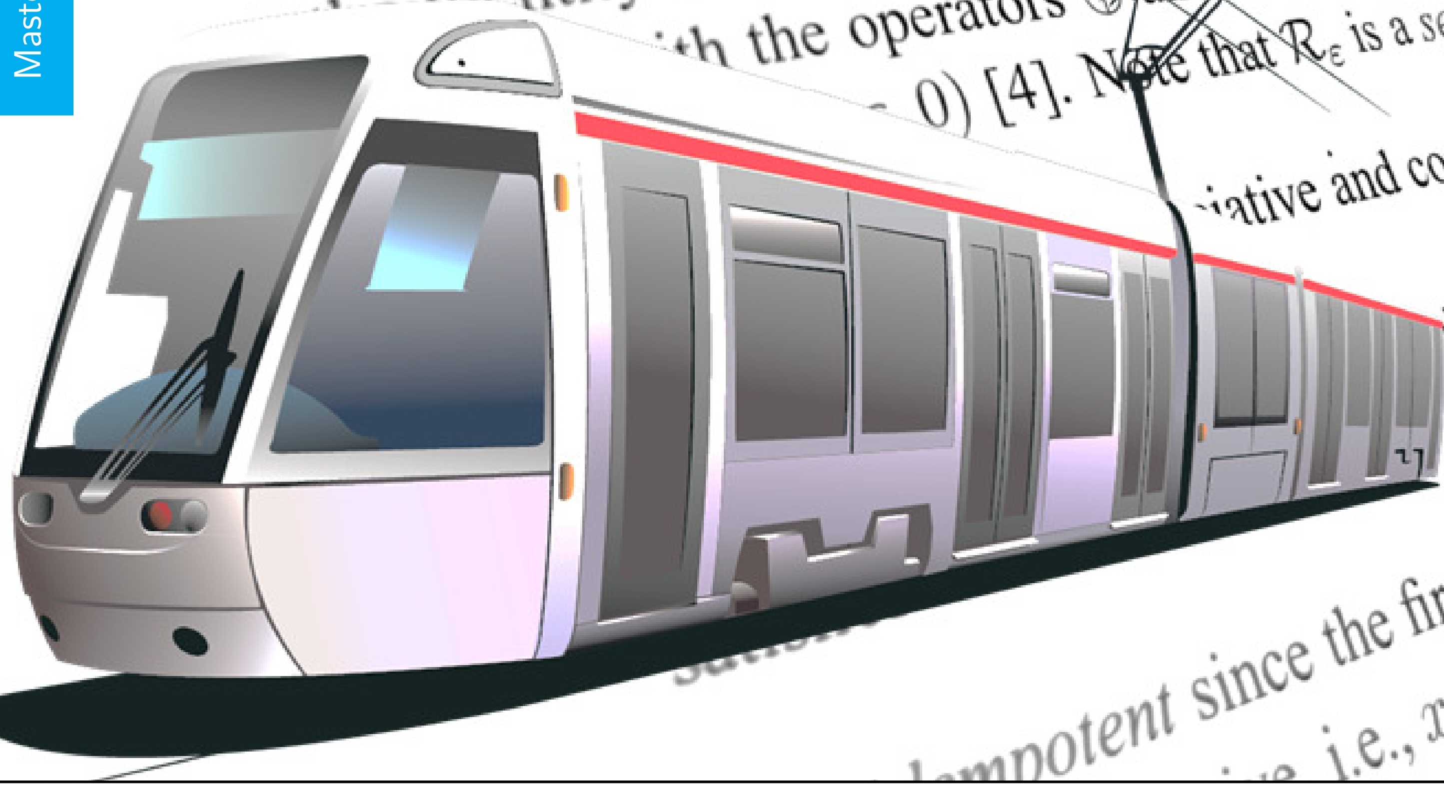
Model Predictive Control on Max-min-plus-scaling Systems

Control procedure and stability conditions

J.C.M. Kroese

Master of Science Thesis



Delft University of Technology

Delft Center for Systems and Control

Model Predictive Control on Max-min-plus-scaling Systems

Control procedure and stability conditions

Master of Science Thesis

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

J.C.M. Kroese

November 7, 2022

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

Abstract

Max-plus-linear (MPL) systems are systems that are linear in max-plus algebra. A generalization of these systems are Max-Min-Plus-Scaling (MMPS) systems. Next to maximization and addition (plus), MMPS systems use the operations minimization and scaling. They are discrete-event (DE) systems, which means that the changing of the states is triggered by the occurrence of events and (part of) the states in the state vector represent time instances. One way to control MMPS systems is by using Model predictive control (MPC). This is a powerful on-line control strategy that uses a receding horizon. However, an efficient control procedure that works for all time-invariant DE MMPS systems had not yet been described. The goal of this master thesis is to fully design such a framework. To achieve this, the state vector is altered, such that the difference in the states that represent a time instance is included as well. Next to this, the MPC problem on an MMPS system is altered to a Mixed integer quadratic programming (MIQP) problem, in order to optimize it more efficiently. That this framework works is supported by a stability analysis. Next to that, it is tested on a simulation example of an urban railway line. Based on this example, it is shown that the procedure does indeed work. The thesis ends with several suggestions for future research.

Contents

Preface and Acknowledgements

Dear reader,

This is it. My last ever assignment at the TU Delft. I was not sure what I would think about working on an assignment for such a long time and, especially, doing so on my own. I have to say it was definitely not always easy, but it was also a lot better than I expected. To dive into a subject for such a long time, was a great experience and I learnt a lot from it. This is why I would like to thank everyone that helped me along the way.

I would like to start off by thanking my supervisor Ton van den Boom. The meetings we had every other week were definitely of great help. Next to that, your enthusiasm about the subject always made me leave the meeting feeling highly motivated to continue.

Furthermore, I would like to thank my study friends and in particular Femke, with whom I have taken not only all my bachelor courses, but also almost all of my master courses. It was great to have someone to talk to about my thesis, who could actually fully understand what I was doing.

Next to this, I would like to thank all my friends who have asked me how my thesis was going and how I was doing during the entire process and who reminded that I should not forget to enjoy my non-working life for as long as I could, when I was maybe putting a bit too much time in.

Lastly, I would like to thank my parents and my brother. My parents have always been a great support and have always celebrated my achievements without ever putting pressure on me. My brother has been a great role model in his academic achievements, but definitely in life as well.

I hope you enjoy the read,

Justine

Delft, University of Technology J.C.M. Kroese
November 7, 2022

"I never understood how anyone could feel small compared with the universe. After all, man knows how overwhelmingly large it is, and a few others things besides, and that means he is not small. The fact that man has discovered all this precisely proves his greatness."

— *Harry Mulisch*

Chapter 1

Introduction

1-1 Background

In this research the emphasis is on a relatively new kind on algebra: max-plus algebra. The system theory where max-plus algebra plays a central role, emerged in the early 1980's [1]. Here, the conventional plus and times operators are replaced by a maximization and a plus operator, respectively. The advantage of this translation is that some systems that would be nonlinear in conventional (plus-times) algebra can be described in a linear way in max-plus algebra [2]. These systems are called Max-plus-linear (MPL) systems and are used to model discrete-event systems with synchronization, but no choice. The presence of choice can lead to the necessity of a minimization operator [3].

