railway system, it makes sense to implement a control effort on the traveling speed of the trains. Realistically, this makes sense; within boundaries, a train can increase or decrease its speed. Up until now, we have assumed a fixed running time $\tau_{r,j}$ for trains between consecutive stations. As a control effort, we will therefore introduce $u_{1,j}(k)$ as running time input. This running time input can increase or decrease the running time of the train. This input is therefore analogue to an increase or decrease of running speed. We can include this running time input in the following manner:

$$a_j(k) = \max\Big(d_{j-1}(k) + \tau_{r,j} + u_{1,j}(k), d_j(k-1) + \tau_h\Big) \tag{5-54}$$

As there realistically is a limit on the running speed, we will need to include constraints on $u_{1,j}(k)$, that represent the boundaries on the running time:

$$\tau_{r,j}^{lb} \leq \tau_{r,j} + u_{1,j}(k) \leq \tau_{r,j}^{ub} \tag{5-55}$$

The values of these boundaries will be assigned later on. Let us furthermore include introduce another input $u_{2,j}(k)$; this input can decrease or increase the headway time τ_h between consecutive trains. For this, we assume that there is some slack in this safety distance. We can include this second input in the following manner:

$$a_j(k) = \max\Big(d_{j-1}(k) + \tau_{r,j} + u_{1,j}(k), d_j(k-1) + \tau_h + u_{2,j}(k)\Big) \tag{5-56}$$

Clearly we must assume a lower bound on this input $u_{2,j}(k)$, because we would otherwise tolerate collision between consecutive trains. Let us include therefore the following boundaries:

$$\tau_h^{lb} \leq \tau_h + u_{2,j}(k) \leq \tau_h^{ub} \tag{5-57}$$

Again, we will assign the values of these boundaries later on. As we have adjusted the arrival time expression, let us rewrite the arrival time expression in MLD format. For this, let us re-define the expression for $h_{1,j}(k)$ and $h_{2,j}(k)$ in Equation 5-32:

$$\begin{aligned} a_j(k) &= \max(\underbrace{d_{j-1}(k) + \tau_{r,j} + u_{1,j}(k)}_{h_{1,j}(k)}, \underbrace{d_j(k-1) + \tau_h + u_{2,j}(k)}_{h_{2,j}(k)}) \\ &= \max(h_{1,j}(k), h_{2,j}(k)) = -\min(-h_{1,j}(k), -h_{2,j}(k)) \end{aligned} \tag{5-58}$$

The expressions for $f_{1,j}(k)$ and $z_{1,j}(k)$ can therefore be kept equal. Note that another option for the control effort could be to control the departure time $d_j(k)$. This could be done by for example with a delay time, a 'force to leave early' time, a variable minimum dwell time or a combination of these input times. However, as the departure time is contained in the expression of the number of passengers boarding $q_j(k)$, these input types would also effect $q_j(k)$. And as the departure time again depends on the number of passengers boarding (and thus boarding speed), the resulting URS will then be more complex to stabilize. If we compare this to the running time input $u_{1,j}(k)$ and headway time input $u_{2,j}(k)$, we can observe that these values do not directly influence the number of passengers boarding. Furthermore, to prevent the trains from skipping stops, an input including a variable minimum dwell time should always be nonzero. As we will observe in the following section that the input effort is included in the objective function, this would be far from ideal.

5-3-2 Objective Function

In optimizing the scheduling problem for the URS, there are many possibilities for the objective function. The optimization problem can be train based or passenger based. Criteria for a train-based problem are for example energy consumption, train-delay, frequency optimization and many more. A few examples for passenger based criteria are passenger-delay, in-vehicle time and use of train capacity. The objective functions resulting from these examples will often result in a nonlinear function. For simplicity, we will therefore consider only one objective; we aim to minimize the number of passengers waiting on the platform. At each station there is a constant flow of passengers arriving per time index defined as $e_j \in \mathbb{Z}$. Passengers will therefore arrive in between departures of consecutive trains and have to wait some time before the next train departs. Furthermore, as the trains have a limited capacity, not all passengers will always fit and therefore will be left behind on the platform, which we have defined as $\sigma_j(k)$. We can define the number of passengers that are waiting on train k at station j at moment of arrival of train k therefore in the following manner:

$$p_j^{\text{wait}}(k) = e_j\Big(a_j(k) - d_j(k-1)\Big) + \sigma_j(k-1) \tag{5-59}$$

Minimizing the number of passengers waiting improves the overall travelling experience, as less people have to wait on a train. Another option would be to minimize the total waiting time of all the passengers; minimizing $p^{\text{wait}_j}(k)$ clearly has an influence on the total waiting time. From [40], we can obtain an approximation of the passenger waiting time at station j on train k in the following manner:

$$\tau_j^{\text{wait}}(k) = \sigma_j(k-1)(d_j(k) - d_j(k-1)) + \frac{1}{2}e_j(d_j(k) - d_j(k-1))^2 \tag{5-60}$$

In which the first term represents the waiting time of passengers that where left behind by the previous train, and the second term is an approximation of the waiting time of passengers arriving at j in between departures of the consecutive trains. It can however be observed that this value is in quadratic form. To minimize the optimization effort it is therefore more convenient to consider the number of passengers waiting (as Equation 5-59 is a linear expression). In conventional MPC, the objective function consists out of a reference tracking error and the control effort. This will be extended to the DES environment for the URS in the following manner:

$$J(k) = ||p_j^{\text{wait}}(k) - p_{\text{ref}}^{\text{wait}}||_1 + \lambda_1||u_{1,j}(k)||_1 + \lambda_2||u_{2,j}(k)||_1 \tag{5-61}$$

The value $p_{\text{ref}}^{\text{wait}}$ represents a reference value that we will try to follow during optimization. This value will be based on a steady-state value of $p_j^{\text{wait}}(k)$, which will be defined in section 5-5. We will see that the URS runs optimally when stabilized at a steady-state point. The objective is limited to only linear terms. The problem can thus be solved as an MILP problem. If the objective function would contain quadratic terms, the problem would be solved as an MIQP problem. This will however increase the computational effort, and therefore increase the run time of the script. The 1-norm is furthermore useful as we would like to minimize the total number of passengers waiting, and the total implemented running time input and headway time input, rather than for example using the ∞-norm which minimizes the maximum over all trains and stations. With the latter norm, the total number of passengers can result in a higher value than using the 1-norm. The values λ_1 and λ_2 in the objective function are weight values which are used to assign the importance of minimizing the total number of passengers waiting compared to minimizing the control effort, and will be assigned later on.

5-3-3 Constraints

Finally, let us formulate the constraints for the optimization problem. The first set of linear inequality constraints is already constructed in Equation 5-53 for $i = 1, ..., 4$. Furthermore, we must include the state expressions defined in Equation 5-52. There are still a few extra constraints necessary for the correct definition of the states of the URS, based on the assumptions taken for the system. As we have considered a maximum capacity ($\rho_{\max}$) of the trains, we need to define an upper bound on the number of passengers $\rho_j(k)$ in the train:

$$\rho_j(k) \leq \rho_{\max} \tag{5-62}$$

Furthermore, we clearly must have the following:

$$d_j(k), a_j(k), q_j(k), \rho_j(k), \sigma_j(k) \geq 0 \tag{5-63}$$

As the resulting problem will be solved as an MILP problem, we can assign which state values must be integer. Clearly, all state values representing numbers of passengers, must be integer. We therefore include the following constraint:

$$q_j(k), \rho_j(k), \sigma_j(k) \in \mathbb{Z}_+ \tag{5-64}$$

We furthermore have:

$$d_j(k), a_j(k) \in \mathbb{R} \tag{5-65}$$

And:

$$\delta_{i,j} \in \{0, 1\} \qquad \text{with } i = 1, ..., 4, \tag{5-66}$$

We furthermore need to set constraints on the optimization value $p_j^{\text{wait}}(k)$. Originally this value was not considered as a state, but for programming simplicity, we do consider it as a state value. We can then assign the constraints as:

$$p_j^{\text{wait}}(k) \geq 0 \tag{5-67}$$

And:

$$p_j^{\text{wait}}(k) \in \mathbb{Z}_+ \tag{5-68}$$

Finally, we have seen in Equation 5-55 and 5-57 that we need to include lower and upper bounds on the inputs $u_{j,1}(k)$ and $u_{j,2}(k)$:

$$\begin{aligned} \tau_{r,j}^{lb} \leq \tau_{r,j} + u_{1,j}(k) \leq \tau_{r,j}^{ub} \\ \tau_h^{lb} \leq \tau_h + u_{2,j}(k) \leq \tau_h^{ub} \end{aligned} \tag{5-69}$$

And as both $u_{j,1}(k)$ and $u_{j,2}(k)$ are time values, we have:

$$u_{1,j}(k), u_{2,j}(k) \in \mathbb{R} \tag{5-70}$$

5-4 Modeling Parameters

Now that we have defined the optimization problem that needs to be solved by the MPC, it remains to define the modeling parameters, after which we can actually design the MPC. We separate the modeling parameters into MPC parameters, model parameters and initial conditions.

5-4-1 MPC Parameters

As MPC makes use of the receding horizon principle, we need to assign values for the control horizon and the prediction horizon. We furthermore need to assign the tuning weights λ_1 and λ_2 defined in the objective function in Equation 5-61. We have defined these MPC parameters in the following manner:

Parameter	Definition	Value	Unit
K_c	Train Horizon	4	$[-]$
J_c	Station Horizon	4	$[-]$
K	Train Simulation Horizon	$3 \cdot K_c$	$[-]$
J	Station Simulation Horizon	$3 \cdot J_c$	$[-]$
λ_1	Tuning Weight for $u_{1,j}(k)$	0.1	$[-]$
λ_2	Tuning Weight for $u_{2,j}(k)$	0.1	$[-]$

Table 5-1: MPC control parameters.

As we want to optimize over a number of stations J ánd a number of trains K, we consider the horizon for both j and k. Therefore, the controller is designed over a horizon $K_c = 4$ trains and $J_c = 4$ stations. The same holds for the simulation time; we will simulate over $K_p = 12$ trains and $J_p = 12$ stations. The value $K_c = J_c = 4$ is chosen because it is assumed that in a real life situation, one can to some extend estimate the arrival and departure time of a train up until four trains and stations ahead.

Let us explain furthermore the choice of the tuning weight $\lambda_{1,2} = 0.1$. For DES, it is in general more logical to have a negative $\lambda_{1,2}$; a positive $\lambda_{1,2}$ results in a minimization of the input time instants while optimizing, and this could result in input buffer overflows in for example manufacturing systems. However, since the input considered for the URS represents an influence on the running and headway time, we actually wánt to minimize this value. The bigger the delay time, the less trains will cross the stations resulting in a lower frequency. For urban railways, a high frequency is desired. We will therefore have $\lambda_{1,2} > 0$. Furthermore, as it is more important to minimize the number of passengers waiting than to minimize the input value, we have chosen $0 < \lambda_{1,2} < 1$.

5-4-2 URS Model Parameters

We have seen in the expressions in Equation 5-52 and all the assumptions defined in Chapter 2 and 3 that there are several model parameters that are assumed to be known and therefore fixed. In the following table, we have collected these parameters:

Parameter	Definition	Value	Unit
τ_r	Running time between consecutive stations	180	[sec]
τ_h	Headway (safety) time between consecutive trains	30	[sec]
$\tau_{\text{dwell,min}}$	Minimum dwell time at each station	30	[sec]
$\rho_{\max}$	Maximum capacity of the trains	150	[pas.]
b	Passenger boarding speed	2	[pas./sec]
e	Speed of passengers arriving at the station	0.5	[pas./sec]
f	Passenger disembarking speed	2	[pas./sec]
β	Fraction of passengers disembarking at each station	0.5	[−]

Table 5-2: URS model parameters resulting from the assumptions in Chapter 2 and 3.

We have included the minimum dwell time $\tau_{\text{dwell,min}}$ to prevent the trains from skipping stops. Generally this is preferred to always provide passengers with the possibility to disembark (or board). However, there are scenarios where skipping stops is more useful, for example when a train is full and no one plans to disembark. In reality, it is indeed possible in some urban railway systems for passengers in the vehicle to notify the driver that they want to disembark (for example, by pushing a button in Dutch busses). We will mimic such a scenario in the following section, and analyze the corresponding simulation. For now however, we include the minimum dwell time.

We have seen that rewriting the expression for the departure time into an single implicit format as in Equation 5-28 resulted in another set of parameters, arising from the parameters defined in Table 5-2:

Parameter	Definition	Value	Unit
α_1	$\frac{b}{b-e}$	$\frac{4}{3}$	[−]
α_2	$\frac{b}{b-e}\frac{\beta}{f}$	$\frac{1}{3}$	[sec/pas.]
α_3	$\frac{1}{b-e}$	$\frac{2}{3}$	[sec/pas.]
α_4	$-\frac{e}{b-e}$	$-\frac{1}{3}$	[−]
γ_0	$\frac{1}{b}\rho_{\max}$	75	[sec]
γ_1	[−]	1	[−]
γ_2	$\frac{\beta}{f} - \frac{1-\beta}{b}$	0	[sec/pas.]

Table 5-3: Parameters in the expression for the departure time as defined in Equation 5-28.

Let us furthermore assign values to the lower and upper bounds in the constraints in Equation 5-53 resulting in the formulation of the MLD expression. Let us define these bounds as:

Parameter	Definition	Value
m_1	Lower bound of $f_{1,j}(k)$	$-3 \cdot \tau_r$
M_1	Upper bound of $f_{1,j}(k)$	$3 \cdot \tau_r$
m_2	Lower bound of $f_{2,j}(k)$	$-3 \cdot \rho_{\max}(b+f)$
M_2	Upper bound of $f_{2,j}(k)$	$3 \cdot \rho_{\max}(b+f)$
m_3	Lower bound of $f_{3,j}(k)$	$-3 \cdot \rho_{\max}(b+f)$
M_3	Upper bound of $f_{3,j}(k)$	$3 \cdot \rho_{\max}(b+f)$
m_4	Lower bound of $f_{4,j}(k)$	$-3 \cdot \rho_{\max}$
M_4	Upper bound of $f_{4,j}(k)$	$3 \cdot \rho_{\max}$

Table 5-4: Lower and upper bounds m_i and M_i with $i = 1, ..., 4$ for the constraints defined in Equation 5-53.

The values of these bounds are obtained by estimating the worst case (lower bound) and best case (upper bound) scenarios for the values $f_{i,j}(k)$ with $i = 1, ..., 4$, and then tripling these estimations to ensure that all the possible values are ensured to be within the bounds. It remains to assign the boundary values of the running time and headway time input $u_{1,j}(k)$ and $u_{2,j}(k)$. Let us define these bounds as:

Parameter	Definition	Value
$\tau_{r,j}^{lb}$	Lower bound of $\tau_{r,j} + u_{1,j}$	160
$\tau_{r,j}^{ub}$	Upper bound of $\tau_{r,j} + u_{1,j}$	250
τ_h^{lb}	Lower bound of $\tau_h + u_{2,j}$	20
τ_h^{ub}	Upper bound of $\tau_h + u_{2,j}$	80

Table 5-5: Lower and upper bounds of the input constraints as defined in Equation 5-69.

Note that these bounds are defined in such a manner that the running time input boundaries are:

$$-20 \leq u_{1,j} \leq 70 \tag{5-71}$$

And the headway time input boundaries are:

$$-10 \leq u_{1,j} \leq 50 \tag{5-72}$$

5-4-3 Initial Conditions

Let us finally define the initial conditions for the URS. In Equation 5-52, we have seen that the states depend on several previous values, namely $d_{j-1}(k)$, $d_j(k-1)$, $\rho_{j-1}(k)$ and $\sigma_j(k-1)$. We therefore need to design the controller starting from $j = 2$ and $k = 2$, with therefore $j = 2, ..., J_c + 1$ and $k = 2, ..., K_c + 1$. We can then consider the initial values where $j = 1$ and $k = 1$ as parameters going in the controller. The same holds for the simulation which will run for $j = 2, ..., J+1$ and $k = 2, ..., K+1$. Let us therefore assign values to the initial conditions $d_1(k)$, $d_j(1)$, $\rho_1(k)$ and $\sigma_j(1)$:

$$\begin{aligned} d_1(k) &= (k-1) \cdot 120 && \forall k = 1, ..., K+1 \\ d_j(1) &= (j-1) \cdot 240 && \forall j = 1, ..., J+1 \\ \rho_1(k) &= 120 && \forall k = 1, ..., K+1 \\ \sigma_j(1) &= 0 && \forall j = 1, ..., J+1 \end{aligned} \tag{5-73}$$

In other words: each initial train arriving at the first station contains 120 passengers and every initial platform is considered empty. We will see in the next section that these initial conditions are important for simulations. Assigning the 'wrong' initial conditions could result in a system that is unfeasible.

5-5 Simulations

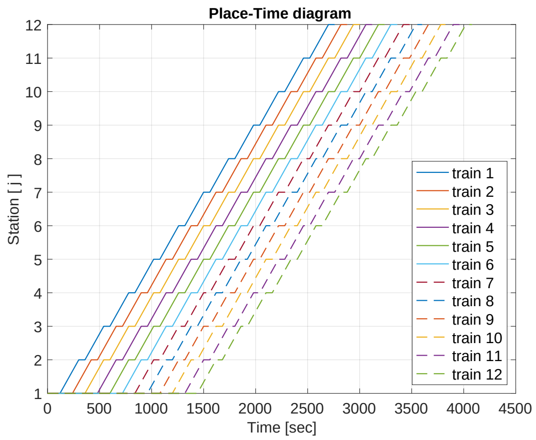
We have assigned all modeling parameters and initial conditions, and are therefore ready to actually solve the optimization problem and obtain simulations. The MPC design will be done in MATLAB using the YALMIP toolbox and GUROBI as optimization solver. We will first analyze the URS without any disturbance or control and therefore without inputs $u_{1,j}(k)$ and $u_{2,j}(k)$. We will investigate the importance of choosing the correct initial conditions. Thereafter, we will add different types of disturbance, and we will investigate the performance of the MPC.

5-5-1 No Control - Initial Conditions

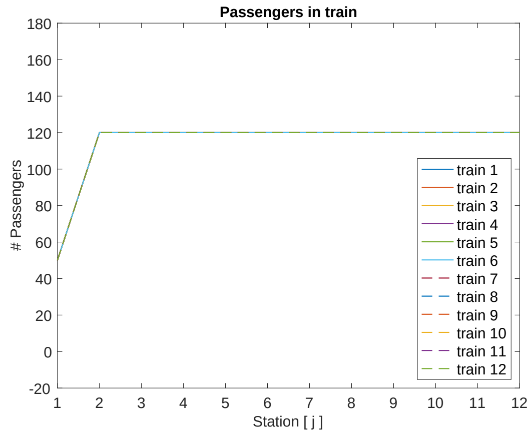
Let us investigate how the train schedule of the URS develops without any control or disturbance. We have used the modeling parameters and initial conditions as defined in the previous section. In Figure 5-1a, the schedule of the trains and the passengers in the train can be observed. In Figure 5-1b shows the number of passengers in the trains at moment of departures, Figure 5-1c shows the passengers boarding, and Figure 5-1d the passengers left behind. Finally in Figure 5-1e, the optimization variable is shown, defined by the number of passengers waiting. As no control is added and we are not considering an optimization problem, $p_j^{\text{wait}}(k)$ is just a state. It can be observed that the initial values for the URS result in a constant or steady-state development of the states of the URS, with as a result:

- A constant dwell time $(d_j(k) - a_j(k))$;
- A stabilized number of passengers in the train ($\rho_j(k) = 120$ passengers);
- A stabilized number of passengers boarding ($q_j(k) = 60$ passengers);
- No passengers left behind ($\sigma_j(k) = 0$ passengers);
- A constant number of passengers waiting ($p_j^{\text{wait}}(k) = 30$ passengers).

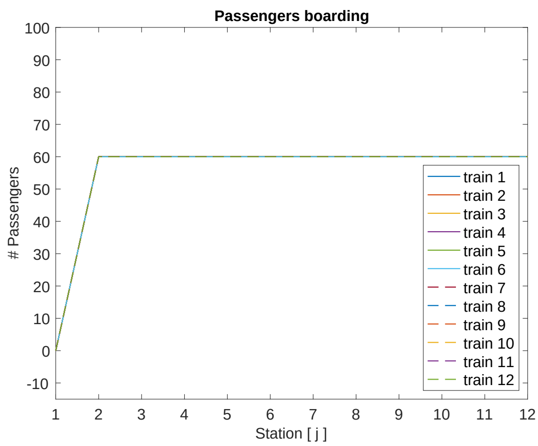
The number of passengers waiting is not zero as it also contains the number of passengers arriving on the platform on the time indices that no train is present, defined by $e(a_j(k)-d_j(k-1))$ with e a continuous inflow of passengers. Note that therefore the number of passengers waiting can never be zero, because there will always be a 'dead' moment without any train present, because of the definition of the arrival time including the headway time. These initial values can be considered as a equilibrium or steady-state condition of the URS.



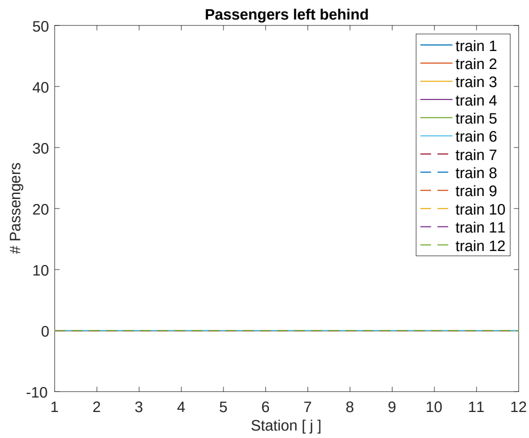
(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$



(b) $\rho_j(k)$

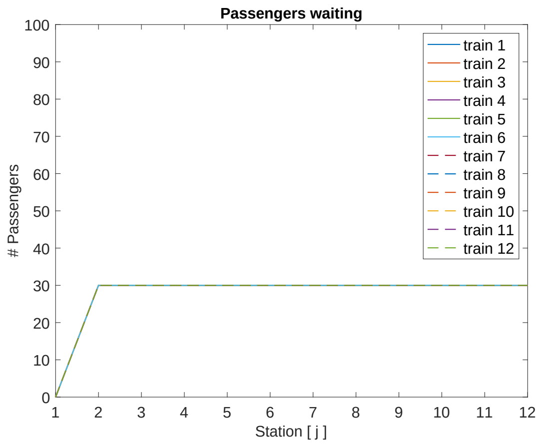


(c) $q_j(k)$



(d) $\sigma_j(k)$

See next page for complete figure description



(e) $p_j^{\mathsf{wait}}(k)$

Figure 5-1: States of the URS for the initial values and parameters as defined in Section 5-4 without control implemented, for J = 12 and K = 12.