A system that consists, next to the operations maximization and addition, of the operation minimization and scaling, is called an Max-Min-Plus-Scaling (MMPS) system. This notation opens up possibilities for even more kinds of systems to be described more efficiently or intuitively. This is why this research focuses on MMPS systems.

In MMPS (and MPL) systems, the evolution parameter that is used is an event counter, instead of the conventional time increment. This is why these systems are called discrete-event (DE) systems. So the evolution of these systems depends on the occurrence of an event. Next to this, (part of) the states in the state vector represent time instances. This prevents the straightforward translation of plus-times properties and algorithms to the max-plus environment.

To optimize a DE MMPS system, a sensible choice would be to use Model predictive control (MPC). This is a powerful on-line control strategy that uses a receding horizon. However, MPC problems for MMPS systems are in general nonconvex nonlinear optimization problems. These problems are known to be hard to solve [2]. Previous research has been able to simplify this control problem in specific situations to problems that can be solved more efficiently. However, an efficient control procedure that works for all time-invariant DE MMPS system has not yet been described.

1-2 Problem description

Several areas have be identified that are in need of further research, in order to describe the complete control procedure for a general time-invariant DE MMPS system. These areas are based on the literature study conducted previously. To guide the investigation, that will be conducted in this master thesis, multiple research questions are formulated. The content in this work is constructed in such a way that these questions can be solved in a structured manner.

1-2-1 Research questions

- How can the process of designing a stabilizing MPC controller for a time-invariant MMPS system be fully described?
  - How should the state vector of a time-invariant MMPS system be defined?
  - How can a time-invariant MMPS system be linearized?
  - How should the objective function be defined to guarantee stability for a time-invariant MMPS system?
  - What will the terminal set for a time-invariant MMPS system look like?

1-2-2 Approach

To answer the research questions, the master thesis will start off with a chapter that will introduce max-plus algebra and the MMPS system. After this, Chapter 3 will present the conventional MPC method applied to a nonlinear discrete-time (DT) system. Furthermore, it raises several concerns that could prevent a straightforward translation to DE MMPS systems. Next, Chapter 4 will address these concerns and Chapter 5 will show the entire process of recasting a general MMPS system as an Mixed integer quadratic programming (MIQP) problem.

After the entire process for designing a stabilizing MPC controller has been described, the procedure will be applied to a case study: the model of an urban railway line. In this way it can be demonstrated that the proposed method does indeed work in practice. The thesis work will end with the main conclusions and contributions of this master thesis and it will give recommendations for further research.

Chapter 2

Max-Min-Plus-Scaling Systems

This chapter starts with an explanation of the basics of discrete-event (DE) systems and Max-plus algebra. Next to that, Max-plus-linear (MPL) systems will be introduced. After this, the chapter continues with a generalization of these systems, namely Max-Min-Plus-Scaling (MMPS) systems and it explains the concept of additive homogeneity, an important property of such systems. Lastly, several other hybrid system descriptions are stated.

2-1 Discrete-event (DE) systems

discrete-event (DE) systems form a large class of dynamic systems in which the evolution of the system is specified by the occurrence of certain discrete events [4]. This opposed to discrete-time (DT) systems where the evolution depends on the clock. Next to this, (part of) the states in a DE system represent time instances. There are multiple frameworks to describe DE systems, but these are usually nonlinear. However, a certain class of these systems can be described by a linear model in max-plus algebra, namely MPL systems. What max-plus algebra entails is discussed in the next section.

2-2 Max-plus algebra

Max-plus algebra is a relatively new form of algebra, where the main operations are maximization and addition. This as opposed to plus-times algebra where the main operations are addition and multiplication.

Define $\varepsilon = -\infty$ and $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$. The notation in max-plus algebra is the following:

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

for any x, $y \in \mathbb{R}_\varepsilon$.[5]

The reason why it is useful to rewrite a model into max-plus algebra, is that a lot of the basic operations that work for plus-times algebra also hold for max-plus algebra. Readers interested in these operations can take a look at Chapter 1.3 of [6] or Chapter 2.1 of [7]. However, there are also some major differences that prevent a straightforward translation of all properties, concepts, and algorithms from conventional linear algebra to max-plus algebra [8].

For matrix operations, max-plus addition and multiplication can be extended in the following way:

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

for A, $B \in \mathbb{R}_{\varepsilon}^{m \times n}$ and $C \in \mathbb{R}_{\varepsilon}^{n \times p}$.

2-3 Max-plus-linear (MPL) systems

A basic state-space system that can be constructed in max-plus algebra is the MPL system.

Definition 2-3.1 (Max-plus-linear system)**.**

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

with $A \in \mathbb{R}_{\varepsilon}^{n \times n}$, $B \in \mathbb{R}_{\varepsilon}^{n \times m}$ and $C \in \mathbb{R}_{\varepsilon}^{l \times n}$. Where n is the number of states, m is the number of inputs and l is the numbers of outputs.

This system is a subclass of DE systems in which only synchronization and no concurrency or choice occurs [8]. One can easily see the resemblance to the linear state space model in plus-times algebra. Thus this means that these systems can be handled in a similar way.

2-4 Max-Min-Plus-Scaling (MMPS) systems

An MMPS system is a generalization of a max-plus linear system. This generalization is useful, because a large portion of hybrid and DE systems can be described in this way. It can consist of the operations maximization, minimization, addition and multiplication.

First, define $\mathbb{R}_{\varepsilon} = \mathbb{R} \cup \{-\infty\}$, $\mathbb{R}_{\mathrm{T}} = \mathbb{R} \cup \{\infty\}$ and $\mathbb{R}_{c} = \mathbb{R} \cup \{-\infty\} \cup \{\infty\}$. The set $\mathcal{R}$ can be any of these three sets.

Definition 2-4.1 (MMPS function [6])**.**

$$f = p_i | \alpha | f_k \oplus f_l | f_k \oplus' f_l | f_k + f_l | \beta \dot{f}_k \tag{2-2}$$

where $\alpha \in \mathcal{R}$, $\beta \in \mathbb{R}$ and f_k and f_l are again MMPS functions over the set $\mathcal{R}$. The symbol $|$ means "or". For vector-valued MMPS functions the above statement holds componentwise.

Definition 2-4.2 (MMPS system [6])**.** Consider the vector:

$$p(k) = \begin{bmatrix} x^T(k) & x^T(k-1) & \dots & x^T(k-M) & u^T(k) & w^T(k) \end{bmatrix}^T \in \mathcal{P}$$

where $\mathcal{P} \subset \mathcal{R}^{n_p}$, $x \in \mathbb{R}_c^n$ is the state, $u \in \mathcal{R}^p$ is the control input and $w \in \mathcal{R}^z$ is an external signal. An MMPS system describes a state-space model of the form

$$x(k) = f_{\text{MMPS}}(p(k)) \tag{2-3}$$

where f_{MMPS} is a vector-valued MMPS function of the variables p.

This MMPS system describes an implicit state-space model. If the system is not a function of $x(k)$, the system is explicit.

2-4-1 Homogeneity and time-invariance

If a system's behaviour does not change over time, a system is said to be time-invariant. So if it is started a day later with the same initial state, the behaviour is the same [6]. In a DE system where the state is a time instance, time-invariance will mean that the system is additive homogeneous.