This steady-state condition we have seen in Figure 5-1 is characterized by the following requirements:

1. Number of passengers disembarking = number of passengers boarding:

$$\beta \rho_{j-1}(k) = q_j(k) \tag{5-74}$$

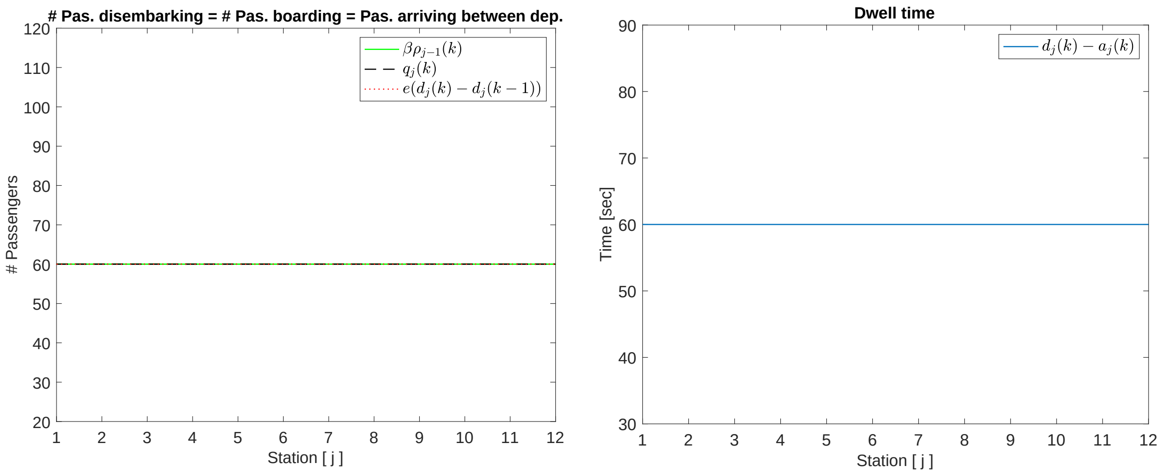
2. Number of passengers arriving in between departures of consecutive trains = number of passengers boarding:

$$e(d_j(k) - d_j(k-1)) = q_j(k) \tag{5-75}$$

3. Constant dwell time:

$$d_j(k) - a_j(k) = d_{j-1}(k) - a_{j-1}(k) = d_j(k-1) - a_j(k-1) \tag{5-76}$$

In Figure 5-2, it can be observed that the above requirements for the URS running in steady-state are indeed fulfilled for the initial values as defined in section 5-4. The number of passengers disembarking, the number of passengers boarding and the number of passengers arriving between consecutive trains are equal at every station, as shown in Figure 5-2a. And furthermore, each train has a constant dwell time at every station, as shown in Figure 5-2b.



(a) Steady-state conditions 1. and 2.

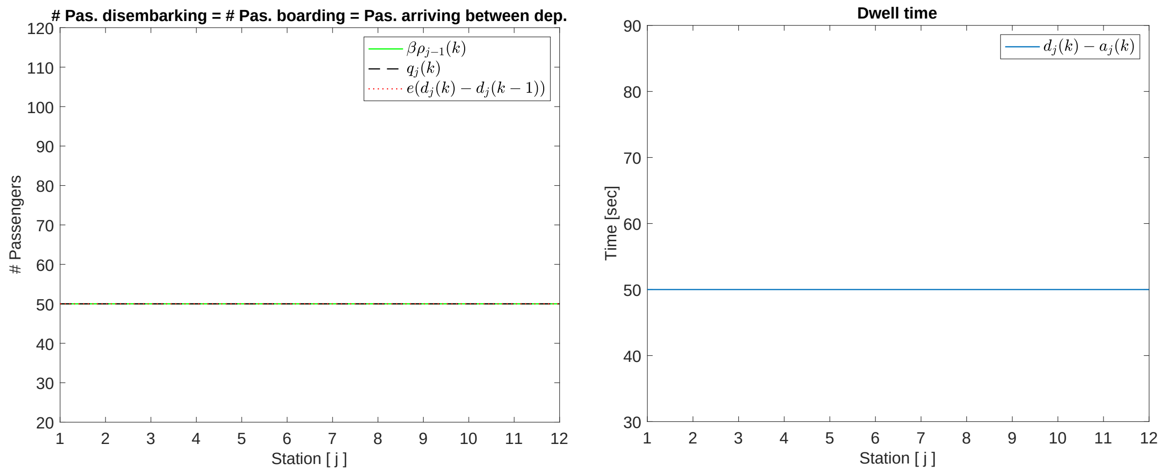
(b) Steady-state condition 3.

Figure 5-2: Steady-state conditions for initial values as defined in section 5-4, for $J = 12$ and $K = 12$.

Let us show that changing the initial states to values that still guarantee the above three necessary conditions. In Figure 5-3, the steady-state conditions of the URS can be observed for the following set of initial values:

$$\begin{aligned} d_1(k) &= (k-1) \cdot 100 & \forall k &= 1, ..., K+1 \\ d_j(1) &= (j-1) \cdot 200 & \forall j &= 1, ..., J+1 \\ \rho_1(k) &= 100 & \forall k &= 1, ..., K+1 \\ \sigma_j(1) &= 0 & \forall j &= 1, ..., J+1 \\ \tau_r &= 150 \end{aligned} \tag{5-77}$$

The remaining parameter values are kept equal to the values defined in section 5-4. In the resulting URS, 50 passengers board and 50 passengers disembark at every station for every train. The number of passengers in the train therefore is stabilized at $\rho_j(k) = 100$ passengers (which can be observed in Figure B-1c), and the dwell time at $d_j(k) - a_j(k) = 50$ seconds as shown in Figure 5-3b.



(a) Steady-state conditions 1. and 2.

(b) Steady-state condition 3.

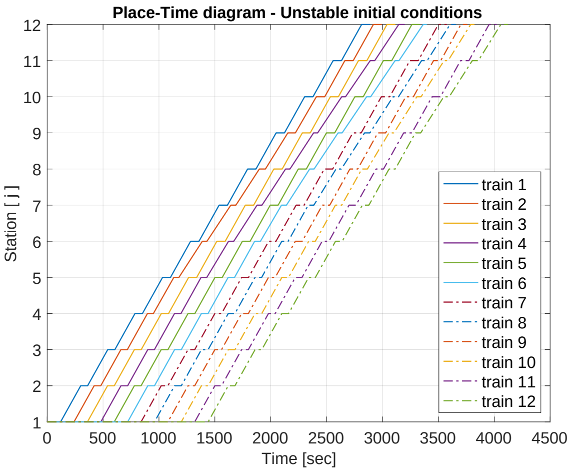
Figure 5-3: Steady-state conditions for initial values as defined in Equation 5-77 without control implemented, for $J = 12$ and $K = 12$.

It can be seen that we indeed fulfill the three conditions that ensure that the URS develops in steady-state. In Appendix B in Figure B-1e, it can furthermore be observed that the number of passengers left behind $p_j^{\text{wait}}(k)$ stabilizes at 25 passengers, the number of passengers that arrives within departure and arrival of consecutive trains.

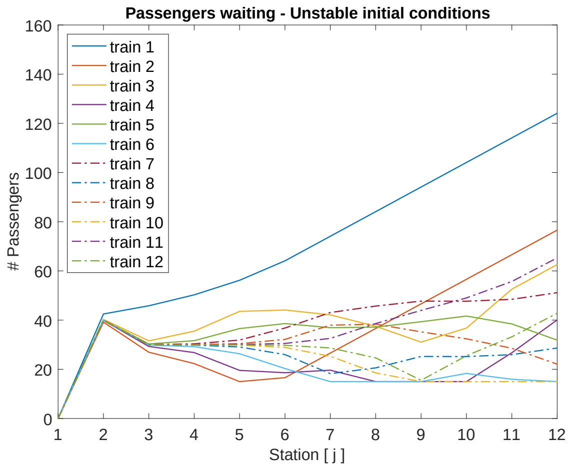
The initial conditions thus are of great importance for the URS. If the initial values are not exactly representing a steady-state condition for the URS, the resulting time-table will be irregular and the number of passengers waiting as well as the number of passengers left behind builds up. The choice of parameters β and e representing the fraction of passengers disembarking and speed of passengers arriving at the station ([pas./sec]) respectively is furthermore important, as these parameters have an influence on conditions 1. and 2. as can be observed in Equation 5-74 and 5-75. To show this importance, let us simulate a case where the initial values are chosen such that the steady-state requirements do not hold. Let us consider the following initial conditions:

$$\begin{aligned}
d_1(k) &= (k-1) \cdot 120 && \forall k = 1, ..., K+1 \\
d_j(1) &= (j-1) \cdot 235 && \forall j = 1, ..., J+1 \\
\rho_1(k) &= 100 && \forall k = 1, ..., K+1 \\
\sigma_j(1) &= 10 && \forall j = 1, ..., J+1 \\
\tau_r &= 180
\end{aligned} \tag{5-78}$$

And the remaining parameter values are kept equal to the values as defined in section 5-4.



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$



(b) $p_j^{\text{wait}}(k)$

Figure 5-4: Place-time diagram and number of passengers waiting of the URS with the initial values as defined in Equation 5-78 without control implemented, for J = 12 and K = 12.

It can be observed that although the place-time diagram seems relatively stable, the number of passengers waiting has a positive growth rate for several trains. Because of this, these numbers of passengers waiting will grow to infinity over infinite number of stations, and are therefore unstable. In Figure B-2 it can furthermore be observed that the steady-state conditions do not hold, as we have a changing number of passengers boarding and arriving in between consecutive trains. For the remainder of the chapter and the simulations in the following subsections, we will use the initial conditions and parameter values as defined in section 5-4.

5-5-2 Control - Disturbance Scenarios

We are now interested in investigating the performance of the MPC. We will investigate different scenarios which represent situations that can occur during the run of a real URS. During all simulations, we will minimize the objective function as defined in Equation 5-61 subject to the constraints defined in subsection 5-3-3. As we are considering the initial conditions and parameter values as defined in section 5-4, we can assign the reference value $p_{\text{ref}}^{\text{wait}}$ introduced in Equation 5-61. We will set this value on 30 passengers, as we have seen in Figure 5-1e that for the initial conditions and parameter values as defined in section 5-4, $p_j^{\text{wait}}(k)$ stabilizes at this value. Thus, the objective function considered in the following subsections is defined as:

$$J(k) = ||p_j^{\text{wait}}(k) - 30||_1 + 0.1||u_{1,j}(k)||_1 + 0.1||u_{2,j}(k)||_1 \tag{5-79}$$

Stabilizing the number of passengers waiting around this steady-state value is preferred over stabilizing around zero as the URS then will be steered to its steady-state mode. For each type of disturbance, we will compare the MPC performance with the simulations of the URS without control.

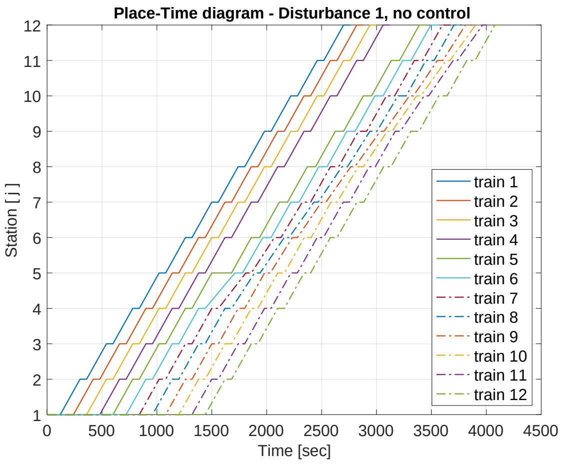
Decrease in Boarding Speed

The first type of disturbance we will include, is an decrease in boarding speed. We will investigate two scenarios. The first scenario considers a sudden decrease in boarding speed for a particular train at a particular station. This can occur in a case where several passengers of senior age with lower physical mobility are boarding the train at the particular station. We will include this disturbance in the following manner:

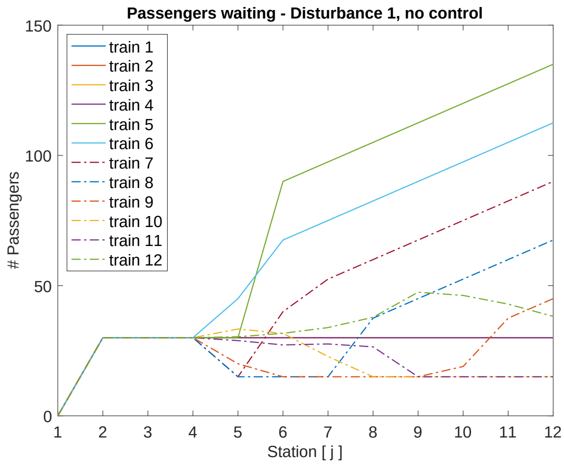
Disturbance 1.

$$\text{for} \quad k = 5,\ j = 5: \quad b = 0.6 \text{ pas./sec} \tag{5-80}$$

In Figure 5-5[1], the place-time diagram and the number of passengers waiting for all trains at all stations can be observed for the URS without any control implemented. It can be seen in Figure 5-5a that the train schedule is not operating in steady-state anymore, and no longer has a constant dwell time. It can furthermore be observed in Figure 5-5b that due to the disturbance, the number of passengers waiting increases and will remain to increase for increasing number of stations. Let us compare these simulations with the URS subject to the same disturbance, but now with the implemented MPC.



(a) $a_j(k)$, $\tau_{\mathrm{dwell},j}(k)$ and $d_j(k)$



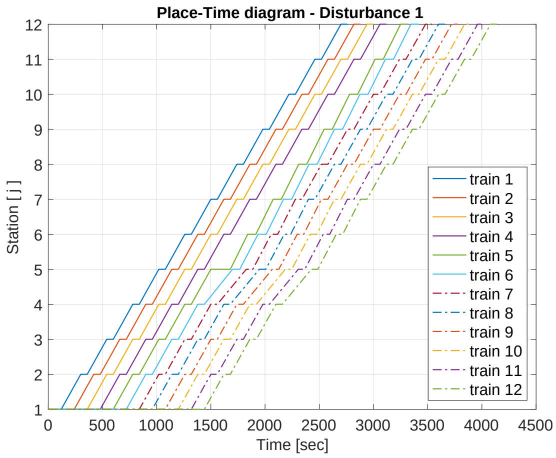
(b) $p_j^{\mathrm{wait}}(k)$

Figure 5-5: Place-time diagram and number of passengers waiting of the URS with Disturbance 1 without control for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

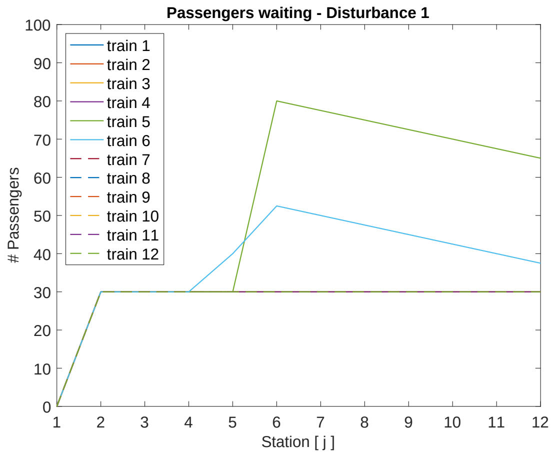
Figure 5-6a shows the place-time diagram of the URS subject to Disturbance 1 and the MPC. It can be observed that the train schedule is still not operating in steady-state mode with a constant dwell time. However, we see in Figure 5-6b that the number of passengers

[1] For simplicity, we have considered all states of the URS to be symbolic decision variables or `sdpvar` $\in \mathbb{R}$ in Matlab, instead of the states representing numbers of passengers being integer decision variables or `intvar` $\in \mathbb{Z}$. After computing the values of the states in the simulations, these values are rounded to the nearest integer. Note that we could also consider the states to be `intvar`, but the results will be more chaotic. Another option is to include a delay time input variable on the departure time, which can easily regulate the URS such that the states representing $q_j(k), \rho_j(k), \sigma_j(k)$ and $p_j^{\mathrm{wait}}(k)$ are integers, resulting in similar results. However, to simplify the optimization problem, we have decided to use `sdpvar`.

waiting $p_j^{\text{wait}}(k)$ is stabilized by the implemented control. Over a longer horizon $J > 12$, the $p_j^{\text{wait}}(k)$ will eventually be steered back to the target steady-state value of 30 passengers. In subsection B-2-1, the remaining states and the implemented control effort can be observed. This control action forces the system to use the maximum capacity $\rho_{\max}$ for trains $k = 5$ and $k = 6$, to minimize the passengers left behind. The trains thereafter are again running at steady-state condition. The MPC therefore succeeds in stabilizing the URS for Disturbance 1.



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$



(b) $p_j^{\text{wait}}(k)$

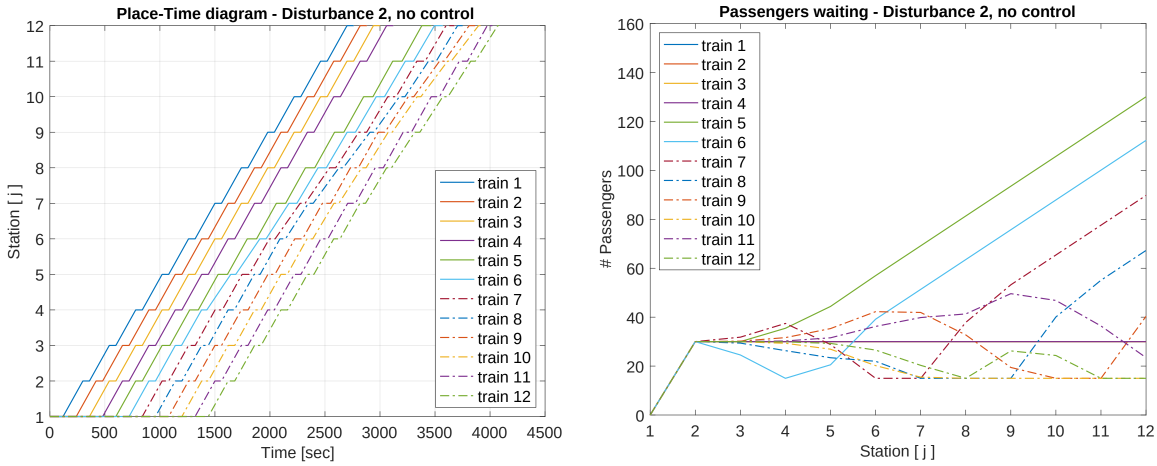
Figure 5-6: Place-time diagram and $p_j^{\text{wait}}(k)$ of the URS with Disturbance 1 with the MPC for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

A second scenario we will observe, is a decrease of boarding speed for particular train over its complete route (for all stations). A realistic example of such a case is if the train is considered to have less train compartments than other trains and therefore has less doors available for boarding and disembarking. This disturbance is defined as:

Disturbance 2.

$$\text{for} \qquad k = 5,\ \forall j \in J \qquad b = 1.6 \text{ pas./sec} \tag{5-81}$$

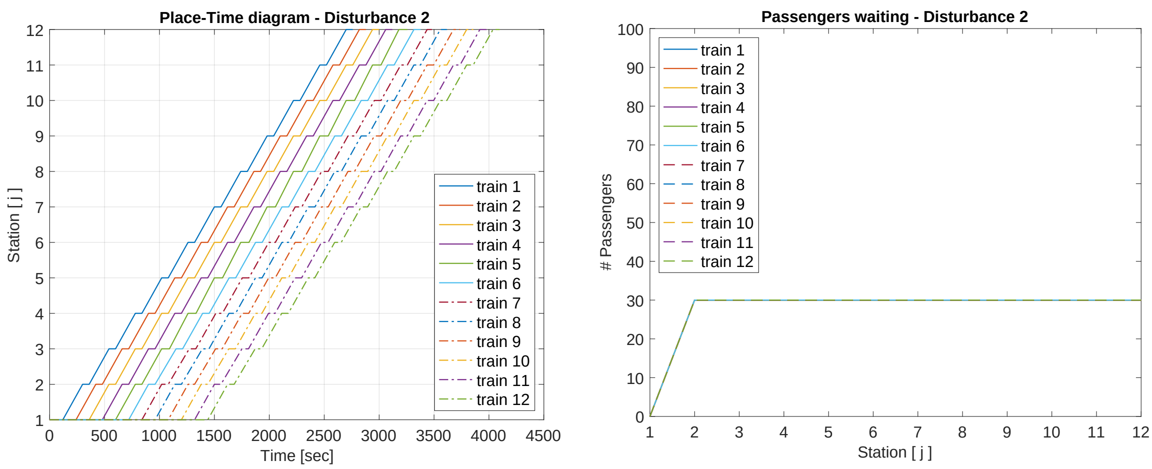
When no control is applied, the decrease in boarding speed will clearly result in a longer dwell time, as it simply takes longer to let an equal number of passengers board. As there is a constant rate of passengers arriving at the platform (e), the number of passengers waiting on the platform will increase. Figure 5-7 shows the place-time diagram and the number of passengers waiting for the URS subject to Disturbance 2 without control. It can again be observed that $p_j^{\text{wait}}(k)$ increases and remains to increase for several trains (train 5 until train 9).



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$ **(b)** $p_j^{\text{wait}}(k)$

Figure 5-7: Place-time diagram and number of passengers waiting of the URS with Disturbance 2 without control for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

Let us compare this result to the URS with the implemented MPC subject to Disturbance 2 in the figure thereafter. If we compare Figure 5-7a with Figure 5-8a, it can be observed that due to the implemented control effort, a stable train schedule with a constant dwell time is computed. Furthermore, if we compare Figure 5-7b with Figure 5-8b, it can be observed that the MPC succeeds in stabilizing the URS without any deviation of the target value of $p_j^{\text{wait}}(k) = 30$ passengers. Thus, the MPC again succeeds in stabilizing the URS. The evolution of the remaining states and the implemented control effort can be observed in subsection B-2-2.



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$ **(b)** $p_j^{\text{wait}}(k)$

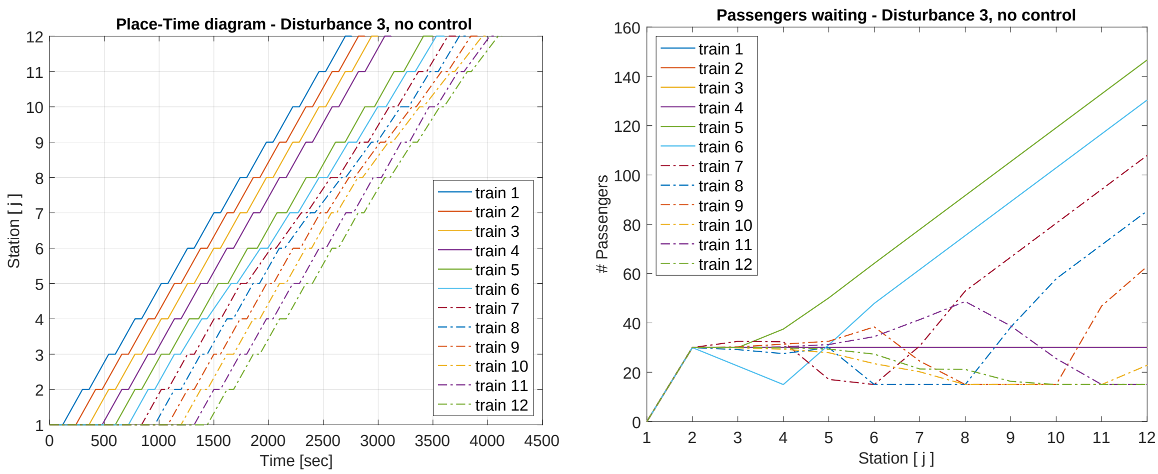
Figure 5-8: Place-time diagram and number of passengers waiting of the URS with Disturbance 2 with the MPC for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

Finally, let us investigate one more disturbance variation on the decrease in boarding speed, namely:

Disturbance 3.

$$\text{for} \quad k = 5, \ \forall j \in J \quad b = 1.5 \text{ pas./sec} \tag{5-82}$$

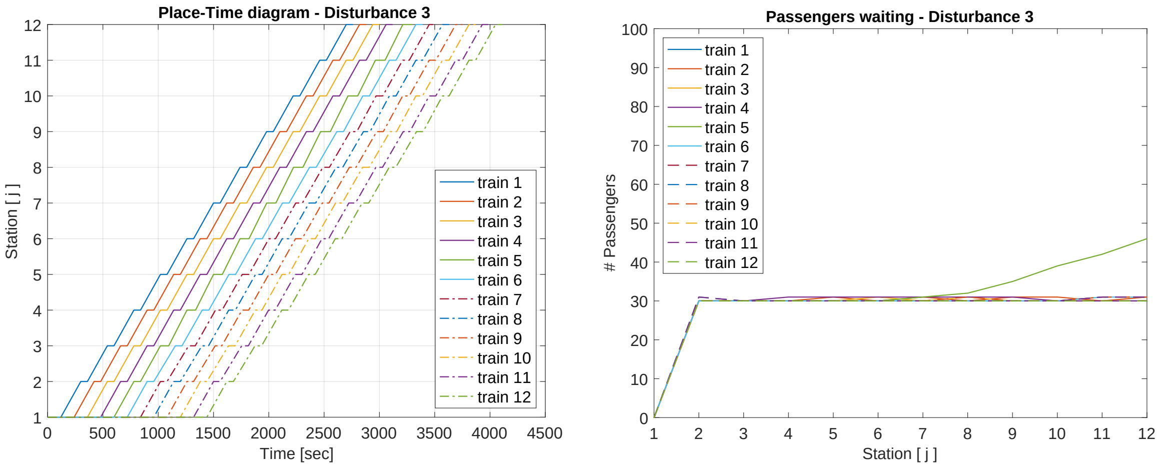
Note that there is only a difference of 0.1 passengers per second compared to Disturbance 2. Let us first present the URS subject to this disturbance without control. It can be observed that the URS reacts similar to Disturbance 3 as to Disturbance 2, with a small overall increase in passengers waiting.



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$ **(b)** $p_j^{\text{wait}}(k)$

Figure 5-9: Place-time diagram and number of passengers waiting of the URS with Disturbance 3 without control for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

Let us observe the place-time diagram and number of passengers waiting of the URS subject to Disturbance 3, now including the control effort. It can be observed in Figure 5-10a that the MPC effort results in a stable train schedule with constant dwell time. However, the MPC does not succeed in stabilizing the number of passengers waiting, as can be observed in Figure 5-10b. Here we see that for train 5 (which is subject to Disturbance 3), $p_j^{\text{wait}}(k)$ is growing for stations $j \geq 7$. In subsection B-2-3, the remaining states of the URS and the control effort can be observed from which we can conclude the following. From station 9 on, the maximum capacity of passengers is reached in the trains, but the maximum control effort is also applied. This maximum capacity is the reason that the MPC cannot stabilize the URS anymore; the passengers are boarding too slow for too long, resulting in a longer dwell time and therefore more passengers arriving on the platforms. As the control effort is already operating at boundary level we cannot compensate more, resulting in an accumulation of passengers left behind and therefore passengers waiting. Thus, we can conclude that the MPC does not succeed in stabilizing the URS subject to Disturbance 3. This situation will be explained more thoroughly in the next section about closed-loop stability of the URS.



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$ **(b)** $p_j^{\text{wait}}(k)$

Figure 5-10: Place-time diagram and number of passengers waiting of the URS with Disturbance 3 with the MPC for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

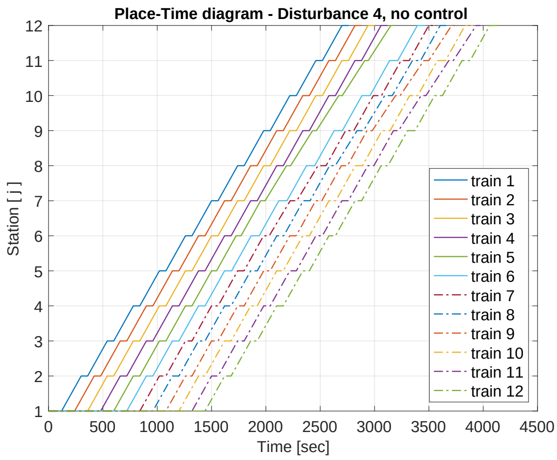
Decrease in Arrival Speed

Let us now investigate the performance of the MPC while the URS is subjected to a decrease of value e, the speed of passengers arriving at the platform. Realistically, this can happen at a platform that is less popular and on a remote location. We will first consider the following disturbance:

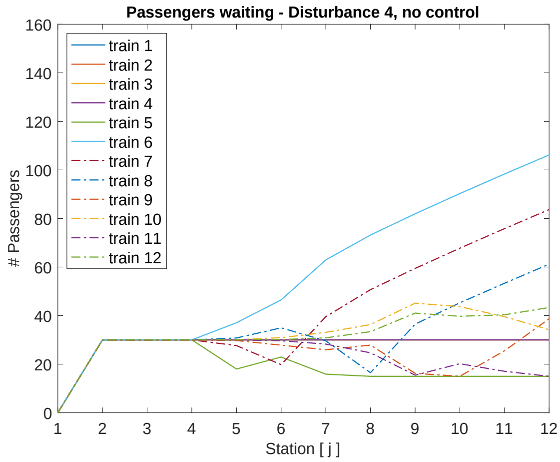
Disturbance 4.

$$\text{for} \quad k = 5,\ j = 5: \quad e = 0.3 \text{ pas./sec} \tag{5-83}$$

A decrease in arrival speed e results in less passengers arriving on the platform, and therefore less passengers wanting to board. The trains will therefore be able to depart sooner, as the dwell time will also decrease. Let us first observe how the states of the URS develop without control implemented. We can observe in Figure 5-11a that train $k = 5$ indeed has a shorter dwell time from station $j = 5$ on. The shorter dwell time for the stations after $j = 5$ can be explained by the following; as less passengers board at station $j = 5$, less passengers disembark on the stations thereafter as a fixed fraction disembarks (β). This irregularity also causes the unstable $p_j^{\text{wait}}(k)$ in Figure 5-11b.



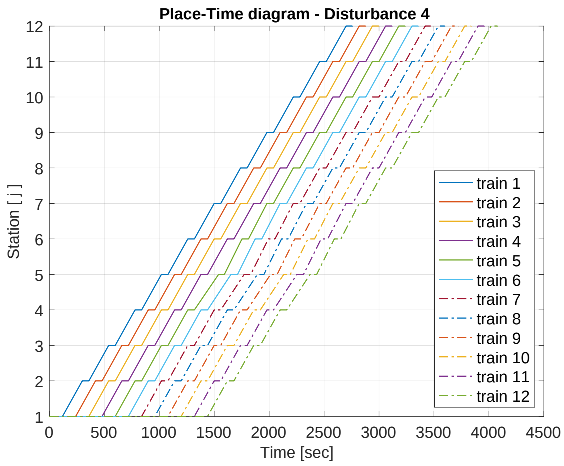
(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$



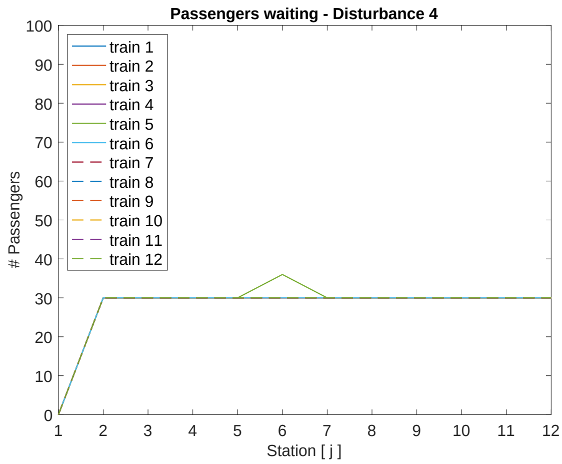
(b) $p_j^{\text{wait}}(k)$

Figure 5-11: Place-time diagram and number of passengers waiting of the URS with Disturbance 4 without control for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

Let us now investigate whether the MPC succeeds in stabilizing the URS subject to Disturbance 4. In Figure 5-12a, it can be observed that the resulting train schedule is operating in steady-state. Furthermore, Figure 5-12b shows that the MPC successfully stabilizes the number of passengers waiting, with a small discontinuity for train 6 as a result of the disturbance. More details on the state evolution and control action can be found in subsection B-2-4.



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$



(b) $p_j^{\text{wait}}(k)$

Figure 5-12: Place-time diagram and number of passengers waiting of the URS with Disturbance 4 with the MPC for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

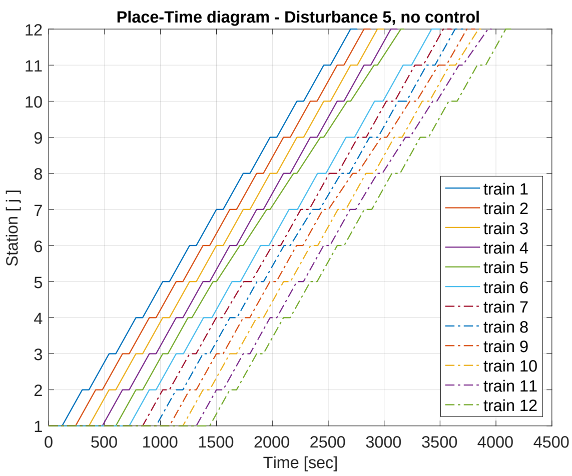
Let us now implement a disturbance on the speed of passengers arriving at the platforms for the complete route of train $k = 5$, resulting in:

Disturbance 5.

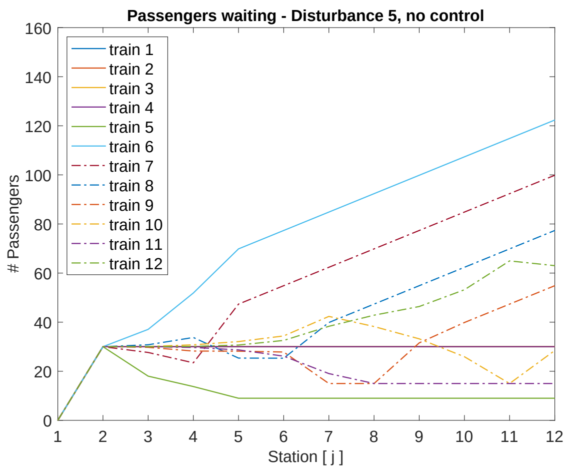
$$\text{for} \qquad k = 5,\ \forall j \in J \qquad e = 0.3 \text{ pas./sec} \tag{5-84}$$

Such a situation occurs for example if train $k = 5$ is running at an off-peak moment such as late evening hours. This disturbance results in the URS schedule and number of passengers waiting throughout the route as shown in Figure 5-13, if no control is implemented. It can be observed that Disturbance 5 results in a slight overall increase in number of passengers waiting compared to Disturbance 4. This can be explained by the following: we have seen that to obtain a steady-state running URS, the number of passengers we need the number of passenger arriving in between departures of consecutive trains to be equal to the number of passengers disembarking. However, due to the disruption of the value of e, these numbers are not equal anymore, resulting in an irregular dwell time. Disturbance 5 can therefore be considered as a more disruptive disturbance version of Disturbance 4.

Let us observe whether the MPC is able to stabilize the URS subject to this disturbance. It can be observed in Figure 5-14a that the dwell time is not constant for all trains at every station. However, this train schedule is still operating close to steady-state. Furthermore, the MPC succeeds in stabilizing $p_j^{\text{wait}}(k)$, as can be observed in Figure 5-14b. If we compare Figure 5-14b with Figure 5-12b, it can be observed that although Disturbance 5 is more disruptive, its effect on $p_j^{\text{wait}}(k)$ is less; in Figure 5-14b, there is no discontinuity present. This can however be explained by the number of passengers boarding ($q_j(k)$) and number of passengers in the train ($\rho_j(k)$); these states stabilize at 43 and 86 passengers respectively for Disturbance 5 as can be observed in Figure B-12b, instead of 60 and 120 passengers respectively for Disturbance 4. Due to these changes, the MPC succeeds in keeping $p_j^{\text{wait}}(k)$ stable during the complete route for Disturbance 5.

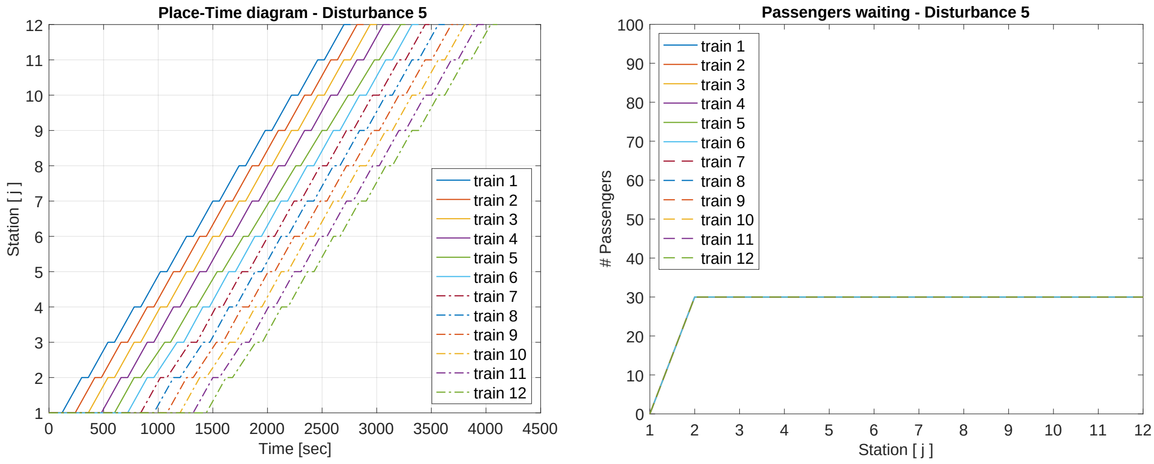


(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$



(b) $p_j^{\text{wait}}(k)$

Figure 5-13: Place-time diagram and number of passengers waiting of the URS with Disturbance 5 without control for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$ **(b)** $p_j^{\text{wait}}(k)$

Figure 5-14: Place-time diagram and number of passengers waiting of the URS with Disturbance 5 with the MPC for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

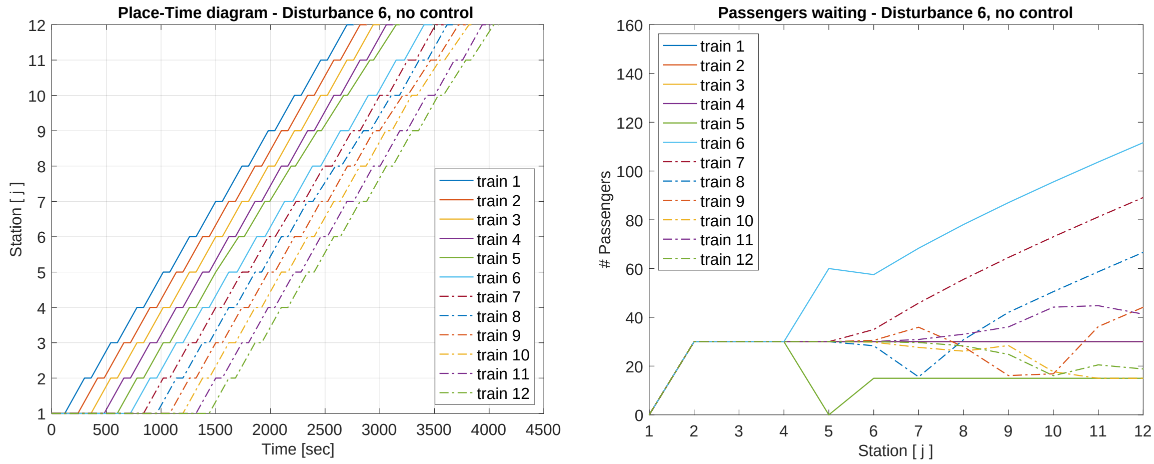
No Minimum Dwell Time

The last situation we will observe, is a scenario in which a train will skip a stop. To mimic such a situation, let us consider the following disturbance:

Disturbance 6.

$$\text{for} \quad k = 5,\ j = 5 \quad e = 0 \text{ pas./sec},\ \beta = 0 \tag{5-85}$$

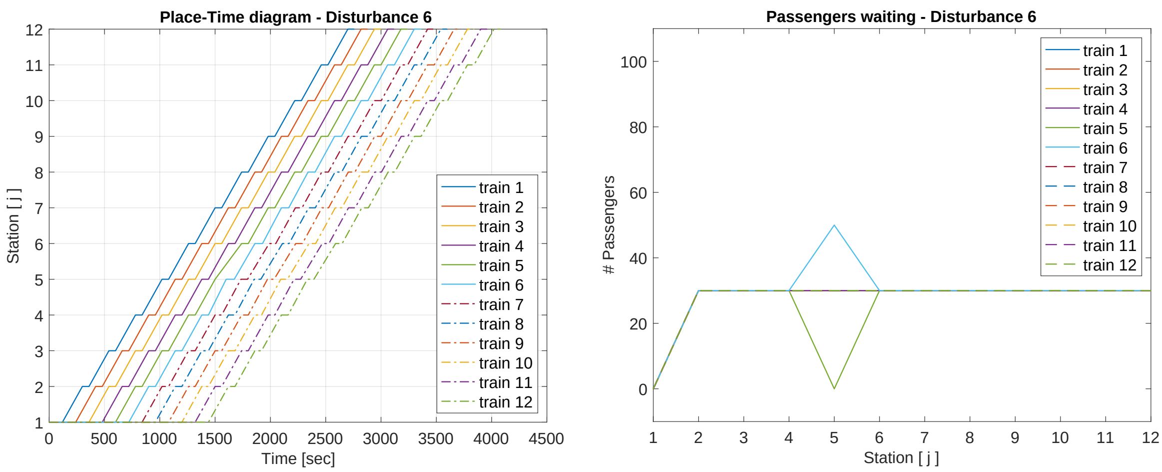
Note that in order to provide the possibility to skip a stop, we must change the minimum dwell time $\tau_{\text{dwell,min}} = 30$ seconds to $\tau_{\text{dwell,min}} = 0$ seconds. Let us first observe again what happens for the URS without any control implemented.



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$ **(b)** $p_j^{\text{wait}}(k)$

Figure 5-15: Place-time diagram and number of passengers waiting of the URS with Disturbance 6 without control for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12 and a minimum dwell time set on $\tau_{\text{dwell,min}} = 0$ seconds.

It can be observed that train $k = 5$ indeed skips station $j = 5$, but due to this irregularity, we again obtain an increasing number of passengers waiting for the later trains. The MPC succeeds to stabilize this, as can be observed in the following figures.



(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$ **(b)** $p_j^{\text{wait}}(k)$

Figure 5-16: Place-time diagram and number of passengers waiting of the URS with Disturbance 6 with the MPC for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

We see in Figure 5-16a that train $k = 5$ again skips the stop at station $j = 5$. This results in an increase in passengers waiting on train $k = 6$, but it can be observed that the MPC covers for these passengers, and stabilizes $p_j^{\text{wait}}(k)$ from station $j = 6$ on. The remaining states and implemented control effort can be found in subsection B-2-6. Note that including a fixed minimum dwell time has advantages and disadvantages. It is useful from a passenger point-of-view, as passengers will always get the opportunity to board or disembark. However, if the fixed minimum dwell time is longer than the actual dwell time, the URS will wait longer than necessary. Because the URS is now defined in a manner that there is a continuous flow of passengers arriving at the platform, this extra and unnecessary dwell time will result in a less efficient schedule than possible. Including a fixed minimum dwell time or not can also be included with a switching strategy, for which more research is necessary.

5-6 Closed-Loop Stability Analysis

In this section, we will investigate the closed-loop stability of the URS subject to MPC as an extension of the results of the previous section. In section subsection 5-6-1, we will elaborate on the definition of closed-loop stability for Discrete Time Systems (DTS) and DES systems, and thereafter in subsection 5-6-2 we will analyze the closed-loop stability performance of the URS.

5-6-1 Closed-Loop Stability of Discrete Event Systems

We have seen in section 5-1 that MPC is a popular control method, often implemented on DTS. It is however not guaranteed that the MPC approach always results in closed-loop stable systems. Let us shortly introduce some background information on stability. Let us define the general idea of stability of time-based system:

Definition 5.3 (Bounded-Input Bounded-Output Closed-Loop Stability)**.** *A time-based system is considered closed-loop stable if, when subjected to control, a bounded input results in a bounded output.*

Such a bounded input can also be considered as a disturbance; stability is a measure of tendency of the system's response to return to its original (or a target) state after being exposed to disturbance. A system can return to such a state without control due to its dynamics (such as a pendulum that is pushed), or with control (such as a heat regulation system). The URS however is considered to be a DES, and the above stability definition cannot directly be extended to such systems. This is due to the difference in evolution of states in time-based systems versus states in event-based systems. Examples of states in time-based systems are characteristics of a system such as speed, acceleration and temperature. Clearly, these are examples of states that can be stabilized at a fixed value. However, we have seen that states in event-based systems such as DES represent the occurrence of events. In the URS for example, the states $a_j(k)$ and $d_j(k)$ represent the arrival and departure time of train k from station j. The values of these states are non-decreasing, as they represent time instances. As it is not possible or logical to stabilize such a state at a fixed value, we therefore define another interpretation for stability of DES; a DES is stable if its buffer levels are stable

[41]. Buffer levels of DES are defined as the time delay between the occurrences of events in the same event cycles (k) or consecutive event cycles ($k-1$ and k). Stable buffer levels can be obtained if the growth rates of the states are stable. The growth rate can be defined as the difference between consecutive states. For stability analysis for closed-loop time-based systems it is common to use Lyapunov stability theory, of which more information can be found in [42]. This Lyapunov stability theory is extended for DES in [43]. In this thesis, we will not dive into the mathematical background on the stability analysis of the URS, but only make several important remarks.

5-6-2 Stability of the Urban Railway System

We have seen in the previous subsection that the meaning of stability for DES is different from the general stability definition for DTS. Let us now define stability for the URS specifically. We have already seen that for the states representing the arrival and the departure time, it is required to have a stable growth rate for stability. However, the remaining states all represent number of passengers. For the latter states, it makes more sense to define stability equivalently as for time-based systems; these states are considered stable if they stabilize at a certain value.