Definition 2-4.3 (Additive homogeneity [6])**.** Consider $p \in \mathbb{R}^{n_t}$, and the functions $f_t : \mathbb{R}^m \to \mathbb{R}^n$. The system is said to be additive homogeneous if:

$$f_t(p + \lambda) = f_t(p) + \lambda$$

A non-homogeneous system consists of both variables with the dimension time and variables related to quantities. Then the states can be split up into two substates. x_t denotes the time variables and x_q the quantitative variables. In a similar way $p(k)$ in Definition 2-4.2 can be split into $p_t(k)$ and $p_q(k)$. So: $x(k) = \begin{bmatrix} x_t(k) \\ x_q(k) \end{bmatrix}$ and $p(k) = \begin{bmatrix} p_t(k) \\ p_q(k) \end{bmatrix}$ with

$$p_t(k) = \begin{bmatrix} x_t^T(k) & x_t^T(k-1) & \dots & x_t^T(k-M) & u_t^T(k) & z_t^T(k) \end{bmatrix}^T \in \mathcal{P}_t$$
$$p_q(k) = \begin{bmatrix} x_q^T(k) & x_q^T(k-1) & \dots & x_q^T(k-M) & u_q^T(k) & z_q^T(k) \end{bmatrix}^T \in \mathcal{P}_q$$

Now the MMPS system can be rewritten as:

$$x_t(k) = f_{\text{MMPS,t}}(p_t(k), p_q(k))$$
$$x_q(k) = f_{\text{MMPS,q}}(p_t(k), p_q(k))$$

The quantitative variables prevent the system from being fully additive homogeneous. That is why the next definition is introduced.

Definition 2-4.4 (Partially additive homogeneity [6])**.** Consider $p_t \in \mathbb{R}^{n_t}$, and $p_q \in \mathbb{R}^{n_q}$ and the functions f_t :$\mathbb{R}^{n_t} \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_t}$ and $f_q : \mathbb{R}^{n_t} \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_q}$. The system is said to be partially additive homogeneous if:

$$\begin{bmatrix} f_t(p_t + \lambda, p_q) \\ f_q(p_t + \lambda, p_q) \end{bmatrix} = \begin{bmatrix} f_t(p_t, p_q) + \lambda \\ f_q(p_t, p_q) \end{bmatrix}$$

This means that a time-invariant DE MMPS system is equal to a (partially) additive homogeneous DE MMPS system.

2-5 Hybrid systems

In this section multiple hybrid systems are outlined that are equal to MMPS systems (sometimes under mild conditions). Here the event counter k is used, instead of the DT time increment t.

2-5-1 Conjunctive MMPS system

The conjunctive MMPS formulation is a canonical formulation of the general MMPS system.

Definition 2-5.1 (Conjunctive MMPS system [6]). *A conjunctive MMPS system describes a state-space model of the form*

$$x(k) = \min_{i=1,\ldots,K} \max_{j=1,\ldots,n_i} (\alpha_{i,j}^T p(k) + \beta_{i,j}) \tag{2-4}$$

for some integers K,$n_1, \ldots, n_K$, *vectors* $\alpha_{i,j}$ *and real numbers* $\beta_{i,j}$. *For vector-valued MMPS functions the above statements hold componentwise.*

This expression is also called the Min-max MMPS formulation.

Theorem 2-5.1 (MMPS to conjunctive MMPS [6]). *The classes of MMPS systems (Equation 2-3) and conjunctive MMPS systems (Equation 2-4) coincide.*

2-5-2 Extended linear complementarity (ELC) Systems

An Extended linear complementarity (ELC) system is an extension of a linear complementarity system.

Definition 2-5.2 (Extended linear complementarity system [9]). The ELC system can be described as:

$$\begin{aligned}
&x(k+1) = Ax(k) + B_1 u(k) + B_2 d(k) \\
&y(k) = Cx(k) + D_1 u(k) + D_2 d(k) \\
&\prod_{j\in\phi_i} (g_4 - E_1 x(k) - E_2 u(k) - E_3 d(k))_j = 0 \text{ for each } i \in \{1, 2, \ldots, p\}
\end{aligned} \tag{2-5}$$

where $u(k) \in \mathbb{R}^m$, $x(k) \in \mathbb{R}^n$ and $y(k) \in \mathbb{R}^l$. g_4 is a constant and $d(k) \in \mathbb{R}^r$ is an auxiliary variable.

Theorem 2-5.2 (MMPS to ELC [9]). *The classes of MMPS and ELC systems coincide.*

2-5-3 Mixed logical dynamical (MLD) Systems

An Mixed logical dynamical (MLD) system is a hybrid system where logic, dynamics and constraints are integrated. [9]

Definition 2-5.3 (Mixed logical dynamical system [9])**.** The MLD system can be described as:

$$\begin{aligned} x(k+1) &= Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \\ y(k) &= Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \\ E_1x(k) + E_2u(k) &+ E_3\delta(k) + E_4z(k) \leq g_5 \end{aligned} \tag{2-6}$$

where $x(k) = \begin{bmatrix} x_r^T(k) & x_b^T(k) \end{bmatrix}^T$ with $x_r(k) \in \mathbb{R}^{n_r}$ and $x_b(k) \in \{0,1\}^{n_b}$ ($y(k)$ and $u(k)$ have a similar structure), and where $z(k) \in \mathbb{R}^{r_r}$ and $\delta(k) \in \{0,1\}^{r_b}$ are auxiliary variables.

Theorem 2-5.3 (ELC to MLD [9])**.** *Every ELC system can be written as an MLD system, provided that the quantity $g_4 - E_1x(k) - E_2u(k) - E_3d(k)$ is (componentwise) bounded.*

So combining Theorem 2-5.2 and Theorem 2-5.3, an MMPS system can be written as an MLD system if the condition in Theorem 2-5.3 and is met. To see what the consequences of this condition are for the original MMPS model, one has to take a look at how the constraints for the ELC model are formed from the MMPS model.

According to [9], the constraints for the ELC model are formed when rewriting the maximizations and minimizations in the MMPS model. $f = \max(f_k, f_l) = -\min(-f_k, -f_l)$ can be rewritten as:

$$f - f_k \geq 0, \quad f - f_l \geq 0, \quad (f - f_k)(f - f_l) = 0$$

So to be able to recast an (conjunctive) MMPS model as an MLD model, it is necessary that the quantities $f - f_k$ and $f - f_l$ are bounded for each maximization and minimization.

2-5-4 Piecewise-affine (PWA) Systems

A Piecewise-affine (PWA) system is a system that consists of a combination of several affine systems that are valid within a certain part of the domain.

Definition 2-5.4 (PWA System [9])**.** PWA systems are described by

$$\begin{aligned} x(k+1) &= A_ix(k) + B_iu(k) + f_i \\ y(k) &= C_ix(k) + D_iu(k) + g_i \\ \text{for} & \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i \end{aligned} \tag{2-7}$$

where Ω_i are convex polyhedra (i.e. given by a finite number of linear inequalities) in the input/state space.