We have seen in subsection 5-5-1 that if we choose the correct initial state values and parameter values, we can obtain an URS in a steady-state condition. In this steady-state condition, the URS is considered stable as the arrival and departure time have a fixed growth rate, and the remaining states stabilize at a fixed value (for example $\rho_j(k)$ stabilizes at 120 in Figure 5-1b). As the objective is to minimize the passengers waiting on the platform at moment of arrival of a train, we therefore measure the performance of the MPC to the degree of ability to stabilize this number $p_j^{\text{wait}}(k)$. We have seen in the formulation of the objective function in Equation 5-79, that we aim to stabilize this value around the value $p_{\text{ref}}^{\text{wait}} = 30$. For the initial conditions and assigned parameter values, the URS is then considered to be stable. Let us formally propose the closed-loop stability of the URS subject to the designed MPC, based on [38]:

Proposition 5.1 (Closed-Loop Stability of the URS). *Let $x_{2,e} - x_{1,e}, x_{3,e}, x_{4,e}, x_{5,e}$ and $x_{6,e}$ be steady-state values for the URS for which the states evolve in a constant manner, while satisfying all the constraints defined in subsection 5-3-3. Let furthermore $x_{1,0}, x_{2,0}, x_{3,0}, x_{4,0}, x_{5,0}$ and $x_{6,0}$ be admissible initial state values for which the constraints hold. The MPC stabilizes the URS if the following holds:*

$$\begin{aligned}
&\lim_{j\to\infty}\lim_{k\to\infty} ||(x_{2,j}(k) - x_{1,j}(k)) - (x_{2,e} - x_{1,e})||_1 = 0\\
&\lim_{j\to\infty}\lim_{k\to\infty} ||x_{3,j}(k) - x_{3,e}||_1 = 0\\
&\lim_{j\to\infty}\lim_{k\to\infty} ||x_{4,j}(k) - x_{4,e}||_1 = 0\\
&\lim_{j\to\infty}\lim_{k\to\infty} ||x_{5,j}(k) - x_{5,e}||_1 = 0\\
&\lim_{j\to\infty}\lim_{k\to\infty} ||x_{6,j}(k) - x_{6,e}||_1 = 0
\end{aligned} \tag{5-86}$$

And furthermore all constraints and system dynamics hold as defined in subsection 5-3-3.

We have seen in the figures in subsection 5-5-2 that the MPC succeeds to stabilize the URS for all disturbances, except for Disturbance 3. We have therefore seen that it depends on the values of the disturbances added to the URS whether the MPC stabilizes the system. We aim to find the set of all disturbances for which the MPC does succeed in stabilizing the URS. We will define this disturbance set as the following:

Definition 5.4 (Admissible Disturbance Set)**.** *We define $\mathcal{D}_{ad}$ as the set of admissible disturbances that can be added to the URS system, such that the MPC succeeds in stabilizing the URS. Then Equation 5-86 holds.*

In order for a disturbance to be admissible and therefore contained in set $\mathcal{D}_{\text{ad}}$, the URS subject to the disturbance has to meet the following condition: the total number of passengers boarding over complete train route ($j = [2 \ \dots \ J]$) must be bigger than or equal to the total number of passengers arriving in between departures of consecutive trains over the complete train route ($j = [2 \ \dots \ J]$) for all trains $k \in K$:

$$\sum_{j=2}^{J} \sum_{k=2}^{K} q_j(k) \geq \sum_{j=2}^{J} \sum_{k=2}^{K} e_j(d_j(k) - d_j(k-1)) \tag{5-87}$$

This condition make sense realistically; if more passengers keep arriving than can board, the number of passengers left behind builds up. For overall closed-loop stability we have considered the limits up until infinity in Proposition 5.1. However for the URS let us consider the limits for finite number of stations J and trains K for simplicity. Let us investigate[2] whether this condition is indeed violated in the evolution of the URS subject to Disturbance 3:

$$\begin{aligned} &\sum_{j=2}^{J} \sum_{k=2}^{K} q_j(k) = 8781 \text{ passengers} \\ &\sum_{j=2}^{J} \sum_{k=2}^{K} e_j(d_j(k) - d_j(k-1)) = 8781 \text{ passengers} \end{aligned} \tag{5-88}$$

In the overall evolution of the URS, the condition in Equation 5-87 seems not to be violated. However, for the URS subject to Disturbance 3 we have seen that the steady-state condition is not obtained for train $k = 5$ specifically. For this train the number of passengers waiting ($p_j^{\text{wait}}(k)$) cannot be stabilized, as shown in Figure 5-10b. Let us investigate the condition for the admissible disturbance set for this train $k = 5$ specifically:

$$\begin{aligned} &\sum_{j=2}^{J} q_j(5) = 861 \text{ passengers} \\ &\sum_{j=2}^{J} e_j(d_j(5) - d_j(4)) = 911 \text{ passengers} \end{aligned} \tag{5-89}$$

It can be observed that for the complete route of train $k = 5$, the condition is indeed violated, as we have:

$$\sum_{j=2}^{J} q_j(5) = 861 < 911 = \sum_{j=2}^{J} e_j(d_j(5) - d_j(4)) \tag{5-90}$$

[2] We have rounded the values obtained in the simulation to the nearest integer to obtain these numbers of passengers. See Footnote 1 for more details.

If we compare the condition values of the URS subject to Disturbance 2, for which the MPC succeeds in stabilizing $p_j^{\text{wait}}(k)$ as can be observed in Figure 5-8b, we have for train $k = 5$:

$$\begin{aligned} &\sum_{j=2}^{J} q_j(5) = 805 \text{ passengers} \\ &\sum_{j=2}^{J} e_j(d_j(5) - d_j(4)) = 805 \text{ passengers} \end{aligned} \tag{5-91}$$

It can be observed that the URS subject to Disturbance 2 does satisfy the condition. We can therefore conclude the following on the admissible disturbance set $\mathcal{D}_{\text{ad}}$ for the URS. We can observe that the MPC succeeds in stabilizing the URS for all disturbances except Disturbance 3. Therefore, we can conclude that the admissible disturbance set $\mathcal{D}_{\text{ad}}$ is not empty. We can however conclude that $\mathcal{D}_{\text{ad}}$ is bounded; although the difference between Disturbance 2 and Disturbance 3 is small, the MPC succeeds in stabilizing the URS subject to Disturbance 2 and not fails if the URS is subject to Disturbance 3. Therefore, these disturbances are located on the boundaries of $\mathcal{D}_{\text{ad}}$.

Chapter 6

Conclusions and Contributions

In this chapter, we conclude on the thesis work and look back on the research questions presented in chapter 1. We will separate the conclusions in three sections; the first section 6-1 is devoted to solvability of Max-Plus Linear Parameter Varying (MP-LPV) systems, thereafter in section 6-2 we discuss the Urban Railway System (URS) and in section 6-3 we present the conclusions and remarks on the Model Predictive Control (MPC) approach. Finally in section 6-4 a clear overview is given of the contributions this thesis provides to the field of systems and control.

6-1 On Solvability of MP-LPV Systems

After giving a thorough background overview on MP-LPV systems, we have investigated a solvability framework for MP-LPV systems in chapter 4, following the first research question and its subquestions as presented in subsection 1-2-1. Let us recall these questions:

1. *How can we define a solvability framework for general implicit MP-LPV systems?*
    (a) What is the definition of solvability for MP-LPV systems?
    (b) What levels of implicitness can we define for MP-LPV systems?
    (c) Which problems arise due to the different levels of implicitness?
    (d) What are the assumptions in the current solvability framework?
    (e) How can we relax these assumptions, resulting in a new and more general solvability framework?

We will conclude on each subquestion, to finally answer the main research question. We have introduced a new solvability framework for MP-LPV systems in section 4-1 consisting of four levels: existence of a solution, existence of a unique solution, solvability and unique solvability. Existence of a solution can be labeled as solvability, if existence is guaranteed for all future states, given the initial states. Equivalently, unique existence can be labeled as unique solvability, if the solutions are unique for all future states, given the initial states. These definitions for (unique) existence and (unique) solvability hold for MP-LPV systems, provided that the dynamics of these systems are guaranteed to hold as well. These definitions therefore provide the answer to the first subquestion 1.(a). In investigating the background on MP-LPV systems, we furthermore provided the different levels of implicitness that can occur resulting from the canonical form of the MP-LPV system. Under mild assumptions, this canonical form can always be obtained. The levels of implicitness are defined as explicit, single implicit and doubly implicit MP-LPV systems (answering subquestion 1.(b)). We have seen that explicit systems are always uniquely solvable. However, for single and doubly implicit systems, a framework needed to be developed (answering subquestion 1.(c)). In analyzing the solvability framework defined prior to this thesis work (answering subquestion 1.(d)), we have concluded the following. For single implicit systems, a condition was obtained sufficient only for existence, not for solvability of the MP-LPV system. This is caused by the of the lack of the guarantee of existence of all future solutions. Furthermore, for doubly implicit MP-LPV systems, the proof of existence was lacking, and only uniqueness was confirmed. Furthermore, this existence framework was subject to several assumptions. We have presented an updated solvability framework for (unique) existence and (unique) solvability for both single and doubly implicit systems, while relaxing all the assumptions in the existence framework for doubly implicit MP-LPV systems (answering subquestion 1.(e)). The resulting two frameworks are both based on finding three sets. The first set contains all parameter vectors that result in existence (and uniqueness) of a next solution. The second set consists out of all previous parameter sets that result in a parameter vector contained in the first set. This second set is therefore a subset of the first set. The final set is crucial for guaranteeing (unique) existence of all future solutions and therefore confirming (unique) solvability; this set contains all parameter vectors resulting in invariance in the first set. This procedure was equivalent for both single and doubly implicit systems, with the only difference being the condition that defines the first set. We can therefore conclude that we have obtained a (unique) solvability framework for all levels of implicitness resulting from the canonical form of MP-LPV systems, answering the first main research question.

6-2 On the Urban Railway System

During this thesis, several case studies were done using the URS. We have seen that this system is useful in presenting Max-Plus-Linear (MPL) systems, but also MP-LPV and Max-Min-Plus-Scaling (MMPS) systems. Recall that the latter two are equivalent to one another. The URS was presented in prior work before the start of this thesis, but was subject to several assumptions. Let us recall the second research question and its subquestions:

2. *How can we maintain solvability of the Urban Railway System, while considering a limited capacity of the trains?*
   (a) How can we include a limited capacity in the system description of the URS?
   (b) What assumptions of the current URS description can furthermore be relaxed, resulting in a more realistic URS?

(c) How can we analyze the solvability of the new URS?

In Case Study III in section 3-4 we have obtained an expression for the URS with limited capacity, answering subquestion 2.(a). We have seen that the resulting system is presents itself in an MMPS format. As we have seen however that MP-LPV systems and MMPS systems are equivalent, we can therefore rewrite the URS into MP-LPV format without any problems. The MP-LPV framework is useful as it can give clear insight in the level of implicitness and therefore in the solvability. Besides the relaxation on the unlimited capacity, we furthermore included a minimum dwell time, and specified the the actual dwell time to a more elaborate extend. The latter gives a more complete system description, including expressions of more system characteristics. The resulting URS contains expressions for the number of passengers present in the train, the number of passengers boarding and disembarking and the number of passengers left behind. We have seen however that introducing these expressions gives rise to a new set of assumptions. These assumptions are mainly based on estimating system parameters such as fixed boarding and disembarking speed. We can however relax these assumptions again; for this, one needs to obtain thorough information on a real time train route, and include approximations on these characteristics. With this, we therefore have answered the second subquestion 2.(b). Finally, we have analyzed the solvability of the URS presented in Case Study II and III. Both of the resulting MP-LPV systems seem doubly implicit on first sight due to the presence of the implicit parameter vector in combination with implicit signal relations. However, both $A_0(\cdot)$ matrices are structured such that we can consider these systems as single implicit. Due to the order in rows, the states individually do not depend on the implicit parameter. As both $A_0(\cdot)$ matrices are furthermore reducible with a strictly lower triangular structure, we can conclude that based on Proposition 4.4, the URS in both Case Study II and III are uniquely solvable. Therefore, unique solvability of the URS with limited capacity is confirmed based on the structure of the $A_0(\cdot)$ matrix. This answers subquestion 2.(c) as well as the second research question.

6-3 On Model Predictive Control for MP-LPV Systems

Let us address the final research question and its subquestions on an MPC approach for MP-LPV systems. We have approached this research question by means of introducing an MPC method for the URS with limited capacity. Let us recall the research questions:

3. *What conditions on MP-LPV systems are necessary such that we can design an MPC controller?*

   (a) Which existing MPC strategies can be extended to the application of MP-LPV systems?

   (b) How can we implement an MPC control design with the updated URS as the concerning system?

   (c) What different disturbances can be added to the system, to investigate if the MPC works properly?

   (d) What conditions are necessary to obtain a stabilizing controller?

To answer the first subquestion 3.(a), we have analyzed MPC methods for MPL and Mixed Logical Dynamical (MLD) systems. We have seen that there have been introduced MPC approaches on several other hybrid system classes, such as for MMPS systems. However, MMPS-MPC problems are nonlinear and nonconvex and therefore computationally hard to solve. As the URS is a discrete-event mixed integer system the MPL- and MLD-MPC approaches will be suitable. We have thereafter seen that by rewriting the URS into MLD format, we can obtain an optimization problem in Mixed Integer Linear Programming (MILP) format that is suitable for MPC, answering subquestion 3.(b). We can furthermore conclude that any disturbance can be added to the URS. However, the MPC will not always succeed in stabilizing the URS. We have defined closed-loop stability for the URS as a mix of a stable growth rate for the dwell time, and stable values for the states representing numbers of passengers. We have seen that the MPC is stabilizing the URS as long as the disturbance is in the admissible disturbance set $\mathcal{D}_{\text{ad}}$, answering subquestions 3.(c) and 3.(d). The condition for a stabilizing MPC is that the disturbance is designed within boundaries guaranteeing that the number of passengers arriving on the platforms is less than or equal to the number of passengers boarding, answering the third main research question. We can furthermore conclude that if a disturbance is not in the admissible disturbance set $\mathcal{D}_{\text{ad}}$, the MPC still succeeds in improving the evolution of the URS compared to the URS when no control is implemented.

6-4 Contributions

This work contributes to research in the field of max-plus algebra and systems and control through the following results:

- A clear overview of the different levels of implicitness in MP-LPV systems;
- A general solvability framework for MP-LPV systems for all levels of implicitness;
- An updated and more realistic URS model, useful for further analysis in MP-LPV systems;
- A first step towards control design for MP-LPV systems.

Chapter 7

Future Work

In this chapter, we present several suggestions for future work, directly following from this thesis work. Let us again separate the future work in three sections, each section focused on one main research question. We will thereafter include another section devoted to general future work suggestions.

7-1 On Solvability for MP-LPV Systems

In this section, we present the suggestions for future work regarding the solvability of Max-Plus Linear Parameter Varying (MP-LPV) systems.

More Relaxations on Format MP-LPV Systems

The solvability framework presented in this thesis is based on the canonical form of MP-LPV systems as presented in Lemma 3.1. For this, we have introduced Assumption 3.1. This assumption however restricts the MP-LPV framework, and it can be interesting to relax this assumption. For example, we can consider the case where the finite entries of the $A_0(\cdot)$ can attain any nonlinear function dependence on the parameter vector $p^{(0)}(k)$. Because of this relaxation, the conditions in the target sets $\mathcal{P}$ and Λ will need to be adjusted properly.

Case Studies analyzing Solvability Framework

The current case studies presented in section 4-7 and 4-8 are rather simple cases; solvability is proved based on the structure of the $A_0(\cdot)$ matrices. It is interesting to consider a single implicit MP-LPV system with irreducible $A_0(\cdot)$, for which we need to compute the sets $\mathcal{P}_{\text{inv}} \subseteq \text{Pre}(\mathcal{P}) \subseteq \mathcal{P}$, or a doubly implicit MP-LPV system for which we need to compute $\Lambda_{\text{inv}} \subseteq \text{Pre}(\Lambda) \subseteq \Lambda$. By considering such case studies, we can investigate the complexity of obtaining these sets.

Obtaining a Control Invariant Set for Solvability

In subsection 4-4-3 and 4-6-3 we have already shortly introduced the concept of a control invariant set. An interesting future work suggestion is formulating the general framework for obtaining such a control invariant set. This control invariant set can then be used to make autonomous, unsolvable MP-LPV systems solvable when subjected to control. This can furthermore be useful in computing control strategies such as Model Predictive Control (MPC) for MP-LPV systems for all levels of implicitness.

7-2 On the Urban Railway System

Let us now present the suggestion for future work regarding the Urban Railway System (URS). We have seen that the URS is a useful case study in the field of MP-LPV systems. This system is however still subject to several important assumptions that distance the URS from reality. And although several relaxations have been done in this thesis (considering a maximum capacity in the trains and considering a minimum dwell time), there are still possibilities to relax the assumptions even more. A few possibilities are:

- Consider unknown or irregular values for the parameters representing the boarding speed (b), disembarking speed (f), fraction of passengers disembarking (β), and so on;
- Consider multiple tracks and routes;
- Consider the possibility to 'turn of' the fixed minimum dwell time;
- Consider cyclicity instead of one way routes.

A few of these adjustments will result in an MP-LPV system wíth choice. Note that up until now, we have considered Discrete Event Systems (DES) with synchronization and no concurrency or choice. Considering the possibility of choice in the MP-LPV framework, will result in a new class of systems, namely switching MP-LPV systems. We will discuss this possible new class more formally in section 7-4.

7-3 On Model Predictive Control for MP-LPV Systems

Let us consider the future work suggestions on MPC implemented on MP-LPV systems. In this thesis work, we have only implemented MPC on the case study of the URS. Several future work suggestions therefore arise regarding obtaining a general MPC approach for MP-LPV systems.

General MPC Framework for MP-LPV Systems

As mentioned in the introduction of this section, we have so far only obtained an MPC for the URS case study. This raises the interest in obtaining a standard MPC framework for

MP-LPV systems, similarly as the existing framework for classes of systems such as Max-Plus-Linear (MPL) and Mixed Logical Dynamical (MLD) systems. The framework can be designed for each level of implicitness separately. The existence of a control invariant set, introduced in section 7-1 can play an important role for both single and doubly implicit MP-LPV systems. If we can construct an control invariant set for which holds that if the initial states are in this set, the MPC controller is able to steer the MP-LPV system such that it remains solvable without violating the constraints, an MPC can be designed to guarantee solvability of the implicit MP-LPV system. The general MPC framework should contain a standard formulation of the optimization problem, for which both Mixed Integer Linear Programming (MILP) and Mixed Integer Quadratic Programming (MIQP) problems provide interesting formats. For further research, the MPC approach developed for Switching Max-Plus Linear (SMPL) systems can be useful as well.

Closed-Loop Stability Framework for MP-LPV Systems

In section 5-6, the first steps are taken in analyzing the closed-loop stability, specifically for the URS case study. However, to extend this to a general closed-loop stability analysis procedure, we need to expand the idea of the admissible disturbance set as introduced in Definition 5.4. A formal definition of a stabilizing MPC for an MP-LPV must be defined, and for this we must investigate the general definition of stable DES. Here again the control invariant set can play a role; we can investigate whether we can construct a control invariant set such that if the initial values of the MP-LPV and the MPC are in this set, the MPC remains to succeed to stabilize the MP-LPV system. If such a set is nonempty, we can confirm closed-loop stability of the MP-LPV system. Another approach is to extend the Lyapunov stability analysis for Discrete Time Systems (DTS) to MP-LPV systems.

Different Objective in Optimization Problem

Besides the future work suggestions for general frameworks regarding MPC and closed-loop stability, there are also several possibilities in expanding the current MPC designed for the URS. We have now only focused on optimizing from a passenger point of view. We can extend the optimization therefore to a train based objective. An example is minimizing the energy consumption, which is specifically interesting considering the input of the running time we have considered in this thesis. Furthermore, if we expand the URS as suggested in the previous section, we can expand the MPC design accordingly.

7-4 General Future Work Recommendations

Finally, let us suggest some general research directions regarding MP-LPV systems. As the class of MP-LPV systems is only introduced recently, there are many directions to investigate. In this section we will however recommend research topics that are closely connected to this thesis work.

General Framework for Stability of Autonomous MP-LPV Systems

We have already elaborated on future work regarding a general framework for closed-loop stability of MP-LPV systems. It is however also interesting to investigate stability of autonomous MP-LPV systems. For this, we recommend to extend the research done in [41], which provides a general framework for stability of autonomous SMPL systems.

Switching MP-LPV Systems

We already introduced the concept of switching MP-LPV shortly in section 7-2. In this thesis, we have given little background on the class of SMPL systems, but this class is nevertheless interesting for future research; a possible direction is to construct a general framework for switching MP-LPV (or switching Max-Min-Plus-Scaling (MMPS)) systems. Here, the parameter vector can contain a switching mode to activate specific parts in the parameter dependencies, resulting in switching parameter varying system matrices. Again, we can consider different levels of implicitness for this new class of systems. It can be investigated if the solvability framework for implicit MP-LPV systems can be extended to switching implicit MP-LPV systems. And finally, the MPC approach designed for SMPL systems combined with the MPC approach defined in this thesis work can be extended to a general MPC framework for switching MP-LPV systems.

Controllability of MP-LPV Systems

As we are working on the first steps of control strategies for MP-LPV systems, the question arises on whether we can define a general framework for analyzing controllability of MP-LPV systems. This research direction is fairly untouched, and we therefore recommend to investigate which existing controllability tools in conventional algebra might be interesting for MP-LPV systems. The general framework for stability of MP-LPV systems might also be helpful in obtaining the first steps in obtaining a controllability framework for MP-LPV systems.

New Application as MP-LPV System

During this thesis, we used the URS as a running case study. The question arises however whether we can find a new application for which the obtained results and research can be tested. An interesting application might be an anesthesia delivery system during surgery [31]. The mathematical model obtained in this paper is in conventional Linear Parameter Varying (LPV) structure, and therefore can possibly be extended to the max-plus analogue.

Appendix A

Matlab

A-1 Simulation of Urban Railway System without Control

The following MATLAB script can be used to observe the evolution of the Urban Railway System (URS) without any control implemented. Different disturbances can be activated, and one can change the initial values to investigate its effect. This script is used in subsection 5-5-1. The script that plots the results has been omitted, as that part is irrelevant for the content of the script.

```
% ------------------------------------ %
% ---------- MLD URS ---------- %
% ------------------------------------ %
% -------- NO CONTROL -------- %
% ------------------------------------ %

% Add yalmip and gurobi to path

yalmip('clear')
clearvars
clear all
close all
clc
format shortG
%% System Parameters

rhomax = 150;            %maximum capacity of passengers per train
tau_dwellmin = 30;       %minimum dwell time [sec]
tau_h = 30;              %headway time [sec]

gamma1 = 1;

m1 = -3*180;
m2 = -3*rhomax*4;
m3 = -3*rhomax*4;
m4 = -3*rhomax;
M1 = 3*180;
M2 = 3*rhomax*4;
M3 = 3*rhomax*4;
M4 = 3*rhomax;

epsilon = 1e-15;         %machine error

%% Controller Variables
dim.J = 4;                 % Station prediction horizon
dim.K = 4;                 % Train prediction horizon

%% Define Decision Variables
x1 = []; x2 = []; x3 = []; x4 = []; x5 = []; x6 = [];
delta1 = []; delta2 = []; delta3 = []; delta4 = [];
h1 = []; h2 = []; h3 = []; h4 = []; h5 = []; h6 = [];
z1 = []; z2 = []; z3 = []; z4 = [];
```

```
f1 = []; f2 = []; f3 = []; f4 = [];
g1 = []; g2 = [];
taur = [];
alpha1 = []; alpha2 = []; alpha3 = []; alpha4 = []; alpha5 = [];
gamma2 = []; gamma0 = [];
e = []; beta = []; b = []; f = [];


for j = 1:dim.J

x1 = [x1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x1(j,k) = a_j(k)       (arrival time)
x2 = [x2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x2(j,k) = d_j(k)       (departure time)
x3 = [x3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x3(j,k) = q_j(k)       (#passengers boarding train k at station j)
x4 = [x4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x4(j,k) = rho_j(k)     (#passengers in train k at departure from station j)
x5 = [x5; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x5(j,k) = sigma_j(k)   (#passengers on platform j at departure k)
x6 = [x6; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x6(j,k) = p^wait_j(k)  (#passengers waiting on station j at moment of arrival train k)


delta1 = [delta1; binvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];     %for a_j(k)
delta2 = [delta2; binvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];     %for d_j(k)
delta3 = [delta3; binvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];     %for d_j(k)
delta4 = [delta4; binvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];     %for q_j(k)

h1 = [h1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h2 = [h2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h3 = [h3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h4 = [h4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h5 = [h5; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h6 = [h6; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

z1 = [z1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
z2 = [z2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
z3 = [z3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
z4 = [z4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

f1 = [f1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
f2 = [f2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
f3 = [f3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
f4 = [f4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

g1 = [g1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
g2 = [g2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

taur = [taur; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

alpha1 = [alpha1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
alpha2 = [alpha2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
alpha3 = [alpha3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
alpha4 = [alpha4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
alpha5 = [alpha5; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

e = [e; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
beta = [beta; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
b = [b; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

f = [f; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

gamma0 = [gamma0; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
gamma2 = [gamma2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

end

%% Controller Design

constraints = [];
objective = 0;

for j = 2:dim.J
for k = 2:dim.K

    % Constraints arrival time x1
    % x1 = max(h1,h2)
    constraints = [constraints, h1{j,k} == x2{j-1,k} + taur{j,k}];
    constraints = [constraints, h2{j,k} == x2{j,k-1} + tau_h];
    constraints = [constraints, f1{j,k} == h1{j,k} - h2{j,k}];
    constraints = [constraints, x1{j,k} == h1{j,k} - z1{j,k}];

    constraints = [constraints, epsilon + (m1 - epsilon)*delta1{j,k} <= f1{j,k}];
    constraints = [constraints, f1{j,k} <= M1*(1-delta1{j,k})];

    constraints = [constraints, m1*delta1{j,k} <= z1{j,k}];
    constraints = [constraints, z1{j,k} <= M1*delta1{j,k}];
    constraints = [constraints, z1{j,k} <= f1{j,k} - m1*(1-delta1{j,k})];
    constraints = [constraints, f1{j,k} - M1*(1-delta1{j,k}) <= z1{j,k}];

    % Constraints departure time x2
```

```
    % x2 = max(h3,min(g1,g2)), h4 = min(g1,g2)
    constraints = [constraints, g1{j,k} == alpha1{j,k}*x1{j,k} + alpha2{j,k}*x4{j-1,k} + alpha3{j,k
        }*x5{j,k-1} + alpha4{j,k}*x2{j,k-1}];
    constraints = [constraints, g2{j,k} == gamma0{j,k} + gamma1*x1{j,k}  + gamma2{j,k}*x4{j-1,k}];
    constraints = [constraints, f2{j,k} == g2{j,k} - g1{j,k}];
    constraints = [constraints, h4{j,k} == g1{j,k} + z2{j,k}];

    constraints = [constraints, epsilon + (m2 - epsilon)*delta2{j,k} <= f2{j,k}];
    constraints = [constraints, f2{j,k} <= M2*(1-delta2{j,k})];

    constraints = [constraints, m2*delta2{j,k} <= z2{j,k}];
    constraints = [constraints, z2{j,k} <= M2*delta2{j,k}];
    constraints = [constraints, z2{j,k} <= f2{j,k} - m2*(1-delta2{j,k})];
    constraints = [constraints, f2{j,k} - M2*(1-delta2{j,k}) <= z2{j,k}];

    constraints = [constraints, h3{j,k} == tau_dwellmin + x1{j,k}];
    constraints = [constraints, f3{j,k} == h4{j,k} - h3{j,k}];
    constraints = [constraints, x2{j,k} == g1{j,k} + z2{j,k} - z3{j,k}];

    constraints = [constraints, epsilon + (m3 - epsilon)*delta3{j,k} <= f3{j,k}];
    constraints = [constraints, f3{j,k} <= M3*(1-delta3{j,k})];

    constraints = [constraints, m3*delta3{j,k} <= z3{j,k}];
    constraints = [constraints, z3{j,k} <= M3*delta3{j,k}];
    constraints = [constraints, z3{j,k} <= f3{j,k} - m3*(1-delta3{j,k})];
    constraints = [constraints, f3{j,k} - M3*(1-delta3{j,k}) <= z3{j,k}];

    % Constraints #passengers boarding train k at station j x3
    % x3 = min(h5,h6)

    constraints = [constraints, h5{j,k} == e{j,k}*(x2{j,k} - x2{j,k-1}) + x5{j,k-1}];
    constraints = [constraints, h6{j,k} == rhomax - (1-beta{j,k})*x4{j-1,k}];
    constraints = [constraints, f4{j,k} == h6{j,k} - h5{j,k}];
    constraints = [constraints, x3{j,k} == h5{j,k} + z4{j,k}];

    constraints = [constraints, epsilon + (m4 - epsilon)*delta4{j,k} <= f4{j,k}];
    constraints = [constraints, f4{j,k} <= M4*(1-delta4{j,k})];

    constraints = [constraints, m4*delta4{j,k} <= z4{j,k}];
    constraints = [constraints, z4{j,k} <= M4*delta4{j,k}];
    constraints = [constraints, z4{j,k} <= f4{j,k} - m4*(1-delta4{j,k})];
    constraints = [constraints, f4{j,k} - M4*(1-delta4{j,k}) <= z4{j,k}];

    % Constraint #passengers in train x4
    constraints = [constraints, x4{j,k} == (1-beta{j,k})*x4{j-1,k} + x3{j,k}];

    % Constraint #passengers left behind x5
    constraints = [constraints, x5{j,k} == e{j,k}*(x2{j,k} - x2{j,k-1}) + x5{j,k-1} - x3{j,k}];

    % Constraint #passengers waiting on j at moment of arrival k x6
    constraints = [constraints, x6{j,k} == e{j,k}*(x1{j,k} - x2{j,k-1}) + x5{j,k-1}];

    % Remaining constraints
    constraints = [constraints, 0 <= x1{j,k}];
    constraints = [constraints, 0 <= x2{j,k}];
    constraints = [constraints, 0 <= x3{j,k}];
    constraints = [constraints, 0 <= x4{j,k}];
    constraints = [constraints, x4{j,k} <= rhomax];
    constraints = [constraints, 0 <= x5{j,k}];
    constraints = [constraints, 0 <= x6{j,k}];

end
end


options = sdpsettings('solver','gurobi');
parameters_in = {x2{2,1},x2{1,2},x4{1,2},x5{2,1},[taur{:,:}],[alpha1{:,:}],[alpha2{:,:}],[alpha3
      {:,:}],[alpha4{:,:}],[alpha5{:,:}],[e{:,:}],[gamma0{:,:}],[gamma2{:,:}],[beta{:,:}],[b{:,:}],[f
      {:,:}]};
parameters_out = {[z1{:,:}], [z2{:,:}], [z3{:,:}], [z4{:,:}], [delta1{:,:}], [delta2{:,:}], [delta3
      {:,:}], [delta4{:,:}], [f1{:,:}], [f2{:,:}], [f3{:,:}], [f4{:,:}], [x1{:,:}], [x2{:,:}], [x3
      {:,:}], [x4{:,:}], [x5{:,:}], [x6{:,:}]};

controller = optimizer(constraints,[],options,parameters_in,parameters_out);


%% Simulation
% Disturbances 1-6 can be activated in the loop. For disturbance 6, minimum
% dwell time must be set on 0.

Khorizon = 3*dim.K+1;
Jhorizon = 3*dim.J+1;

x1hold = []; x2hold = []; x3hold = []; x4hold = []; x5hold = []; x6hold = [];
x1final = []; x2final = []; x3final = []; x4final = []; x5final = []; x6final = [];
z1hold = []; delta1hold = []; f1hold = []; delta4hold = [];

for j = 2:Jhorizon
    for k = 2:Khorizon

```

```
        % Compute initial conditions for parameters_in
        if j == 2 && k == 2
            x2inputk = 240;
            x2inputj = 120;
            x4input = 120;
            x5input = 0;
            einput = 0.5*ones(1,dim.J*dim.K);
            betainput = 0.5*ones(1,dim.J*dim.K);
            taurinput = 180*ones(1,dim.J*dim.K);
            binput = 2*ones(1,dim.J*dim.K);
            finput = 2*ones(1,dim.J*dim.K);
        elseif j == 2 && k > 2
            x2inputk = x2hold(k-2);
            x2inputj = (k-1)*120;
            x4input = 120;
            x5input = 0;
            einput = 0.5*ones(1,dim.J*dim.K);
            betainput = 0.5*ones(1,dim.J*dim.K);
            taurinput = 180*ones(1,dim.J*dim.K);
            binput = 2*ones(1,dim.J*dim.K);
            finput = 2*ones(1,dim.J*dim.K);
        elseif k == 2 && j > 2
            x2inputk = (j-1)*240;
            x2inputj = x2final(j-2,1);
            x4input = 120;
            x5input = 0;
            einput = 0.5*ones(1,dim.J*dim.K);
            betainput = 0.5*ones(1,dim.J*dim.K);
            taurinput = 180*ones(1,dim.J*dim.K);
            binput = 2*ones(1,dim.J*dim.K);
            finput = 2*ones(1,dim.J*dim.K);
% % Disturbance 1
%          elseif k == 6 && j == 5
%              x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%              x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%              x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%              x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%              einput = 0.5*ones(1,dim.J*dim.K);
%              betainput = 0.5*ones(1,dim.J*dim.K);
%              taurinput = 180*ones(1,dim.J*dim.K);
%              binput = 0.6*ones(1,dim.J*dim.K);
%              finput = 2*ones(1,dim.J*dim.K);
% % Disturbance 2
%          elseif k == 6
%              x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%              x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%              x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%              x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%              einput = 0.5*ones(1,dim.J*dim.K);
%              betainput = 0.5*ones(1,dim.J*dim.K);
%              taurinput = 180*ones(1,dim.J*dim.K);
%              binput = 1.6*ones(1,dim.J*dim.K);
%              finput = 2*ones(1,dim.J*dim.K);
% % Disturbance 3
%          elseif k == 6
%              x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%              x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%              x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%              x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%              einput = 0.5*ones(1,dim.J*dim.K);
%              betainput = 0.5*ones(1,dim.J*dim.K);
%              taurinput = 180*ones(1,dim.J*dim.K);
%              binput = 1.5*ones(1,dim.J*dim.K);
%              finput = 2*ones(1,dim.J*dim.K);
% % Disturbance 4
%          elseif k == 6 && j == 5
%              x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%              x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%              x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%              x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%              einput = 0.3*ones(1,dim.J*dim.K);
%              betainput = 0.5*ones(1,dim.J*dim.K);
%              taurinput = 180*ones(1,dim.J*dim.K);
%              binput = 2*ones(1,dim.J*dim.K);
%              finput = 2*ones(1,dim.J*dim.K);
           % Disturbance 5
%          elseif k == 6
%              x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%              x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%              x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%              x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%              einput = 0.3*ones(1,dim.J*dim.K);
%              betainput = 0.5*ones(1,dim.J*dim.K);
%              taurinput = 180*ones(1,dim.J*dim.K);
%              binput = 2*ones(1,dim.J*dim.K);
%              finput = 2*ones(1,dim.J*dim.K);
           % Disturbance 6
%          elseif k == 6 && j == 5
%              x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%              x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
```

```
%               x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%               x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%               einput = 0*ones(1,dim.J*dim.K);
%               betainput = 0*ones(1,dim.J*dim.K);
%               taurinput = 180*ones(1,dim.J*dim.K);
%               binput = 2*ones(1,dim.J*dim.K);
%               finput = 2*ones(1,dim.J*dim.K);

        else
            x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
            x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
            x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
            x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
            einput = 0.5*ones(1,dim.J*dim.K);
            betainput = 0.5*ones(1,dim.J*dim.K);
            taurinput = 180*ones(1,dim.J*dim.K);
            binput = 2*ones(1,dim.J*dim.K);
            finput = 2*ones(1,dim.J*dim.K);
        end

        alpha1input = (binput(1,1)/(binput(1,1)-einput(1,1)))*ones(1,dim.J*dim.K);
        alpha2input = (binput(1,1)/(binput(1,1)-einput(1,1))*(betainput(1,1)/finput(1,1)))*ones(1,
            dim.J*dim.K);
        alpha3input = (1/(binput(1,1)-einput(1,1)))*ones(1,dim.J*dim.K);
        alpha4input = (-einput(1,1)/(binput(1,1)-einput(1,1)))*ones(1,dim.J*dim.K);
        alpha5input = (binput(1,1)/(binput(1,1)-einput(1,1)))*ones(1,dim.J*dim.K);
        gamma0input = (rhomax/binput(1,1))*ones(1,dim.J*dim.K);
        gamma2input = ((betainput(1,1)/finput(1,1)) - (1-betainput(1,1))/binput(1,1))*ones(1,dim.J*
            dim.K);

        [solution,diagnostics] = controller(x2inputk, x2inputj, x4input, x5input, taurinput,
            alpha1input,alpha2input,alpha3input,alpha4input,alpha5input,einput,gamma0input,
            gamma2input,betainput,binput,finput);


        Z1 = reshape(solution{1},dim.K,dim.J)';
        Z2 = reshape(solution{2},dim.K,dim.J)';
        Z3 = reshape(solution{3},dim.K,dim.J)';
        Z4 = reshape(solution{4},dim.K,dim.J)';
        DELTA1 = reshape(solution{5},dim.K,dim.J)';
        DELTA2 = reshape(solution{6},dim.K,dim.J)';
        DELTA3 = reshape(solution{7},dim.K,dim.J)';
        DELTA4 = reshape(solution{8},dim.K,dim.J)';
        F1 = reshape(solution{9},dim.K,dim.J)';
        F2 = reshape(solution{10},dim.K,dim.J)';
        F3 = reshape(solution{11},dim.K,dim.J)';
        F4 = reshape(solution{12},dim.K,dim.J)';

        % Compute initial conditions
        doldj = x2inputj;
        doldk = x2inputk;
        rhooldj = x4input;
        sigmaoldk = x5input;

        % Compute arrival time
        h1 = doldj + taurinput(1,1);
        h2 = doldk + tau_h;
        f1 = h1 - h2;
        x1 = h1 - Z1(2,2);

        x1hold = [x1hold x1];

        % Compute departure time
        g1 = alpha1input(1,1)*x1 + alpha2input(1,1)*x4input + alpha3input(1,1)*x5input + alpha4input
            (1,1)*doldk;
        g2 = gamma0input(1,1) + gamma1*x1  + gamma2input(1,1)*x4input;
        f2 = g2 - g1;
        h4 = g1 + Z2(2,2);

        h3 = tau_dwellmin + x1;
        f3 = h4 - h3;
        x2 = g1 + Z2(2,2) - Z3(2,2);

        x2hold = [x2hold x2];

        % Compute passengers boarding
        h5 = einput(1,1)*(x2 - doldk) + x5input;
        h6 = rhomax - (1-betainput(1,1))*x4input;
        f4 = h6 - h5;
        x3 = h5 + Z4(2,2);

        x3hold = [x3hold x3];

        % Compute passengers in train
        x4 = (1 - betainput(1,1))*rhooldj + x3;

        x4hold = [x4hold x4];

        % Compute passengers left behind
        x5 = einput(1,1)*(x2 - doldk) + sigmaoldk - x3;
```

```
        x5hold = [x5hold x5];

        % Compute passengers waiting
        x6 = einput(1,1)*(x1 - doldk) + sigmaoldk;
        x6hold = [x6hold x6];

    end

    x1final = [x1final;x1hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x2final = [x2final;x2hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x3final = [x3final;x3hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x4final = [x4final;x4hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x5final = [x5final;x5hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x6final = [x6final;x6hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];


end

x1final = [zeros(1,Khorizon-1);x1final];                        %arrival
x1final = [zeros(Jhorizon,1) x1final];
x2final = [zeros(1,Khorizon-1);x2final];                        %departure
x2final = [zeros(Jhorizon,1) x2final];
x2final(1,1:Khorizon)=((1:Khorizon)-1)*120;
x2final(1:Jhorizon,1)=((1:Jhorizon)'-1)*240;
x3final = [zeros(1,Khorizon-1);x3final];                        %q
x3final = [zeros(Jhorizon,1) x3final];
x4final = [120*ones(1,Khorizon-1);x4final];                     %rho
x4final = [120*ones(Jhorizon,1) x4final];
x5final = [zeros(1,Khorizon-1);x5final];                        %sigma
x5final = [zeros(Jhorizon,1) x5final];
x6final = [zeros(1,Khorizon-1);x6final];                        %p wait
x6final = [zeros(Jhorizon,1) x6final];
```