Theorem 2-5.4 (MLD to PWA [9])**.** *A completely well-posed MLD system can be rewritten as a PWA system.*

So combining Theorem 2-5.2, 2-5.3 and 2-5.4, an MMPS system can be written as a PWA system if the conditions in Theorem 2-5.3 and 2-5.4 are met. So the MMPS problem should be bounded and well-posed. As a result, a continuous PWA system is equivalent to an MMPS system. [10]

2-6 Conclusion

Chapter 2 gives an overview of the basic aspects of max-plus algebra. The first takeaway in this chapter is that in DE systems, the evolution of a system is specified relative to the (repeated) occurrence of events. Secondly, several operations in plus-times algebra can be translated to max-plus algebra, which is why writing a model as an MPL system can be useful. In that way (simple) systems that consists of maximizations as well as addition can still be handled in a "linear" way.

Furthermore, the concept of max-plus algebra has been extended to MMPS systems. This is helpful, because a large amount of DE systems can be described in this way. In non-homogeneous MMPS systems, the system consists of time and quantitative variables. Here the difference between additive homogeneity and time-invariance comes into play. It can be concluded that an MMPS system being time-invariant actually comes down to the system being (partially) additive homogeneous. In Chapter 4 it will become apparent that the difference between additive homogeneity and time-invariance will prevent a straightforward translation of Model predictive control (MPC) from DT to DE systems.

Finally, it can be concluded that several other hybrid systems are (sometimes under mild conditions) equivalent to MMPS systems. This will prove useful because there are not many optimization techniques specifically for MMPS systems. Rewriting the MMPS system into one of the hybrid systems mentioned in this chapter, will pave the way for the use of more conventional and efficient optimization techniques. The conjunctive MMPS formulation can be a useful first step in the recasting process.

Chapter 3

Model Predictive Control

Model predictive control (MPC) is an on-line control strategy which makes use of a receding horizon N_p. It is a popular control method for discrete-time (DT) systems. One of the reasons for this is that it is simple to add constraints in the optimization, as opposed to for example Linear-quadratic regulator (LQR) control. This chapter introduces the MPC control technique and its procedure for controlling nonlinear DT systems. This will provide a solid background and show where alterations are necessary to be able to use MPC on (nonlinear) discrete-event (DE) Max-Min-Plus-Scaling (MMPS) systems as well.

3-1 Introduction to Model predictive control (MPC)

An MPC controller is a controller that predicts, at each time step t, the optimal control inputs over the finite horizon N_p: u_t,u_{t+1},...,u_{t+N_p} . It only applies the first input u_t to the system and shifts the horizon one time step, such that it now runs from $t+1$ to $t+N_p+1$. Then a new optimal control problem is solved to determine these inputs. This is continued for each cycle, such that the future control actions are optimized by minimizing the cost function over prediction window N_p subject to constraints [1]. This is outlined in Figure 3-1.

The cost function in MPC is typically chosen as [11]

$$J(x_0, \mathbf{u}) = V_N(x_0, \mathbf{u}) = \sum_{t=0}^{N_p-1} \ell(x(t), u(t)) + V_f(x(N_p)) \tag{3-1}$$

where $\ell(x, u)$ is the stage cost and $V_f(x)$ is the terminal penalty.

Here it is assumed that the state and input vector should be steered to zero. If this is not the case, the vectors should be altered by subtracting their equilibrium values: $x' = x - x_{eq}$, $u' = u - u_{eq}$. Then these altered vectors x' and u' should again be steered to zero.

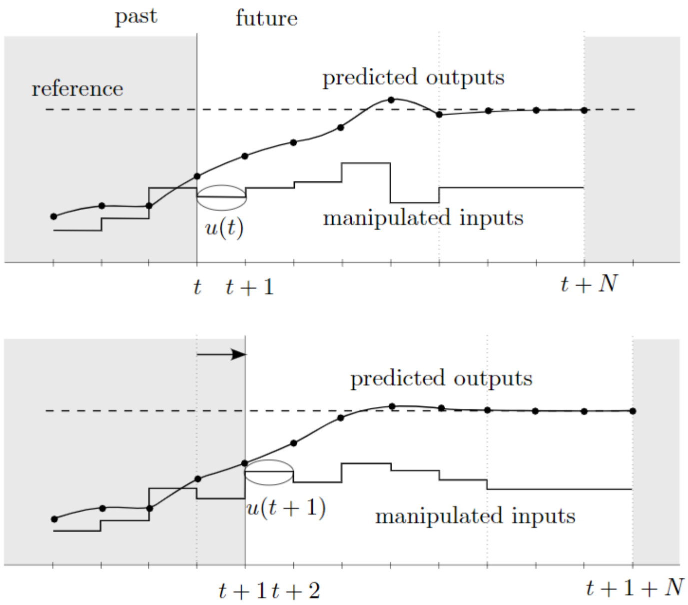


Figure 3-1: Graphical representation of the MPC control strategy [12]

The cost function has to be defined in such a way that stability is established. This means that

$$V_N^0(f(x, \kappa_N(x)) - V_N^0(x) \leq -\ell(x, \kappa_N(x)) \tag{3-2}$$

where $\kappa_N = K_N x$ is the control law at stage N [13] and $V_N^0(x)$ is the MPC optimal value function.

Because a DE MMPS system is a nonlinear system, the next section will focus on how to guarantee stability for a nonlinear DT system.

3-2 MPC on nonlinear DT systems

The MPC problem for nonlinear DT systems is given in Definition 3-2.1. This section will use the DT time increment t.

Definition 3-2.1 (MPC problem for nonlinear DT systems)**.**

$$\min_{u(0),\ldots,u(N_p-1)} J(x_0, \mathbf{u}) = \min_{u(0),\ldots,u(N_p-1)} \sum_{t=0}^{N_p-1} \ell(x(t), u(t)) + V_f(x(N_p)) \tag{3-3}$$

subject to:

$$\begin{aligned}
&x(0) = x_0 \\
&x(t+1) = f(x(t), u(t)) && \forall t \\
&(x(t), u(t)) \in \mathbb{Z} && \forall t \\
&u(t+j) = u(t+N_c-1) && j = N_c, N_c+1, \ldots, N_p-1 \\
&x(t+N_p) \in \mathbb{X}_f \subset \mathbb{X}
\end{aligned} \tag{3-4}$$

where $J(x_0, \mathbf{u})$ is the cost function, $\ell(x, u)$ is the stage cost, $V_f(x)$ is the terminal cost, $\mathbb{X}_f$ is the terminal constraint set and $\mathbb{X}$ is the state constraint set.

The cost function that will be used in this section is a quadratic (or 2 norm) cost function. This means

$$\begin{aligned} \ell(x,u) &= \frac{1}{2}x^T Q x + \frac{1}{2}u^T R u \\ V_f(x) &= \frac{1}{2}x^T P x \end{aligned} \tag{3-5}$$

The 2 norm cost function is a popular choice in conventional MPC. In this cost function larger deviations from the reference value will be more severely punished than small deviations. Other common choices for the cost function are a 1 norm or ∞ norm cost function.

The procedure in this section is based on the process in Chapter 2.5.5 of [11]. First, several assumptions that need to hold for the procedure to be valid will be mentioned. Subsection 3-2-1 outlines these assumptions. It also introduces two theorems that are needed later on. Before these are given, the definition for a Lyapunov function is given, that will be used later on.