A-2 MPC Design and Simulation Urban Railway System

In the following MATLAB script, the Model Predictive Control (MPC) design for the URS can be found. This script is used to simulate the results in subsection 5-5-2 .The main difference with the previous script is the presence of the input values $u_{1,j}(k)$ and $u_{2,j}(k)$ and the objective function as defined in Equation 5-79. It is again possible to activate different disturbances or to change the initial values.

```
% ———————————————————— %
% ———— MLD URS ———— %
% ———————————————————— %
% ——— WITH MPC ———— %
% ———————————————————— %

% Add yalmip and gurobi to path

yalmip('clear')
clearvars
clear all
close all
clc
format shortG

%% System Parameters

rhomax = 150;               %maximum capacity of passengers per train
tau_dwellmin = 30;          %minimum dwell time [sec]
tau_h = 30;                 %headway time [sec]


gamma1 = 1;

m1 = -3*180;
m2 = -3*rhomax*4;
m3 = -3*rhomax*4;
m4 = -3*rhomax;
M1 = 3*180;
M2 = 3*rhomax*4;
M3 = 3*rhomax*4;
M4 = 3*rhomax;

epsilon = 1e-15;            %machine error

%% Controller Variables
dim.J = 4;                   % Station prediction horizon
dim.K = 4;                   % Train prediction horizon
```

```
lambda2 = 0.1;                % 0 < lambda \leq 1
lambda3 = 0.1;                % 0 < lambda \leq 1

%% Define Decision Variables
x1 = []; x2 = []; x3 = []; x4 = []; x5 = []; x6 = [];
delta1 = []; delta2 = []; delta3 = []; delta4 = [];
h1 = []; h2 = []; h3 = []; h4 = []; h5 = []; h6 = [];
z1 = []; z2 = []; z3 = []; z4 = [];
f1 = []; f2 = []; f3 = []; f4 = [];
g1 = []; g2 = [];
taur = [];
alpha1 = []; alpha2 = []; alpha3 = []; alpha4 = []; alpha5 = [];
gamma2 = []; gamma0 = [];
e = []; beta = []; b = []; f = [];

u1 = [];
u2 = [];

for j = 1:dim.J

x1 = [x1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x1(j,k) = a_j(k)       (arrival time)
x2 = [x2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x2(j,k) = d_j(k)       (departure time)
x3 = [x3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x3(j,k) = q_j(k)       (#passengers
     boarding train k at station j)
x4 = [x4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x4(j,k) = rho_j(k)     (#passengers in
     train k at departure from station j)
x5 = [x5; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x5(j,k) = sigma_j(k)   (#passengers on
     platform j at departure k)
x6 = [x6; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];  % x6(j,k) = p^wait_j(k)  (#passengers
     waiting on station j at moment of arrival train k)

u1 = [u1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
u2 = [u2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

delta1 = [delta1; binvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];     %for a_j(k)
delta2 = [delta2; binvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];     %for d_j(k)
delta3 = [delta3; binvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];     %for d_j(k)
delta4 = [delta4; binvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];     %for q_j(k)

h1 = [h1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h2 = [h2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h3 = [h3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h4 = [h4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h5 = [h5; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
h6 = [h6; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

z1 = [z1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
z2 = [z2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
z3 = [z3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
z4 = [z4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

f1 = [f1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
f2 = [f2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
f3 = [f3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
f4 = [f4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

g1 = [g1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
g2 = [g2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

taur = [taur; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

alpha1 = [alpha1; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
alpha2 = [alpha2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
alpha3 = [alpha3; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
alpha4 = [alpha4; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
alpha5 = [alpha5; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

e = [e; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
beta = [beta; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
b = [b; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

f = [f; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];


gamma0 = [gamma0; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];
gamma2 = [gamma2; sdpvar(repmat(1,1,dim.K),repmat(1,1,dim.K))];

end

%% Controller Design

constraints = [];
objective = 0;

for j = 2:dim.J
for k = 2:dim.K

    % Constraints arrival time x1
    % x1 = max(h1,h2)
    constraints = [constraints, h1{j,k} == x2{j-1,k} + taur{j,k} + u1{j,k}];
```

```
        constraints = [constraints, h2{j,k} == x2{j,k-1} + tau_h + u2{j,k}];
        constraints = [constraints, f1{j,k} == h1{j,k} - h2{j,k}];
        constraints = [constraints, x1{j,k} == h1{j,k} - z1{j,k}];

        constraints = [constraints, epsilon + (m1 - epsilon)*delta1{j,k} <= f1{j,k}];
        constraints = [constraints, f1{j,k} <= M1*(1-delta1{j,k})];

        constraints = [constraints, m1*delta1{j,k} <= z1{j,k}];
        constraints = [constraints, z1{j,k} <= M1*delta1{j,k}];
        constraints = [constraints, z1{j,k} <= f1{j,k} - m1*(1-delta1{j,k})];
        constraints = [constraints, f1{j,k} - M1*(1-delta1{j,k}) <= z1{j,k}];

        % Constraints departure time x2
        % x2 = max(h3,min(g1,g2)), h4 = min(g1,g2)
        constraints = [constraints, g1{j,k} == alpha1{j,k}*x1{j,k} + alpha2{j,k}*x4{j-1,k} + alpha3{j,k
            }*x5{j,k-1} + alpha4{j,k}*x2{j,k-1}];
        constraints = [constraints, g2{j,k} == gamma0{j,k} + gamma1*x1{j,k}  + gamma2{j,k}*x4{j-1,k}];
        constraints = [constraints, f2{j,k} == g2{j,k} - g1{j,k}];
        constraints = [constraints, h4{j,k} == g1{j,k} + z2{j,k}];

        constraints = [constraints, epsilon + (m2 - epsilon)*delta2{j,k} <= f2{j,k}];
        constraints = [constraints, f2{j,k} <= M2*(1-delta2{j,k})];

        constraints = [constraints, m2*delta2{j,k} <= z2{j,k}];
        constraints = [constraints, z2{j,k} <= M2*delta2{j,k}];
        constraints = [constraints, z2{j,k} <= f2{j,k} - m2*(1-delta2{j,k})];
        constraints = [constraints, f2{j,k} - M2*(1-delta2{j,k}) <= z2{j,k}];

        constraints = [constraints, h3{j,k} == tau_dwellmin + x1{j,k}];
        constraints = [constraints, f3{j,k} == h4{j,k} - h3{j,k}];
        constraints = [constraints, x2{j,k} == g1{j,k} + z2{j,k} - z3{j,k}];

        constraints = [constraints, epsilon + (m3 - epsilon)*delta3{j,k} <= f3{j,k}];
        constraints = [constraints, f3{j,k} <= M3*(1-delta3{j,k})];

        constraints = [constraints, m3*delta3{j,k} <= z3{j,k}];
        constraints = [constraints, z3{j,k} <= M3*delta3{j,k}];
        constraints = [constraints, z3{j,k} <= f3{j,k} - m3*(1-delta3{j,k})];
        constraints = [constraints, f3{j,k} - M3*(1-delta3{j,k}) <= z3{j,k}];

        % Constraints #passengers boarding train k at station j x3
        % x3 = min(h5,h6)

        constraints = [constraints, h5{j,k} == e{j,k}*(x2{j,k} - x2{j,k-1}) + x5{j,k-1}];
        constraints = [constraints, h6{j,k} == rhomax - (1-beta{j,k})*x4{j-1,k}];
        constraints = [constraints, f4{j,k} == h6{j,k} - h5{j,k}];
        constraints = [constraints, x3{j,k} == h5{j,k} + z4{j,k}];

        constraints = [constraints, epsilon + (m4 - epsilon)*delta4{j,k} <= f4{j,k}];
        constraints = [constraints, f4{j,k} <= M4*(1-delta4{j,k})];

        constraints = [constraints, m4*delta4{j,k} <= z4{j,k}];
        constraints = [constraints, z4{j,k} <= M4*delta4{j,k}];
        constraints = [constraints, z4{j,k} <= f4{j,k} - m4*(1-delta4{j,k})];
        constraints = [constraints, f4{j,k} - M4*(1-delta4{j,k}) <= z4{j,k}];

        % Constraint #passengers in train x4
        constraints = [constraints, x4{j,k} == (1-beta{j,k})*x4{j-1,k} + x3{j,k}];

        % Constraint #passengers left behind x5
        constraints = [constraints, x5{j,k} == e{j,k}*(x2{j,k} - x2{j,k-1}) + x5{j,k-1} - x3{j,k}];

        % Constraint #passengers waiting on j at moment of arrival k x6
        constraints = [constraints, x6{j,k} == e{j,k}*(x1{j,k} - x2{j,k-1}) + x5{j,k-1}];

        % Remaining constraints
        constraints = [constraints, 0 <= x1{j,k}];
        constraints = [constraints, 0 <= x2{j,k}];
        constraints = [constraints, 0 <= x3{j,k}];
        constraints = [constraints, 0 <= x4{j,k}];
        constraints = [constraints, x4{j,k} <= rhomax];
        constraints = [constraints, 0 <= x5{j,k}];
        constraints = [constraints, 0 <= x6{j,k}];
        constraints = [constraints, tau_dwellmin <= x2{j,k}-x1{j,k}];
        constraints = [constraints, 160 <= taur{j,k} + u1{j,k}];
        constraints = [constraints, taur{j,k} + u1{j,k} <= 250];
        constraints = [constraints, 20 <= tau_h + u2{j,k}];
        constraints = [constraints, tau_h + u2{j,k} <= 80];

       objective = objective + norm((x6{j,k}-30),1) + lambda2*norm(u1{j,k},1)+ lambda3*norm(u2{j,k},1);

end
end


options = sdpsettings('solver','gurobi');
parameters_in = {x2{2,1},x2{1,2},x4{1,2},x5{2,1},[taur{:,:}],[alpha1{:,:}],[alpha2{:,:}],[alpha3
     {:,:}],[alpha4{:,:}],[alpha5{:,:}],[e{:,:}],[gamma0{:,:}],[gamma2{:,:}],[beta{:,:}],[b{:,:}],[f
     {:,:}]};
```

```
parameters_out = {[u1{:}],[u2{:}], [z1{:,:}], [z2{:,:}], [z3{:,:}], [z4{:,:}], [delta1{:,:}], [
     delta2{:,:}], [delta3{:,:}], [delta4{:,:}], [f1{:,:}], [f2{:,:}], [f3{:,:}], [f4{:,:}], [x1
     {:,:}], [x2{:,:}], [x3{:,:}], [x4{:,:}], [x5{:,:}], [x6{:,:}]};

controller = optimizer(constraints, objective,options,parameters_in,parameters_out);


%% Simulation
% Disturbances 1-6 can be activated in the loop. For disturbance 6, minimum
% dwell time must be set on 0.

Khorizon = 3*dim.K+1;
Jhorizon = 3*dim.J+1;

implementedU1 = [];
implementedU2 = [];

x1hold = []; x2hold = []; x3hold = []; x4hold = []; x5hold = []; x6hold = [];
x1final = []; x2final = []; x3final = []; x4final = []; x5final = []; x6final = [];
z1hold = []; delta1hold = []; f1hold = []; delta4hold = [];
diag = [];

for j = 2:Jhorizon
    for k = 2:Khorizon

        % Compute initial conditions for parameters_in
        if j == 2 && k == 2
            x2inputk = 240;
            x2inputj = 120;
            x4input = 120;
            x5input = 0;
            einput = 0.5*ones(1,dim.J*dim.K);
            betainput = 0.5*ones(1,dim.J*dim.K);
            taurinput = 180*ones(1,dim.J*dim.K);
            binput = 2*ones(1,dim.J*dim.K);
            finput = 2*ones(1,dim.J*dim.K);

        elseif j == 2 && k > 2
            x2inputk = x2hold(k-2);
            x2inputj = (k-1)*120;
            x4input = 120;
            x5input = 0;
            einput = 0.5*ones(1,dim.J*dim.K);
            betainput = 0.5*ones(1,dim.J*dim.K);
            taurinput = 180*ones(1,dim.J*dim.K);
            binput = 2*ones(1,dim.J*dim.K);
            finput = 2*ones(1,dim.J*dim.K);
        elseif k == 2 && j > 2
            x2inputk = (j-1)*240;
            x2inputj = x2final(j-2,1);
            x4input = 120;
            x5input = 0;
            einput = 0.5*ones(1,dim.J*dim.K);
            betainput = 0.5*ones(1,dim.J*dim.K);
            taurinput = 180*ones(1,dim.J*dim.K);
            binput = 2*ones(1,dim.J*dim.K);
            finput = 2*ones(1,dim.J*dim.K);
% % Disturbance 1
%         elseif k == 6 && j == 5
%             x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%             x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%             x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%             x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%             einput = 0.5*ones(1,dim.J*dim.K);
%             betainput = 0.5*ones(1,dim.J*dim.K);
%             taurinput = 180*ones(1,dim.J*dim.K);
%             binput = 0.6*ones(1,dim.J*dim.K);
%             finput = 2*ones(1,dim.J*dim.K);
% % Disturbance 2
%         elseif k == 6
%             x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%             x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%             x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%             x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%             einput = 0.5*ones(1,dim.J*dim.K);
%             betainput = 0.5*ones(1,dim.J*dim.K);
%             taurinput = 180*ones(1,dim.J*dim.K);
%             binput = 1.6*ones(1,dim.J*dim.K);
%             finput = 2*ones(1,dim.J*dim.K);
% % Disturbance 3
%         elseif k == 6
%             x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%             x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%             x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%             x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%             einput = 0.5*ones(1,dim.J*dim.K);
%             betainput = 0.5*ones(1,dim.J*dim.K);
%             taurinput = 180*ones(1,dim.J*dim.K);
%             binput = 1.5*ones(1,dim.J*dim.K);
%             finput = 2*ones(1,dim.J*dim.K);
```

```
% % Disturbance 4
%         elseif k == 6 && j == 5
%             x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%             x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%             x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%             x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%             einput = 0.3*ones(1,dim.J*dim.K);
%             betainput = 0.5*ones(1,dim.J*dim.K);
%             taurinput = 180*ones(1,dim.J*dim.K);
%             binput = 2*ones(1,dim.J*dim.K);
%             finput = 2*ones(1,dim.J*dim.K);
% %           Disturbance 6
%         elseif k == 6 && j == 5
%             x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
%             x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
%             x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
%             x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
%             einput = 0*ones(1,dim.J*dim.K);
%             betainput = 0*ones(1,dim.J*dim.K);
%             taurinput = 180*ones(1,dim.J*dim.K);
%             binput = 2*ones(1,dim.J*dim.K);
%             finput = 2*ones(1,dim.J*dim.K);
        else
            x2inputk = x2hold(1,(j-2)*(Khorizon-1)+k-2);
            x2inputj = x2hold(1,(j-3)*(Khorizon-1)+k-1);
            x4input = x4hold(1,(j-3)*(Khorizon-1)+k-1);
            x5input = x5hold(1,(j-2)*(Khorizon-1)+k-2);
            einput = 0.5*ones(1,dim.J*dim.K);
            betainput = 0.5*ones(1,dim.J*dim.K);
            taurinput = 180*ones(1,dim.J*dim.K);
            binput = 2*ones(1,dim.J*dim.K);
            finput = 2*ones(1,dim.J*dim.K);
        end
        alpha1input = (binput(1,1)/(binput(1,1)-einput(1,1)))*ones(1,dim.J*dim.K);
        alpha2input = (binput(1,1)/(binput(1,1)-einput(1,1))*(betainput(1,1)/finput(1,1)))*ones(1,
            dim.J*dim.K);
        alpha3input = (1/(binput(1,1)-einput(1,1)))*ones(1,dim.J*dim.K);
        alpha4input = (-einput(1,1)/(binput(1,1)-einput(1,1)))*ones(1,dim.J*dim.K);
        alpha5input = (binput(1,1)/(binput(1,1)-einput(1,1)))*ones(1,dim.J*dim.K);
        gamma0input = (rhomax/binput(1,1))*ones(1,dim.J*dim.K);
        gamma2input = ((betainput(1,1)/finput(1,1)) - (1-betainput(1,1))/binput(1,1))*ones(1,dim.J*
            dim.K);

        [solution,diagnostics] = controller(x2inputk, x2inputj, x4input, x5input, taurinput,
            alpha1input,alpha2input,alpha3input,alpha4input,alpha5input,einput,gamma0input,
            gamma2input,betainput,binput,finput);

        diag = [diag diagnostics];
        U1 = reshape(solution{1},dim.K,dim.J)';
        U2 = reshape(solution{2},dim.K,dim.J)';
        Z1 = reshape(solution{3},dim.K,dim.J)';
        Z2 = reshape(solution{4},dim.K,dim.J)';
        Z3 = reshape(solution{5},dim.K,dim.J)';
        Z4 = reshape(solution{6},dim.K,dim.J)';
        DELTA1 = reshape(solution{7},dim.K,dim.J)';
        DELTA2 = reshape(solution{8},dim.K,dim.J)';
        DELTA3 = reshape(solution{9},dim.K,dim.J)';
        DELTA4 = reshape(solution{10},dim.K,dim.J)';
        F1 = reshape(solution{11},dim.K,dim.J)';
        F2 = reshape(solution{12},dim.K,dim.J)';
        F3 = reshape(solution{13},dim.K,dim.J)';
        F4 = reshape(solution{14},dim.K,dim.J)';

        % Compute initial conditions
        doldj = x2inputj;
        doldk = x2inputk;
        rhooldj = x4input;
        sigmaoldk = x5input;

        % Compute arrival time
        h1 = doldj + taurinput(1,1)+U1(2,2);
        h2 = doldk + tau_h+U2(2,2);
        f1 = h1 - h2;
        x1 = h1 - Z1(2,2);

        x1hold = [x1hold x1];

        % Compute departure time
        g1 = alpha1input(1,1)*x1 + alpha2input(1,1)*x4input + alpha3input(1,1)*x5input + alpha4input
            (1,1)*doldk;
        g2 = gamma0input(1,1) + gamma1*x1  + gamma2input(1,1)*x4input;
        f2 = g2 - g1;
        h4 = g1 + Z2(2,2);

        h3 = tau_dwellmin + x1;
        f3 = h4 - h3;
        x2 = g1 + Z2(2,2) - Z3(2,2);

        x2hold = [x2hold x2];

```

```
        % Compute passengers boarding
        h5 = einput(1,1)*(x2 - doldk) + x5input;
        h6 = rhomax - (1-betainput(1,1))*x4input;
        f4 = h6 - h5;
        x3 = h5 + Z4(2,2);

        x3hold = [x3hold x3];

        % Compute passengers in train
        x4 = (1 - betainput(1,1))*rhooldj + x3;

        x4hold = [x4hold x4];

        % Compute passengers left behind
        x5 = einput(1,1)*(x2 - doldk) + sigmaoldk - x3;
        x5hold = [x5hold x5];

        % Compute passengers waiting
        x6 = einput(1,1)*(x1 - doldk) + sigmaoldk;
        x6hold = [x6hold x6];

        implementedU1 = [implementedU1;U1(2,2)];
        implementedU2 = [implementedU2;U2(2,2)];


    end

    x1final = [x1final;x1hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x2final = [x2final;x2hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x3final = [x3final;x3hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x4final = [x4final;x4hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x5final = [x5final;x5hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];
    x6final = [x6final;x6hold((j-2)*(Khorizon-1) + 1:(j-1)*(Khorizon-1))];

end

implementedU1 = reshape(implementedU1,Jhorizon-1,Khorizon-1)';
implementedU1 = [zeros(1,Khorizon-1);implementedU1];
implementedU1 = [zeros(Jhorizon,1) implementedU1];
implementedU2 = reshape(implementedU2,Jhorizon-1,Khorizon-1)';
implementedU2 = [zeros(1,Khorizon-1);implementedU2];
implementedU2 = [zeros(Jhorizon,1) implementedU2];
x1final = [zeros(1,Khorizon-1);x1final];                    %arrival
x1final = [zeros(Jhorizon,1) x1final];
x2final = [zeros(1,Khorizon-1);x2final];                    %departure
x2final = [zeros(Jhorizon,1) x2final];
x2final(1,1:Khorizon)=((1:Khorizon)-1)*120;
x2final(1:Jhorizon,1)=((1:Jhorizon)'-1)*240;
x3final = [zeros(1,Khorizon-1);x3final];                    %q
x3final = [zeros(Jhorizon,1) x3final];
x4final = [120*ones(1,Khorizon-1);x4final];                  %rho
x4final = [120*ones(Jhorizon,1) x4final];
x5final = [zeros(1,Khorizon-1);x5final];                    %sigma
x5final = [zeros(Jhorizon,1) x5final];
x6final = [zeros(1,Khorizon-1);x6final];                    %p wait
x6final = [zeros(Jhorizon,1) x6final];
```

Appendix B

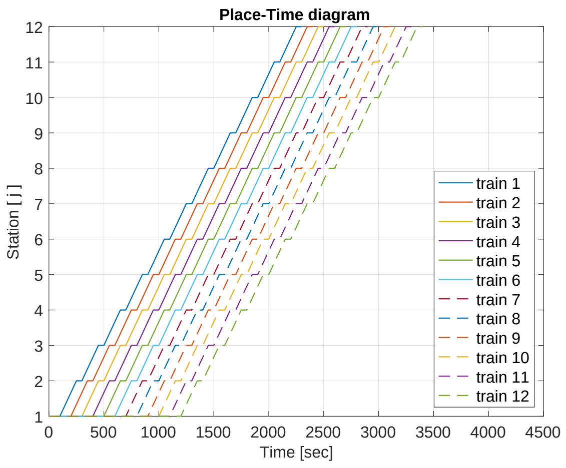
Simulations

In this Appendix, additional figures of the simulations are shown. For each simulation of the Urban Railway System (URS) subject to a type of disturbance, we present the remaining states that have not been presented in the main body of the thesis.

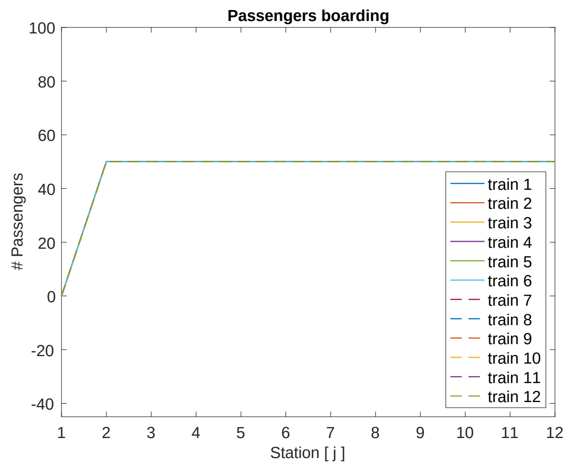
B-1 No Control - Initial Conditions

In subsection 5-5-1 we have simulated the URS using the script in section A-1, without any control implemented. We have investigated three different sets of initial values: two sets of initial values that result in steady-state evolution of the URS, and one set of initial values for which the URS evolves in an unstable manner. In Figure 5-1, the evolution of all states of the URS are shown for the initial values as defined in section 5-4. In Figure B-1, we present the additional figures for the state evolution of the URS subject to the initial values as defined in Equation 5-77. It can be observed in Figure B-1a that the dwell time is constant resulting in a constant train schedule. It can furthermore be observed that the remaining states all settle at constant values:

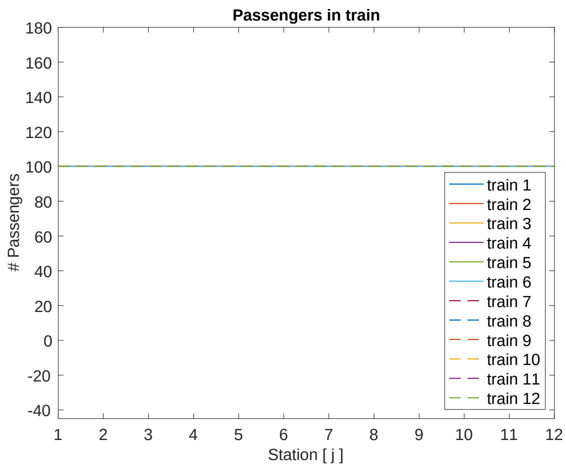
- $q_j(k)$ stabilizes at 50 passengers;
- $\rho_j(k)$ stabilizes at 100 passengers;
- $\sigma_j(k)$ stabilizes at 0 passengers;
- $p_j^{\text{wait}}(k)$ stabilizes as 25 passengers.



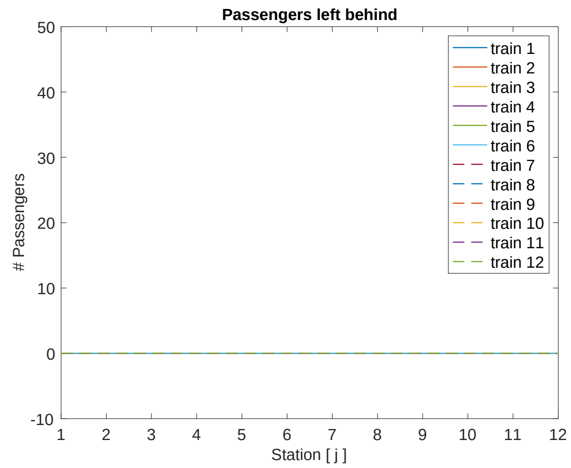
(a) $a_j(k)$, $\tau_{\text{dwell},j}(k)$ and $d_j(k)$



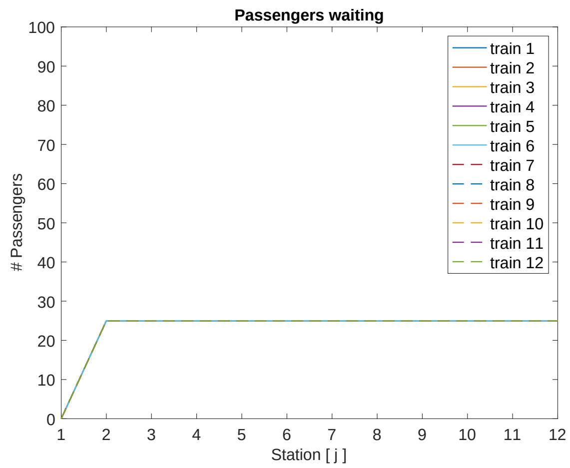
(b) $q_j(k)$



(c) $\rho_j(k)$



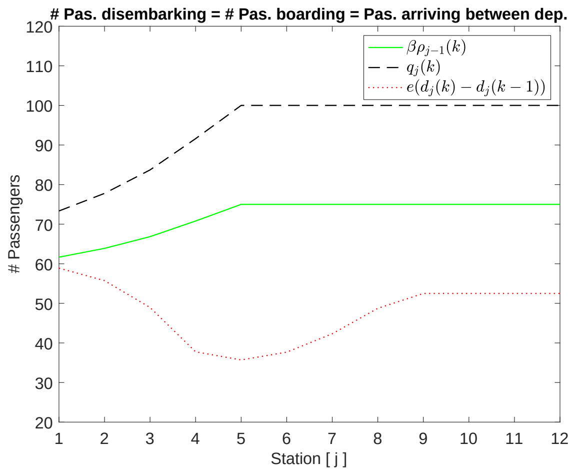
(d) $\sigma_j(k)$



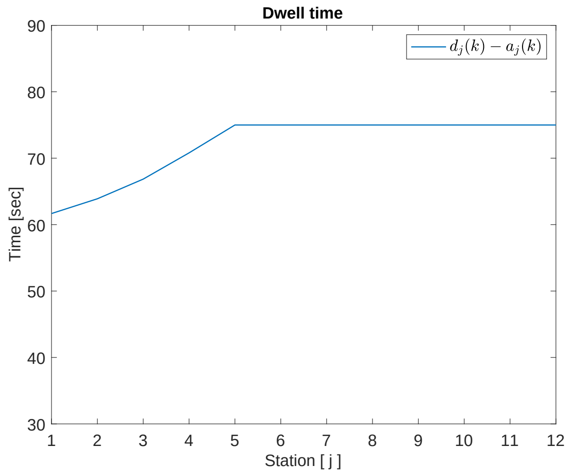
(e) $p_j^{\text{wait}}(k)$

Figure B-1: States of the URS for the initial values as defined in Equation 5-77, for J = 12 and K = 12.

Finally, let us present the additional figures to the third set of initial values we have investigated as defined in Equation 5-78. Figure B-2 shows the steady-state conditions for these initial values. It can be observed that these conditions are not satisfied; Figure B-2a shows that the number of passengers disembarking, the number of passengers boarding and the number of passengers arriving between consecutive departures are not equal. Furthermore we can see in Figure B-2b that the dwell time is not constant over all stations. In Figure 5-4 we have seen the evolution of the place-time diagram and the number of passengers waiting, and Figure B-3 shows the evolution of the remaining states of the URS subject to the initial values as defined in Equation 5-78. It can be observed that without any control, the URS evolves in an unstable manner due to the initial conditions.

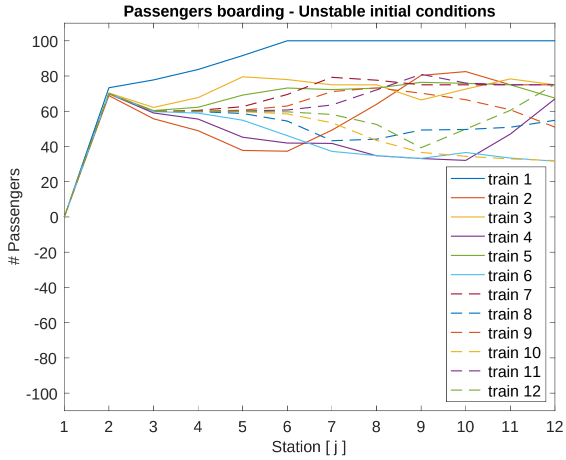


(a) Steady-state conditions 1. and 2.

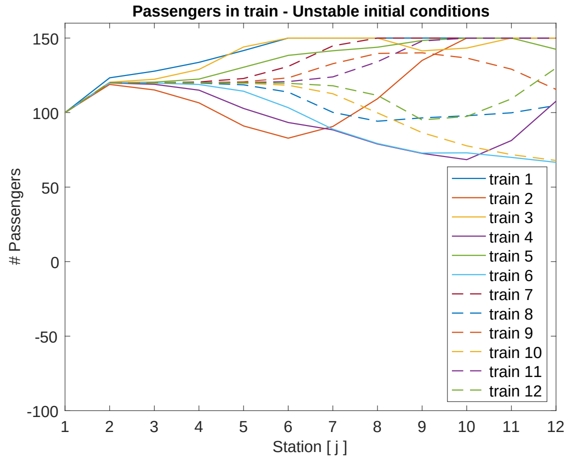


(b) Steady-state condition 3.

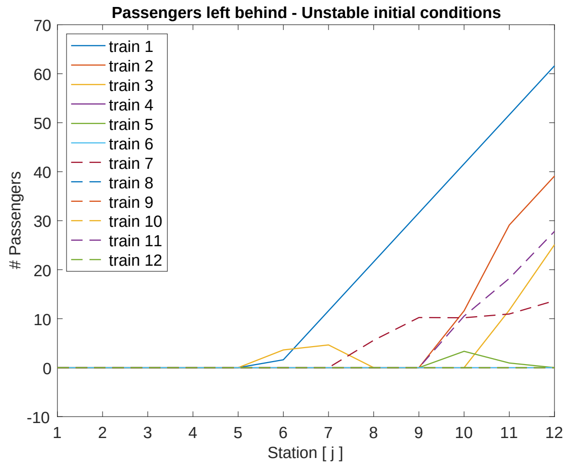
Figure B-2: Steady-state conditions for initial values as defined in Equation 5-78 without control implemented, for $J = 12$ and $K = 12$.



(a) $q_j(k)$



(b) $\rho_j(k)$



(c) $\sigma_j(k)$

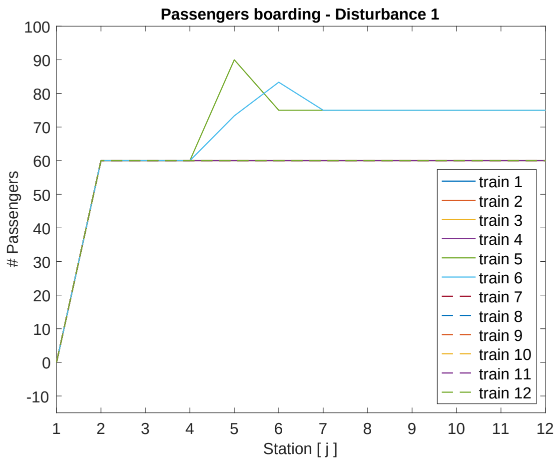
Figure B-3: Remaining states of the URS with initial values as defined in Equation 5-78 without control implemented, for J = 12 and K = 12.

B-2 Control - Disturbance Scenarios

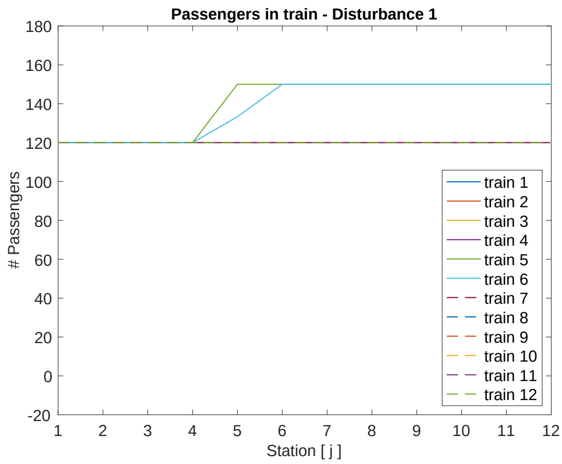
In subsection 5-5-2 we have exposed the URS to several disturbance scenarios, and we have observed the performance of the Model Predictive Controller in rejecting these disturbances. In subsection 5-5-2, we have included the place-time diagrams and the figure presenting the number of passengers waiting of the URS for all disturbances without and with control. In this appendix we present the remaining figures of $q_j(k)$, $\rho_j(k)$ and $\sigma_j(k)$, and we will present the implemented control effort $u_{1,j}(k)$ and $u_{2,j}(k)$.

B-2-1 Disturbance 1

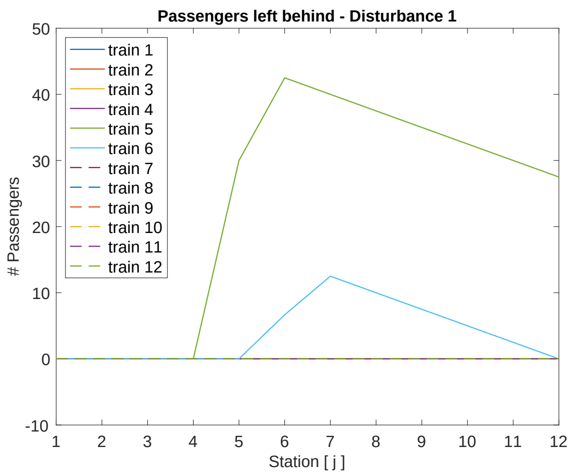
In Figure B-4, we can observe the evolution of the remaining states of the URS when exposed to Disturbance 1. It can be observed that $q_j(k)$ and $\rho_j(k)$ of the trains 5 and 6 are stabilized at constant values after being exposed to the disturbance due to the control effort which can be observed in Figure B-5. Furthermore, we can observe that $\sigma_j(k)$ evolves to zero.



(a) $q_j(k)$

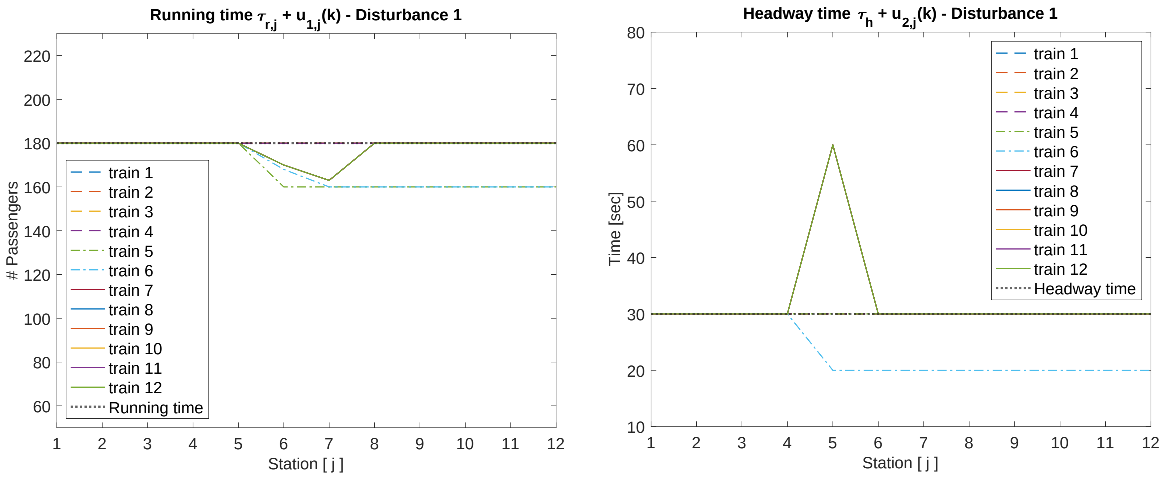


(b) $\rho_j(k)$



(c) $\sigma_j(k)$

Figure B-4: Remaining states of the URS with Disturbance 1 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

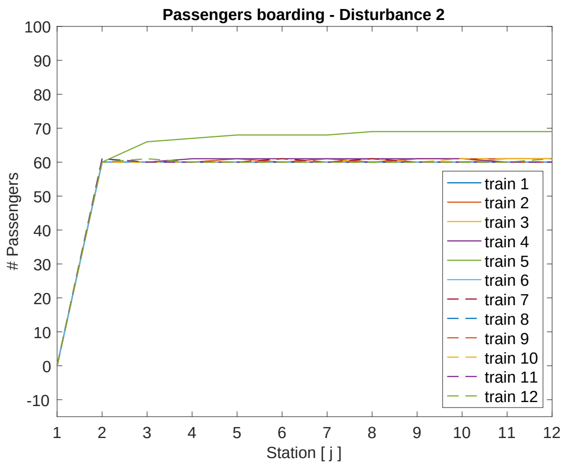


(a) $\tau_{r,j} + u_{1,j}(k)$ **(b)** $\tau_h + u_{2,j}(k)$

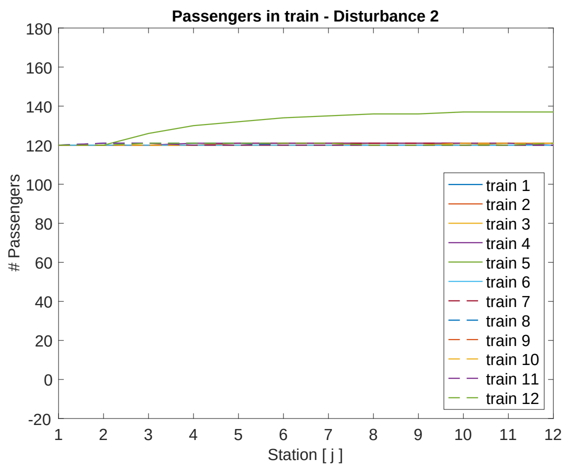
Figure B-5: Implemented control on the URS subject to Disturbance 1 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

B-2-2 Disturbance 2

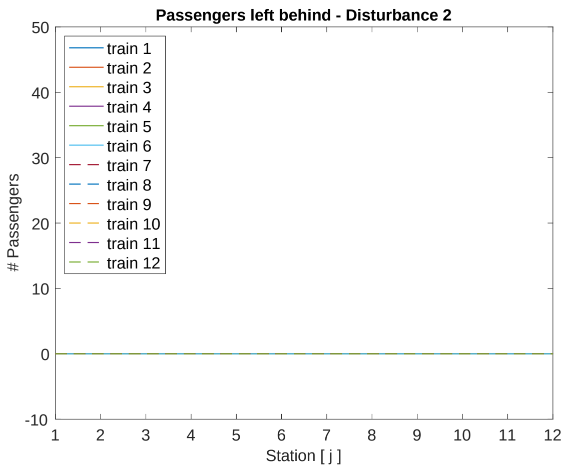
In Figure B-6, we can observe the evolution of the remaining states of the URS when exposed to Disturbance 2. It can be observed that all states are kept within boundaries due to the control effort which can be observed in Figure B-7.



(a) $q_j(k)$

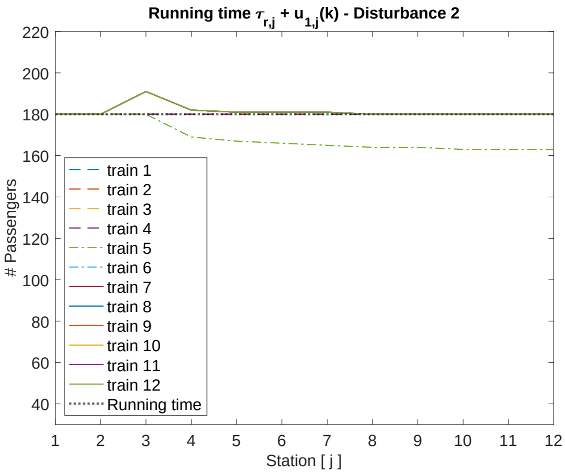


(b) $\rho_j(k)$

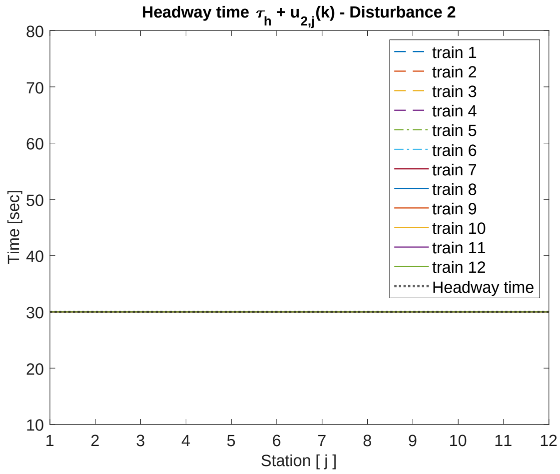


(c) $\sigma_j(k)$

Figure B-6: Remaining states of the URS with Disturbance 2 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.



(a) $\tau_{r,j} + u_{1,j}(k)$

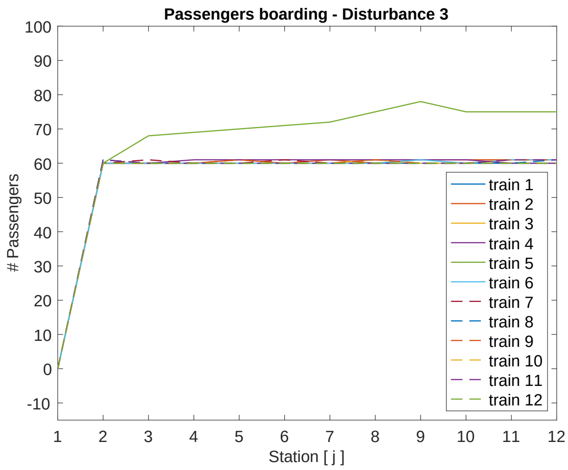


(b) $\tau_h + u_{2,j}(k)$

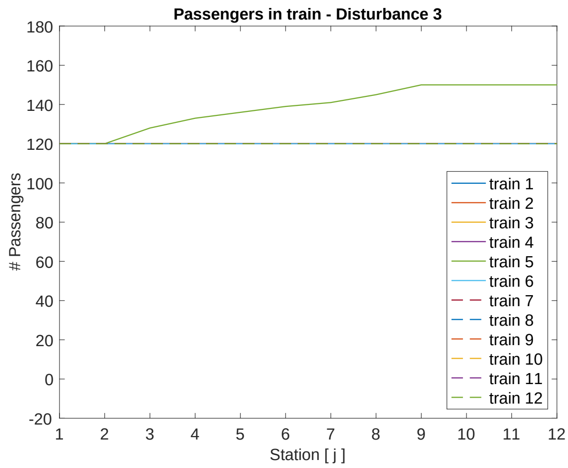
Figure B-7: Implemented control on the URS subject to Disturbance 2 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

B-2-3 Disturbance 3

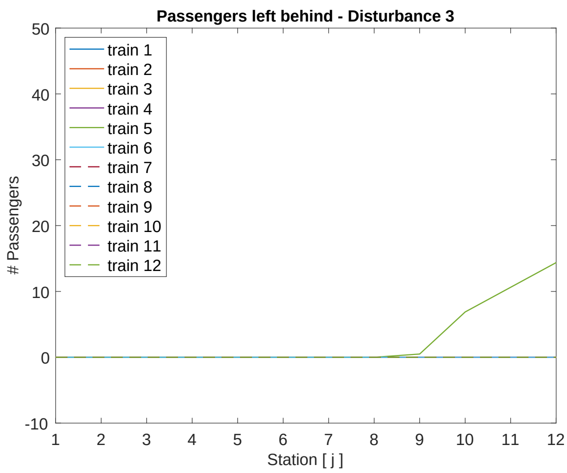
In Figure B-8, we can observe the evolution of the remaining states of the URS when exposed to Disturbance 3. It can be observed that the running time input is implemented on lower boundary level in Figure B-9a. However, this input is not enough to stabilize the URS, as we have seen in Figure 5-10b that $p_j^{\text{wait}}(k)$ blows up for train $k = 5$. We can furthermore see in Figure B-8c that the number of passengers left behind also blows up for train $k = 5$, due to the maximum capacity that is reached for train $k = 5$ and $k = 6$ as can be observed in Figure B-8b.



(a) $q_j(k)$

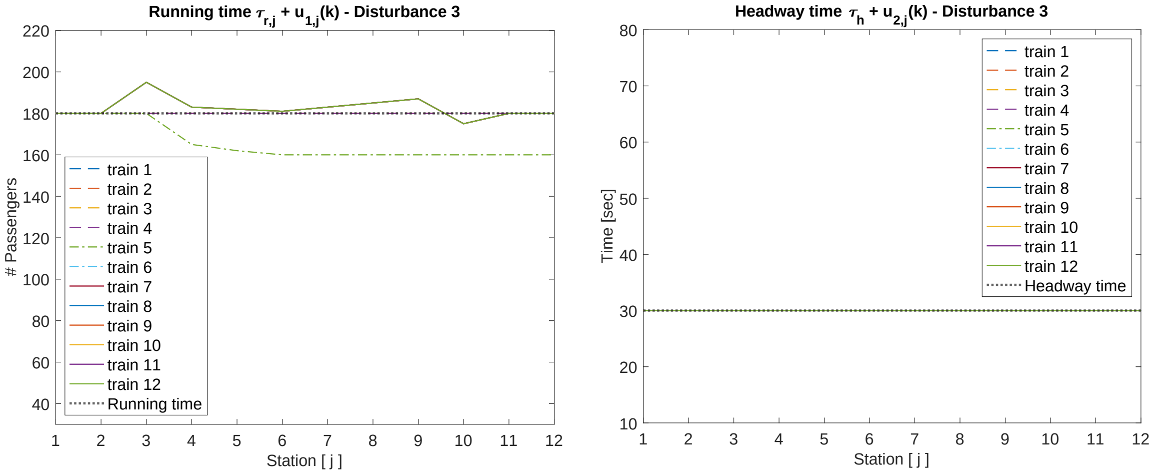


(b) $\rho_j(k)$



(c) $\sigma_j(k)$

Figure B-8: Remaining states of the URS with Disturbance 3 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

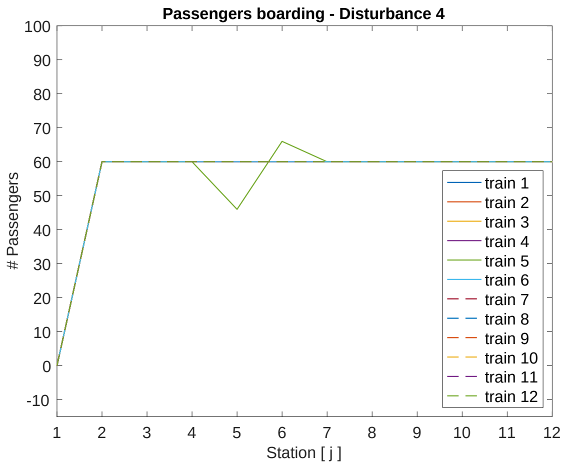


(a) $\tau_{r,j} + u_{1,j}(k)$ **(b)** $\tau_h + u_{2,j}(k)$

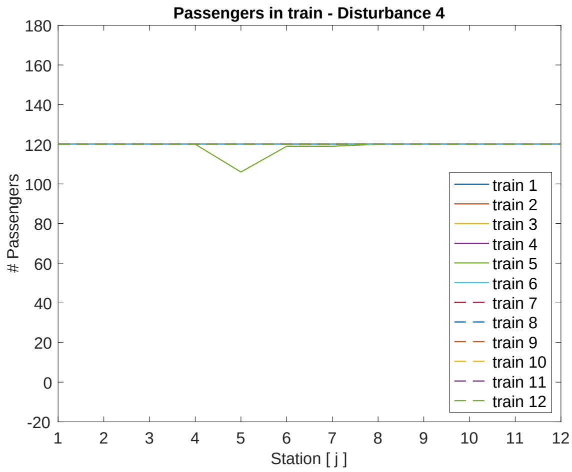
Figure B-9: Implemented control on the URS subject to Disturbance 3 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

B-2-4 Disturbance 4

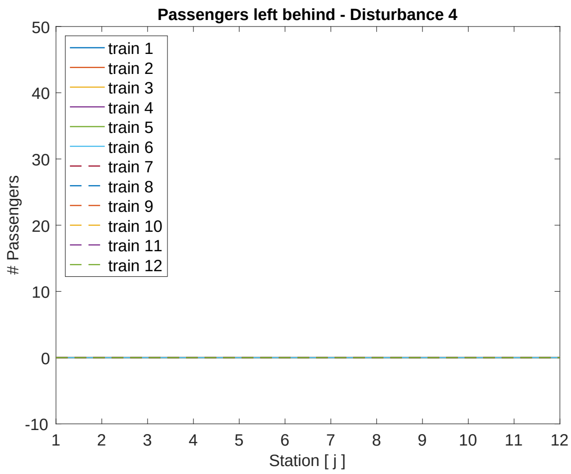
In Figure B-10, we can observe the evolution of the remaining states of the URS when exposed to Disturbance 4. It can be observed that $q_j(k)$ and $\rho_j(k)$ are stabilized at constant values after being exposed to the disturbance, due to the control effort which can be observed in Figure B-5. Furthermore, we can observe that $\sigma_j(k)$ remains zero.



(a) $q_j(k)$

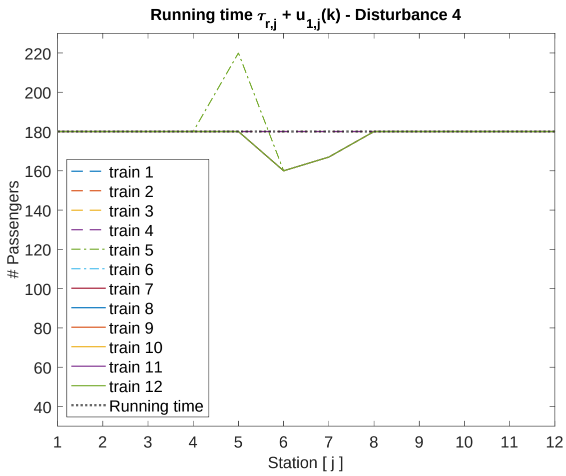


(b) $\rho_j(k)$

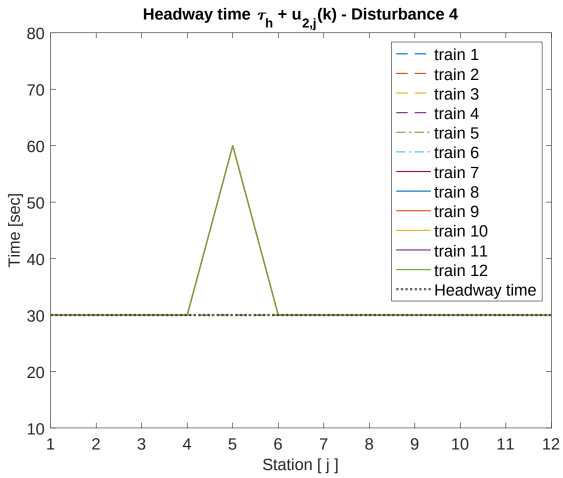


(c) $\sigma_j(k)$

Figure B-10: Remaining states of the URS with Disturbance 4 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.