Definition 3-2.2 (Lyapunov function)**.** Suppose that $\mathbb{X}$ is positive invariant for $x^+ = f(x)$. A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is said to be a Lyapunov function in $\mathbb{X}$ for $x^+ = f(x)$ if there exist functions α_1, $\alpha_2 \in \mathcal{K}_\infty$ and a continuous, positive definite function α_3 such that for any $x \in \mathbb{X}$

$$\begin{aligned} V(x) &\geq \alpha_1(|x|) \\ V(x) &\leq \alpha_2(|x|) \\ V(f(x)) - V(x) &\leq -\alpha_3(|x|) \end{aligned}$$

3-2-1 Assumptions and Theorems

$\mathbb{X}$ is the state constraint set, $\mathbb{U}$ is the input constraint set and $\mathbb{X}_f$ is the terminal constraint set. $\mathbb{Z}$ is the system constraint region given by $\mathbb{Z} \subseteq \mathbb{X} \times \mathbb{U}$. This is generally a polyhedron, i.e. $\mathbb{Z} = \{(x,u) \mid Fx + Eu \leq e\}$ for some F, E and e.

Assumption 3-2.1 (Continuity of system and cost [11])**.** *The function $f : \mathbb{Z} \to \mathbb{X}$, $\ell : \mathbb{Z} \to \mathbb{R}_{\geq 0}$ and $V_f : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$ are continuous, $f(x_{eq}, u_{eq}) = x_{eq}$, $\ell(x_{eq}, u_{eq}) = 0$ and $V_f(x_{eq}) = 0$.*

Assumption 3-2.2 (Properties of constraint sets [11])**.** *The set $\mathbb{Z}$ is closed and the set $\mathbb{X}_f \subseteq \mathbb{X}$ is compact. Each set contains the origin (= equilibrium). If $\mathbb{U}$ is bounded (hence compact), the set $\mathbb{U}(x)$ is compact for all $x \in \mathbb{X}$. If $\mathbb{U}$ is unbounded, the function $\mathbf{u} \mapsto V_N(x, \mathbf{u})$ is coercive, i.e., $V_N(x, \mathbf{u}) \to \infty$ as $|\mathbf{u}| \to \infty$ for all $x \in \mathbb{X}$.*

Assumption 3-2.3 (Basic stability assumption [11])**.** *$V_f(\cdot)$, $\mathbb{X}_f$ and $\ell(\cdot)$ have the following properties:*

- *For all $x \in \mathbb{X}_f$, there exists a u (such that $(x,u) \in \mathbb{Z}$) satisfying*

$$\begin{aligned} &f(x,u) \in \mathbb{X}_f \\ &V_f(f(x,u)) - V_f(x) \leq -\ell(x,u) \end{aligned}$$

- *There exist $\mathcal{K}_\infty$ functions $\alpha_1(\cdot)$ and $\alpha_f(\cdot)$ satisfying*

$$\begin{aligned} &\ell(x,u) \geq \alpha_1(|x|) \quad \forall x \in \mathcal{X}_N, \forall u \text{ such that } (x,u) \in \mathbb{Z} \\ &V_f(x) \leq \alpha_f(|x|) \quad \forall x \in \mathbb{X}_f \end{aligned}$$

where $\mathcal{X}_N$ is the set of feasible states for the optimal control problem at stage N. A function belongs to class $\mathcal{K}_\infty$ if it is continuous, zero at zero, strictly increasing and unbounded [11].

Theorem 3-2.1 (Asymptotic stability of the equilibrium)**.** *Suppose Assumptions 3-2.1, 3-2.2 and 3-2.3 are satisfied and $\mathbb{X}_f$ contains the equilibrium in its interior. Then*

- *There exists $\mathcal{K}_\infty$ functions $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ such that for all $x \in \mathcal{X}_N$ ($\bar{\mathcal{X}}_N^c$, for each $c \in \mathbb{R}_{>0}$)*

$$\begin{aligned} \alpha_1(|x|) \leq & V_N^0(x) \leq \alpha_2(|x|) \\ V_N^0(f(x,\kappa_N(x))) - & V_N^0(x) \leq -\alpha_1(|x|) \end{aligned}$$

- *The origin is asymptotically stable in $\mathcal{X}_N$ ($\bar{\mathcal{X}}_N^c$, for each $c \in \mathbb{R}_{>0}$) for $x(k) = f(x(k-1), \kappa_N(x))$.*

Theorem 3-2.2 (Lyapunov function and exponential stability)**.** *Suppose $\mathbb{X} \subset \mathbb{R}^n$ is positive invariant for $x(k) = f(x(k-1))$. If there exists a Lyapunov function in $\mathbb{X}$ for the system $x(k) = f(x(k-1))$ with $\alpha_i(\cdot) = c_i|\cdot|^a$ in which a, $c_i \in \mathbb{R}_{>0}$ $i = 1, 2, 3$ then the equilibrium is exponentially stable for $x(k) = f(x(k-1))$ in $\mathbb{X}$.*

3-2-2 The Procedure

The first step in solving the MPC problem for nonlinear DT systems (Definition 3-2.1) is determining the terminal cost V_f and terminal constraint set $\mathbb{X}_f$. These should ensure asymptotic stability of the origin for the controlled problem [11]. The first step is linearizing the nonlinear system $x(t+1) = f(x(t), u(t))$ around its equilibrium. To be able to do this Assumption 3-2.1 needs to hold.

The linearization results in the following system:

$$\begin{aligned} &x(t+1) = Ax(t) + Bu(t) \\ &\text{where } A = f_x(x_{eq}, u_{eq}),\ B = f_u(x_{eq}, u_{eq}) \end{aligned}$$

Assuming (A, B) is stabilizable, the controller gain K ($u = Kx$) should be chosen such that the equilibrium is globally exponentially stable for the system $x(t+1) = A_K x(t)$, $A_K = A + BK$. This means that A_K will be stable. Now the stage cost in Equation 3-5 for $u = Kx$ is given as $\ell(x, Kx) = \frac{1}{2}x^T Q_K x$, $Q_K = Q + K^T RK$.

To determine V_f, we should first define P by the Lyapunov function

$$A_K^T P A_K + \mu Q_K = P \tag{3-6}$$

for some $\mu > 1$. If we make sure Q and R are positive definite, so is Q_K. Next to that, A_K is stable and thus P is positive definite as well. Now $V_f = \frac{1}{2}x^T Px$ is a global Control Lyapunov function (CLF) for the linearized system, since:

$$V_f(A_K x) + \frac{\mu}{2}x^T Q_K x - V_f(x) = 0 \tag{3-7}$$

If we now consider the nonlinear system $x(t+1) = f(x(t), u(t))$ again, with linear control $u = Kx$. The goal is to show that V_f is a local CLF for $x(t+1) = f(x(t), u(t))$ in some neighbourhood of the equilibrium. This means that we need to show that there exists an $a \in (0, \infty)$ such that

$$V_f(f(x, Kx)) + \frac{1}{2}x^T Q_K x - V_f(x) \leq 0 \qquad \forall x \in lev_a\, V_f \tag{3-8}$$

in which, for all $a > 0$, $lev_a\, V_f = \{x | V_f(x) \leq a\}$ is a sublevel set of V_f.

Comparing Equation 3-8 and 3-7 shows that Equation 3-8 holds when:

$$V_f(f(x, Kx)) - V_f(A_K x) \leq \frac{\mu - 1}{2}x^T Q_K x \qquad \forall x \in lev_a\, V_f \tag{3-9}$$

This also demonstrates the reason for including $\mu > 1$. It allows for a larger difference in $V_f(f(x, Kx))$ and $V_f(A_K x)$, while still proving that Equation 3-8 holds.

To prove that Equation 3-9 holds, let $e(\cdot) = f(x, Kx) - A_K x$. Then

$$V_f(f(x, Kx)) - V_f(A_K x) = (A_K x)^T Pe(x) + \frac{1}{2}e(x)^T Pe(x) \tag{3-10}$$

It holds that $e(0) = f(0, K \cdot 0) - A_K \cdot 0 = 0$ and $e_x(x) = f_x(x, Kx) + f_u(x, Kx)K - A_K$. So $e_x(0) = 0$. If $f(\cdot)$ is twice continuously differentiable, for any $\delta > 0$, there exists a $c_\delta > 0$ such that $|e_{xx}(x)| \leq c_\delta$ for all x in $\delta\mathcal{B}$. Here $\mathcal{B}$ is a ball in $\mathbb{R}^n$ of unit radius. That $f(\cdot)$ is twice continuously differentiable also means that [11]

$$\begin{aligned} |e(x)| &= |e(0) + e_x(0)x + \int_0^1 (1-s)x^T e_{xx}(sx)x ds| \\ &\leq \int_0^1 (1-s)c_\delta |x|^2 ds \leq \frac{1}{2}c_\delta |x|^2 \end{aligned} \tag{3-11}$$

for all $x \in \delta\mathcal{B}$. From Equation 3-10, one can observe that there exists an $\varepsilon \in (0, \delta]$ such that Equation 3-9 holds and thus Equation 3-8 is satisfied as well for all $x \in \varepsilon\mathcal{B}$ [11]. Because of choosing $\ell(\cdot)$ as in Equation 3-5, there exists a $c_1 > 0$ such that $V_f(x) \geq \ell(x, Kx) \geq c_1|x|^2$ for all $x \in \mathbb{R}^n$. From this it follows that $x \in lev_a\, V_f$ implies $|x| \leq \sqrt{\frac{a}{c_1}}$. By making a satisfy $\sqrt{\frac{a}{c_1}} = \varepsilon$, $x \in lev_a\, V_f$ implies $|x| \leq \varepsilon \leq \delta$. This results in Equation 3-8 being satisfied.

So indeed, there exists an $a > 0$ such that $V_f(\cdot)$ and $\mathbb{X}_f = lev_a V_f$ satisfy Assumptions 3-2.1 and 3-2.2. To make sure Assumption 3-2.3 is satisfied as well and the prerequisites of Theorem 3-2.1 hold, the first step is to see that for each $x \in \mathbb{X}_f$ there exists a $u = \kappa_f(x) = Kx$ such that $V_f(x, u) \leq V_f(x) - \ell(x, u)$, since $\ell(x, Kx) = (1/2)x^T Q_K x$. We stated that in $\ell(x, u) = \frac{1}{2}(x^T Qx + u^T Ru)$ both Q and R are positive definite. Combining this with our definition of $V_f(\cdot)$ will ensure the existence of positive constants c_1, c_2 and c_3, such that $V_N^0(x) \geq c_1|x|^2$ for all $\mathbb{R}^n$, $V_f(x) \leq c_2|x|^2$ and $V_N^0(f(x, \kappa_f(x))) \leq V_N^0(x) - c_3|x|^2$ for all $x \in \mathbb{X}_f$. Thus Assumption 3-2.3 is satisfied. Finally, by definition, the set $\mathbb{X}_f$ contains the origin (= equilibrium) in its interior.

So in summary: if Assumptions 3-2.1, 3-2.2 and 3-2.3 are satisfied, $\mathbb{X}_f$ contains the origin (= equilibrium) in its interior and $\alpha_1(\cdot)$, $\alpha_2(\cdot)$ and $\alpha_3(\cdot)$ satisfy the hypotheses of Theorem 3-2.2, the origin (its equilibrium) is exponentially stable for the system $x(t+1) = f(x(t), \kappa_N(x(t)))$ in $\mathcal{X}_N$ by Theorems 3-2.1 and 3-2.2 [11].

3-3 Conclusion

This chapter gives an overview of the process of applying MPC on a nonlinear DT system. In MPC a certain cost function needs to be minimized. This chapter shows how the cost function should be chosen such that stability for the nonlinear DT system is ensured. This can be used as a basis to describe this process for a (nonlinear) DE MMPS system. There are several concerns that arise when trying to apply the DT procedure to a (non homogeneous) DE system.

Firstly, the equilibrium point for time variables (which are present in a DE system) are nonzero and also not constant, because this would mean that in the equilibrium all events happen at the same time. This means that for time variables x_t and u_t it is not possible to subtract a constant equilibrium value ($x' = x - x_{eq}$, $u' = u - u_{eq}$) and steer the altered vectors to zero.

Next to that, there are multiple concerns with the assumptions that are made in Section 3-2-1. In Assumption 3-2.2, it is stated that the set $\mathbb{Z}$ is closed. However, for a DE system $\mathbb{Z}$ depends on the initial conditions. This is the case, because time variables are additive homogeneous (Definition 2-4.3). So if a system is started at a later time, the state constraint set and system constraint region are shifted by that same amount. Next to that, the location of the set depends on the event counter k, because the value of x_t grows as the event counter increases. Thus these regions can only be closed for a certain initial condition and a certain value of k specifically. A same reasoning holds for the terminal set $\mathbb{X}_f$. It is not compact, due to additive homogeneity. It is only shift invariant. This means that it is positive invariant for a certain initial condition of a time variable, but the set shifts when these initial conditions shift and when the event counter k increases. So it needs to be checked whether these differences will lead to problems.

Another concern is that the procedure mentions that the system needs to be linearized around the equilibrium point. Calculating the derivative for a MMPS system is not straightforward, as it is hard to differentiate a system that consists of the operations maximization and minimization. This same problem comes into play again, when the procedure mentions that the system should be twice continuously differentiable for Equation 3-11 to hold.

All these concerns will be addressed in Chapter 4, to make sure that it is possible to perform MPC on a general (partially) additive homogeneous DE system.

Chapter 4

Stability of the Discrete-event MPC Problem

The process to assure stability in a Model predictive control (MPC) problem on a general discrete-event (DE) Max-Min-Plus-Scaling (MMPS) system, is not entirely the same as assuring stability in a discrete-time (DT) system. In the conclusion of Chapter 3, multiple concerns were raised. This chapter will discuss these concerns and solve these problems if needed.

4-1 Altering the MPC procedure

The goal in MPC is to stabilize the system to an equilibrium point. When translating the MPC control method to DE systems, a problem arises here. Time variables $x_t(k)$ in a DE system cannot stay constant after a certain event, because then all those events will happen at the same time. This is why the aim for these variables should be to keep the growth rate constant. So $x_t(k) = x_t(k-1) + constant$ or $x_t(k) - x_t(k-1) = x_t(k-1) - x_t(k-2)$.