(a) $\tau_{r,j} + u_{1,j}(k)$

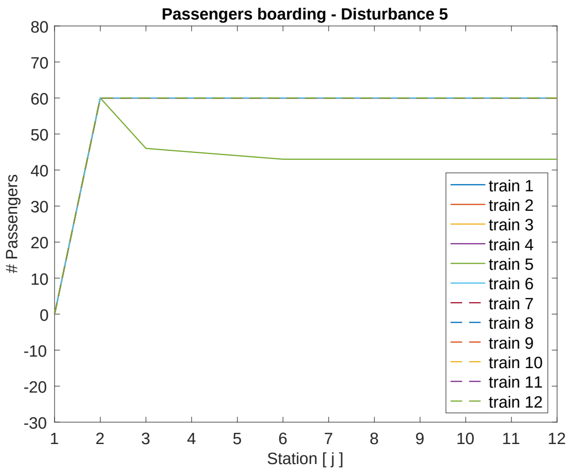


(b) $\tau_h + u_{2,j}(k)$

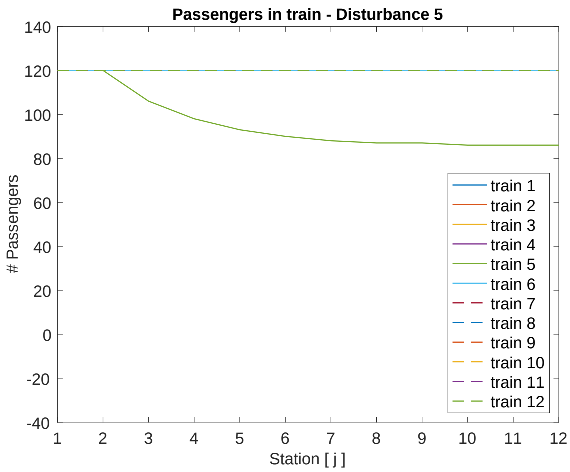
Figure B-11: Implemented control on the URS subject to Disturbance 4 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

B-2-5 Disturbance 5

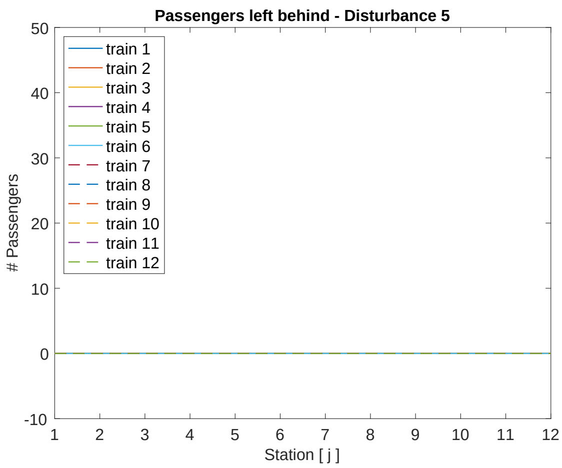
In Figure B-12, we can observe the evolution of the remaining states of the URS when exposed to Disturbance 5. It can be observed that $q_j(k)$ and $\rho_j(k)$ of the disturbed train $k = 5$ are stabilized at constant values due to the control effort which can be observed in Figure B-13. Furthermore, we can observe that $\sigma_j(k)$ remains at zero.



(a) $q_j(k)$

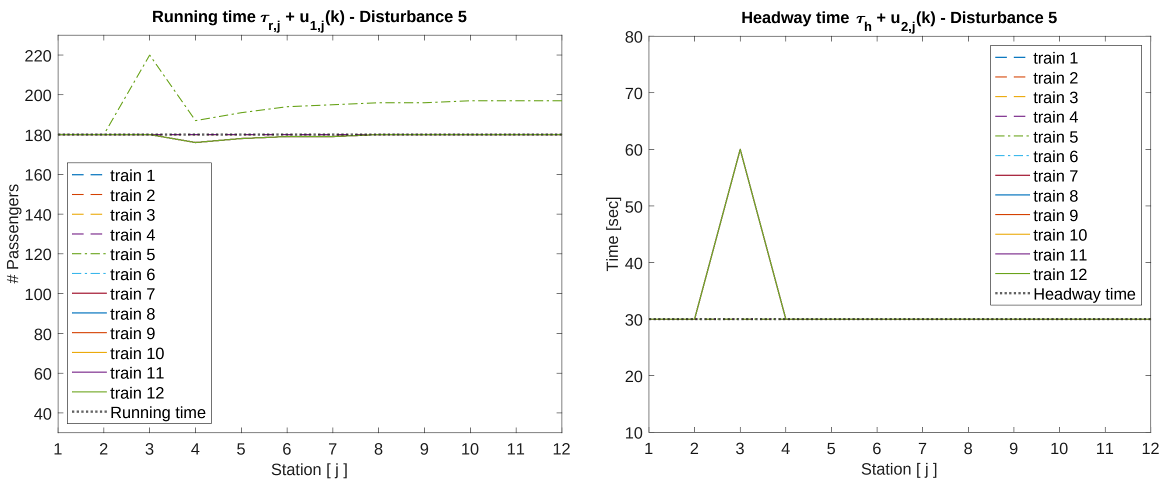


(b) $\rho_j(k)$



(c) $\sigma_j(k)$

Figure B-12: Remaining states of the URS with Disturbance 5 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

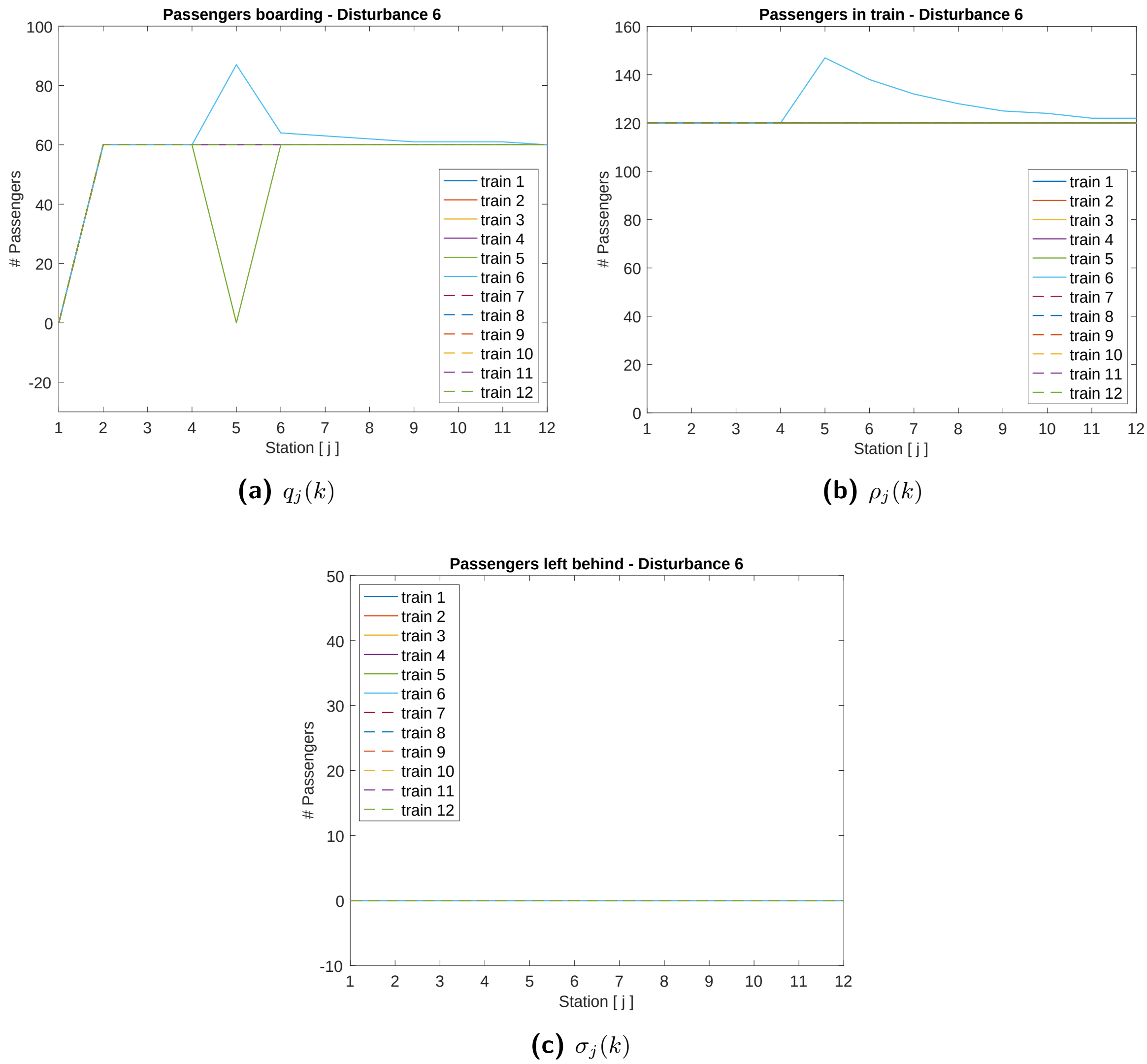


(a) $\tau_{r,j} + u_{1,j}(k)$ **(b)** $\tau_h + u_{2,j}(k)$

Figure B-13: Implemented control on the URS subject to Disturbance 5 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12.

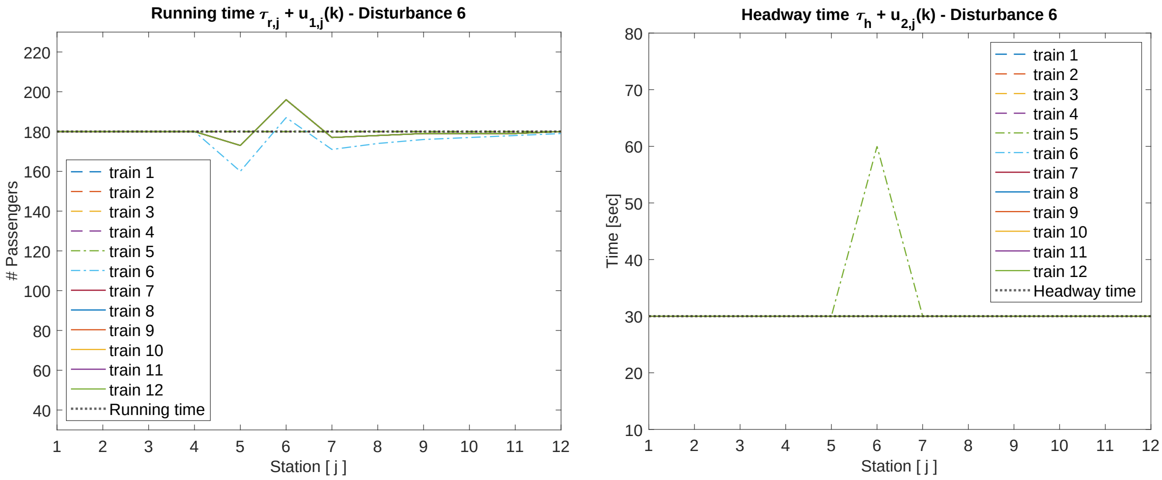
B-2-6 Disturbance 6

In Figure B-14, we can observe the evolution of the remaining states of the URS when exposed to Disturbance 6. It can be observed that $q_j(k)$ and $\rho_j(k)$ of the disturbed trains $k = 5$ and $k = 6$ are stabilized at constant values after being exposed to the disturbance due to the control effort which can be observed in Figure B-15. Furthermore, we can observe that $\sigma_j(k)$ remains at zero.



(a) $q_j(k)$ **(b)** $\rho_j(k)$ **(c)** $\sigma_j(k)$

Figure B-14: Remaining states of the URS with Disturbance 6 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12 and a minimum dwell time set on $\tau_{\text{dwell,min}} = 0$ seconds.



(a) $\tau_{r,j} + u_{1,j}(k)$ **(b)** $\tau_h + u_{2,j}(k)$

Figure B-15: Implemented control on the URS subject to Disturbance 6 for the initial values and parameters as defined in section 5-4, for J = 12 and K = 12 and a minimum dwell time set on $\tau_{\text{dwell,min}} = 0$ seconds.

Bibliography

[1] D. E. R. Beek, *An agent-based opinion dynamics model with a language model-based belief system - Briding language and opinion modeling.* PhD thesis, Delft University of Technology, Center of Systems and Control, 2021.

[2] M. Pesselse, *Modelling and Optimal Scheduling of Inland Waterway Transport Systems: A Switching Max-Plus Linear Systems Approach.* PhD thesis, Delft University of Technology, Center of Systems and Control, 2022.

[3] B. de Jong, *Throughput and Stabilisability Analysis of Mode-Constrained Stochastic Switching Max-Plus Linear Systems.* PhD thesis, Delft University of Technology, Center of Systems and Control, 2022.

[4] R. E. S. Beek, "Literature Research: Max-Plus Linear Varying Systems - An Overview on Subclasses," 2021.

[5] G. Cohen, S. Gaubert, and J.-P. Quadrat, "Max-plus algebra and system theory: Where we are and where we go now," *Annual Reviews in Control (IFAC, Elsevier)*, 1999.

[6] T. van den Boom and B. De Schutter, "Max-plus algebra and max-plus linear discrete event systems: An introduction," *Proceedings of the 9th International Workshop on Discrete Event Systems (WODES'08)*, pp. 36–42, 2008.

[7] S. Masuda, H. Goto, T. Amemiya, and K. Takeyasu, "An inverse system for linear parameter-varying max-plus-linear systems," *Proceedings of the IEEE Conference on Decision and Control*, vol. 4, pp. 4549–4554, 2002.

[8] M. Abdelmoumni, A. Gupta, J. van der Woude, and T. van den Boom, "Max-plus linear parameter-varying systems: An application to urban railway," *Internal report*, 2021.

[9] P. Gáspár, Z. Szabó, J. Bokor, and B. Németh, *Robust Control Design for Active Driver Assistance Systems - A Linear-Parameter-Varying Approach.* Springer International Publishing Switzerland, 2017.

[10] M. Abdelmoumni, "Max-plus linear parameter varying systems - A framework and solvability," Master's thesis, Technical University of Delft, 2021.

[11] B. De Schutter and T. van den Boom, "Model predictive control for max-plus-linear discrete event systems," *Automatica*, vol. 37, pp. 1049–1056, 2001.

[12] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat, *Synchronization and Linearity.* John Wiley & Sons, New York, 1992.

[13] B. Heidergott, G. J. Olsder, and J. van der Woude, *Max Plus at Work.* Princeton University Press., 2005.

[14] B. De Schutter and M. Heemels, *Modeling and Control of Hybrid Systems: A Survey.* Lecture Notes for the course SC4160, Delft Center for Systems and Control, Delft University of Technology and Hybrid & Networked Systems group, Eindhoven University of Technology, 2015.

[15] B. D. Schutter, T. van den Boom, J. Xu, and S. S. Farahani, "Analysis and control of max-plus linear discrete-event systems: An introduction," *Discrete Event Dynamic Systems: Theory and Applications 30*, p. 25–54, March 2020.

[16] P. Butkovič, *Max-linear Systems: Theory and Algorithms.* Springer Monographs in Mathematics, Springer London, 2010.

[17] R. M. Karp, "A characterization of the minimum cycle mean in a digraph," *Discrete Mathematics*, vol. 23, no. 3, pp. 309–311, 1978.

[18] J. G. Braker and G. J. Olsder, "The power algorithm in max algebra," *Linear Algebra and Its Applications*, vol. 182, pp. 67–89, 1993.

[19] B. Zwerink Arbonés, "An exploratory study in max-plus linear parameter varying systems," Master's thesis, Delft, University of Technology, 2019.

[20] G. A. Lopes, B. Kersbergen, T. J. van den Boom, B. De Schutter, and R. Babuska, "Modeling and control of legged locomotion via switching max-plus models," *IEEE Transactions on Robotics*, vol. 30, pp. 652–665, June 2014.

[21] B. De Schutter, "The extended linear complementarity problem and its applications in analysis and control of discrete-event systems," *Springer Optimization and Its Applications*, vol. 17, pp. 541–570, 2008.

[22] B. Kersbergen, T. van den Boom, and B. De Schutter, "On implicit versus explicit max-plus modeling for the rescheduling of trains," *Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis (RailCopenhagen)*, p. 15, May 2013.

[23] J. W. Larsen and S. R. Nielsen, "Non-linear dynamics of wind turbine wings," *International Journal of Non-Linear Mechanics*, vol. 41, no. 5, pp. 629–643, 2006.

[24] J. S. Shamma, *Analysis and design of gain scheduled control systems.* PhD thesis, Massachusetts Institute of Technology, 1988.

[25] J. S. Shamma and M. Athans, "Gain scheduling: Potential hazards and possible remedies," *IEEE control systems*, vol. 12, pp. 101 – 107, 1992.

[26] J. Mohammadpour and C. W. Scherer, *Control of Linear Parameter Varying Systems with Applications.* Springer Science+Business Media, LLC, 2012.

[27] R. Tóth, J. C. Willems, P. S. Heuberger, and P. M. van den Hof, "The behavioral approach to linear parameter-varying systems," *IEEE Transactions on Automatic Control*, vol. 56, pp. 2499–2514, November 2011.

[28] R. Toth, *Modeling and Identification of Linear Parameter-Varying Systems.* Springer-Verlag Berlin Heidelberg, 2010.

[29] M. Schoukens and R. Tóth, "Linear parameter varying representation of a class of MIMO nonlinear systems," *IFAC-PapersOnLine*, vol. 51, no. 26, pp. 94–99, 2018.

[30] J. De Caigny, R. Pintelon, J. F. Camino, and J. Swevers, "Interpolated modeling of LPV systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2232–2246, 2014.

[31] H. H. Lin, C. Beck, and M. Bloom, "Multivariable LPV control of anesthesia delivery during surgery," *Proceedings of the American Control Conference*, pp. 825–831, 2008.

[32] S. Ç. Baslamisli, I. E. Köse, and G. Anlaş, "Gain-scheduled integrated active steering and differential control for vehicle handling improvement," *Vehicle System Dynamics*, vol. 47, no. 1, pp. 99–119, 2009.

[33] F. Bianchi, H. de Battista, and R. Mantz, *Wind Turbine Control Systems.* Springer, 2007.

[34] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems.* Cambridge University Press, Cambridge, UK, 2017.

[35] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control. Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[36] C. R. Cutler and B. L. Ramaker, "Dynamic Matrix Control - A Computer Control Algorithm," *Proceedings of the 86th AIChE National Meeting*, vol. 1, 1979.

[37] B. De Schutter and T. van den Boom, "On model predictive control for max-min-plus-scaling discrete event systems," *Systems Engineering*, vol. 19, 2000.

[38] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[39] A. Bemporad, F. Borrelli, and M. Morari, "Optimal controllers for hybrid systems: stability and piecewise linear explicit form," *Proceedings of the IEEE Conference on Decision and Control 2*, vol. 2, pp. 1810 – 1815, 2000.

[40] Y. Wang, T. Ning, B. van den Boom, and B. De Schutter, *Optimal Trajectory Planning and Train Scheduling for Urban Rail Transit Systems.* Springer International Publishing Switzerland 2016, 2016.

[41] A. Gupta, T. van den Boom, J. van der Woude, and B. De Schutter, "Framework for Studying Stability of Switching Max-Plus Linear Systems," *Submitted to Workshop on Discrete event systems (WODES)*, 2020.

[42] J. B. Rawlings, D. Q. Mayne, and D. M. Moritz, *Model predictive control: Theory, Computation and Design.* Nob Hill Publishing, LLC, 2nd ed., 2019.

[43] B. De Schutter, "Stability Analysis of Discrete Event Systems (by K.M. Passino and K.L. Burgess, A volume in the Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control (S. Haykin, ed.)," *Automatica*, vol. 37, pp. 799–801, May 2001.

Glossary

List of Acronyms

MP-LPV	Max-Plus Linear Parameter Varying
MPL	Max-Plus-Linear
DES	Discrete Event Systems
LPV	Linear Parameter Varying
MMPS	Max-Min-Plus-Scaling
URS	Urban Railway System
MPC	Model Predictive Control
MLD	Mixed Logical Dynamical
PWA	Piecewise Affine
MLD	Mixed Logical Dynamical
LTI	Linear Time Invariant
SMPL	Switching Max-Plus Linear
MPS	Max-Plus-Scaling
MILP	Mixed Integer Linear Programming
MIQP	Mixed Integer Quadratic Programming
DTS	Discrete Time Systems

List of Symbols

Δ	Difference operator, $\Delta x(k) = x(k) - x(k-1)$
Λ	Target set of doubly implicit MP-LPV systems ensuring (unique) existence
λ	Eigenvalue
Λ_{inv}	Maximal positive invariant set of doubly implicit MP-LPV systems ensuring (unique) solvability

$\mathbb{R}$	Set of real numbers
$\mathbb{R}_{\max}$	Max-plus semiring
$\mathbb{R}_\varepsilon$	The set $\mathbb{R} \cup \{-\infty\}$
$\mathbb{Z}_+$	Set of (positive) integers
$\mathcal{C}$	Convex set
$\mathcal{D}$	Set of links associated with graph $\mathcal{G}$
$\mathcal{D}_{\mathrm{ad}}$	Admissible Disturbance Set
$\mathcal{E}$	Max-plus null matrix
$\mathcal{G}(A)$	Communication graph of matrix A
$\mathcal{G}^c(A)$	Critical graph of matrix A
$\mathcal{N}$	Set of nodes associated with graph $\mathcal{G}$
$\mathcal{P}$	Target set for (unique) existence of single implicit MP-LPV systems
$\mathcal{P}_{\mathrm{inv}}$	Maximal positive invariant set of single implicit MP-LPV systems ensuring (unique) solvability
$\mathcal{V}$	Halfspace
$\mathcal{X}$	Polyhedral set
$\oplus$	Max-plus addition operator or maximization ('oplus')
$\otimes$	Max-plus multiplication operator or addition ('otimes')
$\stackrel{\mathrm{def}}{=}$	Is defined as
$\tilde{A}$	Frobenius normal form of matrix A
ε	Max-plus zero element, $\varepsilon = -\infty$
A^*	Kleene star of matrix A
A_λ	Normalized matrix of matrix A
E	Max-plus identity matrix
e	Max-plus unit element, $e = 0$
f_{cPWA}	Continuous PWA function
f_{MPS}	Max-plus scaling function
P	Parameter set
$p(k)$	Parameter vector with standard length r
P_ε	Max-plus parameter set
$u(k)$	Input vector with standard length m
v	Eigenvector
$x(k)$	State vector with standard length n
$y(k)$	Output vector with standard length l
$\mathrm{Pre}(\Lambda)$	Precursor set of Λ
$\mathrm{Pre}(\mathcal{P})$	Precursor set of $\mathcal{P}$