This means that to fit the original MPC framework, the state matrix should be altered. It should use the growth rate of the time variables, rather than the time variables $x_t(k)$ themselves, and the normal quantitative variables $x_q(k)$. So the new vector would look like:

$$x_{new}(k) = \begin{bmatrix} x_t(k) - x_t(k-1) \\ x_q(k) \end{bmatrix}$$

Now this new state vector can be steered to a constant equilibrium.

However, for a general MMPS system it might not be possible to describe the growth rate of x_t and original states x_q (if the system is non-homogeneous) without the original x_t states. This means that the state vector needs to contain both $x_t(k) - x_t(k-1)$ and x_t. So then:

$$x_{new}(k) = \begin{bmatrix} x_t(k) - x_t(k-1) \\ x_t(k) \\ x_q(k) \end{bmatrix}$$

To be able to perform MPC on the x_t variable(s), we should use an equilibrium point where the values for the time variables vary, based on the event k the system is in. Now the state vector should stay close to the varying equilibrium point $x_{eq}(k)$.

Because $x_{eq}(k)$ is not a constant value, it cannot be easily subtracted from the state vector beforehand. This is why it makes more sense to alter the formulation of the stage cost $\ell(x,u)$ and terminal cost $V_f(x)$, such that $x_{eq}(k)$ is part of the description. So for an MMPS system the cost function will have the following form:

Definition 4-1.1 (Cost function DE MMPS system)**.**

$$J(x_0,\mathbf{u}) = V_N(x_0,\mathbf{u}) = \sum_{i=0}^{N_p-1} \ell(x(k-1+i),u(k+i)) + V_f(x(k+N_p-1)) \tag{4-1}$$

where

$$\ell(x,u) = \frac{1}{2}(x(k)-x_{eq}(k))^T Q(x(k)-x_{eq}(k)) + \frac{1}{2}(u(k)-u_{eq}(k))^T R(u(k)-u_{eq}(k))$$
$$V_f(x) = \frac{1}{2}(x(k)-x_{eq}(k))^T P(x(k)-x_{eq}(k))$$

4-2 System constraint region

The system constraint region $\mathbb{Z}$ is computed as $\mathbb{Z} \subseteq \mathbb{X} \times \mathbb{U}$. As (part of) the state vector is additive homogeneous, the state constraint set will be as well. This means that the system constraint region is (partially) additive homogeneous too. In Assumption 3-2.1, $\mathbb{Z}$ is assumed to be closed. Clearly, this is not the case for the MMPS case, since the region can be shifted by an infinite amount. However, for a certain initial condition and a certain event counter k, $\mathbb{Z}$ is bounded. So the additive homogeneity of the system constraint region will not be a problem in the optimization. It just does not always lay at the same location.

4-3 Terminal constraint set

In the same way as the system constraint region, the terminal constraint set is (partially) additive homogeneous, because it depends on the initial condition of the time variables $x_t(k)$. So for the time variables, when the system is started at a later time, all $x_{eq,t}$ values will be shifted by a similar amount. The equilibria for $x_q(k)$ and $x_t(k) - x_t(k-1)$ will stay at the same location. As the previous section already pointed out, this shift based on $x_t(0)$ will not be a problem, because the initial conditions are known at the beginning of the optimization.

Another concern to take a look at is that the terminal set $\mathbb{X}_f$ for an MMPS system depends on the event counter k. In Section 3-2, the terminal set is described as a sub-level set of the terminal cost V_f ($\mathbb{X}_f = lev_a\, V_f$). So it has the shape of an ellipsoid with x_{eq} at its center. Now that the equilibrium point for the time variables $x_{eq,t}$ varies with event counter k, the terminal constraint set varies with k as well. So actually $\mathbb{X}_f(k)$. Next to that, both $x_t(k)$ and $x_t(k) - x_t(k-1)$ can only be positive. So the terminal set will only consist of the positive values of the sub-level set in the direction of $x_t(k)$ and $x_t(k) - x_t(k-1)$. For a certain $x(0)$ and k, this terminal set $\mathbb{X}_f(k)$ will be closed and bounded. So it is a compact set, which is needed for Assumption 3-2.2 to hold.

4-4 Linearizing the system

In order to determine the terminal cost V_f and the terminal constraint set $\mathbb{X}_f$, a linearization of the MMPS system around the equilibrium point is needed. Because the system contains the operations minimization and maximization, this is not straightforward to acquire.

The equilibrium point can be computed based on the original MMPS formulation together with the conditions that should hold at the equilibrium: $x_t(k)-x_t(k-1) = x_t(k-1)-x_t(k-2)$ and $x_q(k) = x_q(k-1)$. Then the MMPS system should be written into the conjunctive canonical format. According to Theorem 2-5.1, this is always possible. The calculated equilibrium value can be used to figure out which part of the conjunctive formulation is responsible for the value of $x(k)$ when the system is at the equilibrium. So it needs to be determined which part of the conjunctive formulation "wins" at the equilibrium.

To acquire the linearization of the MMPS system at the equilibrium, one now just needs to linearize this "winning" part of the conjunctive formulation. This is straightforward, because the "winning" part only consists of the operations addition and scaling. Now that the linearization is acquired, the state feedback gain K and the terminal cost matrix P can be determined in the same way as in Section 3-2-2.

4-5 Conclusion

This chapter addressed the concerns that had been raised in the conclusion of Chapter 3. To fix the issue that the equilibrium is not constant, the MPC procedure was altered. $x_t(k) - x_t(k-1)$ was added to the state vector. As it is generally not possible to describe the states without the original state x_t, x_t was also kept a part of the new state vector. Next to that, the equilibrium values are included in the description for the cost function, since subtracting them in the beginning is not as straightforward as in the DT case.

Dealing with the additive homogeneity property of the time variables comes down to looking at the sets in a different way. They will not always have the same value. However, if the initial values are known the sets can be computed. Dealing with a dependency on the event counter k needs some extra care. A set will move as the horizon moves, so its value will shift during the optimization. One needs to make sure that at each optimization the right set is taken into account.

The concerns on how to linearize an MMPS system, was solved by looking at the conjunctive formulation and finding out which part of this formulation is responsible for the value of the state at the equilibrium. From here, the linearization is easy to acquire.

As all concerns raised in Chapter 3 have been addressed and a solution was proposed if needed, the MPC procedure from Chapter 3 can now be applied to (non-homogenous) DE MMPS systems. Chapter 6 will investigate an example of such a system.

Chapter 5

Control of the MMPS System

This chapter demonstrates how a Max-Min-Plus-Scaling (MMPS) system should be rewritten such that it can be controlled efficiently with Model predictive control (MPC). Firstly, the model will be rewritten as a Mixed logical dynamical (MLD) system. Combined with a cost function this can be recast as a Mixed integer quadratic programming (MIQP) problem.

This chapter uses the altered state vector

$$x(k) = \begin{bmatrix} x_t(k) - x_t(k-1) \\ x_t(k) \\ x_q(k) \end{bmatrix}$$

that was proposed in Section 4-1.

5-1 Rewriting the MMPS system as an MLD system

To be able to apply MPC on an MMPS system in an efficient way, the problem needs to be written into another format. This section describes the first step to achieve this: recasting the MMPS system as an MLD system. It follows a similar procedure as in [14].

According to Theorem 2-5.1, all MMPS systems can be rewritten into the conjunctive MMPS format.

$$x(k) = \min_{i=1,\ldots,K} \max_{j=1,\ldots,n_i} (\alpha_{i,j}^T p(k) + \beta_{i,j}) \tag{5-1}$$

To be able to rewrite this conjunctive MMPS model as an MLD model the conditions at the end of Section 2-5-3 should hold. What these conditions entail for a conjunctive MMPS system, is summarized in Assumption 5-1.1.

Assumption 5-1.1 (Bounded conjunctive MMPS system)**.** *Define:*

$$g_i = \max_{j=1,\dots,n_i}(\alpha_{i,j}^T p(k) + \beta_{i,j}) = \max_{j=1,\dots,n_i} h_{i,j} \quad \textit{for } i = 1,\dots,K$$
$$f = \min_{i=1,\dots,K} g_i$$

Then it holds that:

- $g_i - h_{i,j}$ *is bounded for* $i = 1,\dots,K$ *and* $j = 1,\dots,n_i$.
- $g_i - f$ *is bounded for* $i = 1,\dots,K$.

To describe the procedure on how to rewrite a conjunctive MMPS system as an MLD system, we will first take a look at a one-dimensional example system:

$$x(k) = \min\Big(\max(\underbrace{\alpha_{1,1}^T p(k) + \beta_{1,1}}_{h_{11}}, \underbrace{\alpha_{1,2}^T p(k) + \beta_{1,2}}_{h_{12}}, \underbrace{\alpha_{1,3}^T p(k) + \beta_{1,3}}_{h_{13}}, \underbrace{\alpha_{1,4}^T p(k) + \beta_{1,4}}_{h_{14}}),$$
$$\max(\underbrace{\alpha_{2,1}^T p(k) + \beta_{2,1}}_{h_{21}}, \underbrace{\alpha_{2,2}^T p(k) + \beta_{2,2}}_{h_{22}}, \underbrace{\alpha_{2,3}^T p(k) + \beta_{2,3}}_{h_{23}})\Big)$$

For each extra term in the maximizations and minimization of the example, a variable needs to be defined. With $l_{10} = h_{11}$ and $l_{20} = h_{21}$, it can be defined that:

$$l_{11} = \max(l_{10}, h_{12})$$
$$l_{12} = \max(l_{11}, h_{13})$$
$$l_{13} = \max(l_{12}, h_{14})$$
$$l_{21} = \max(l_{20}, h_{22})$$
$$l_{22} = \max(l_{21}, h_{23})$$
$$x(k) = \min(l_{13}, l_{22})$$

With these definitions, the MMPS model can be written as an MLD model. For the first maximization it holds that:

$$o_{11} = h_{12} - l_{10}$$
$$\delta_{11} = \begin{cases} 1 & \text{if } o_{11} \geq 0 \\ 0 & \text{if } o_{11} < 0 \end{cases}$$
then $l_{11} = l_{10} + (h_{12} - h_{10}) \cdot \delta_{11} = l_{11} + o_{11} \cdot \delta_{11} = h_{11} + z_{11}$

$$o_{12} = h_{13} - l_{11}$$
$$\delta_{12} = \begin{cases} 1 & \text{if } o_{12} \geq 0 \\ 0 & \text{if } o_{12} < 0 \end{cases}$$
then $l_{12} = l_{11} + (h_{13} - l_{11}) \cdot \delta_{12} = l_{11} + o_{12} \cdot \delta_{12} = l_{11} + z_{12} = h_{11} + z_{11} + z_{12}$

$$o_{13} = h_{14} - l_{12}$$
$$\delta_{13} = \begin{cases} 1 & \text{if } o_{13} \geq 0 \\ 0 & \text{if } o_{13} < 0 \end{cases}$$
then $l_{13} = l_{12} + (h_{14} - l_{12}) \cdot \delta_{13} = l_{12} + o_{13} \cdot \delta_{13} = l_{12} + z_{13} = h_{11} + z_{11} + z_{12} + z_{13}$

The second maximization can be transformed in a similar way. The minimization can be represented as:

$$
\begin{aligned}
&o_1 = l_{22} - l_{12}\\
&\delta_1 = \begin{cases} 1 & \text{if } o_1 \leq 0 \\ 0 & \text{if } o_1 > 0 \end{cases}\\
&\text{then } x(k) = l_{12} + (l_{22} - l_{12}) \cdot \delta_1 = l_{12} + o_1 \cdot \delta_1 = l_{12} + z_1 = h_{11} + z_{11} + z_{12} + z_{13} + z_1
\end{aligned}
$$

Now the δ constraints should be altered to fit the MLD format. First, we define $m_{ij} = \min o_{ij}$ and $M_{ij} = \max o_{ij}$. In the same way $m_i = \min o_i$ and $M_i = \max o_i$. Now the maximization δ constraint

$$[o_{ij}(k) \geq 0] \Leftrightarrow [\delta_{ij} = 1]$$

can be represented by

$$\begin{cases} o_{ij}(k) \geq \varepsilon + (m_{ij} - \varepsilon)(1 - \delta_{ij}) \\ o_{ij}(k) \leq M_{ij}\delta_{ij} \end{cases}$$

and the minimization δ constraint

$$[o_i(k) \leq 0] \Leftrightarrow [\delta_i = 1]$$

can be represented by

$$\begin{cases} o_i(k) \geq \varepsilon + (m_i - \varepsilon)\delta_i \\ o_i(k) \leq M_i(1 - \delta_i) \end{cases}$$

The variables $z_{ij} = \delta_{ij} \cdot o_{ij}$, can be expressed as:

$$
\begin{aligned}
&[\delta_{ij} = 1] \Rightarrow [z_{ij} = o_{ij}(k)]\\
&[\delta_{ij} = 0] \Rightarrow [z_{ij} = 0]
\end{aligned}
$$

This is the same as the four constraints:

$$\begin{cases} z_{ij} \leq M_{ij}\delta_{ij} \\ z_{ij} \geq m_{ij}\delta_{ij} \\ z_{ij} \leq o_{ij} - m_{ij}(1 - \delta_{ij}) \\ z_{ij} \geq o_{ij} - M_{ij}(1 - \delta_{ij}) \end{cases}$$

The same conversion holds for $z_i = \delta_i \cdot o_i$.

For the example model, all of the above can be summarized as the following MLD system:

$$x(k) = h_{11} + z_{11} + z_{12} + z_{13} + z_1 = \alpha_1 p(k) + \beta_1 + \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_{11}(k) \\ z_{12}(k) \\ z_{13}(k) \\ z_{21}(k) \\ z_{22}(k) \\ z_1(k) \end{bmatrix}$$

subject to:

s.t.

$$
\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} o_{11}(k) \\ o_{12}(k) \\ o_{13}(k) \\ o_{21}(k) \\ o_{22}(k) \\ o_1(k) \end{bmatrix}
+
\begin{bmatrix}
-m_{11} & 0 & 0 & 0 & 0 & \\
-M_{11}-\varepsilon & 0 & 0 & 0 & 0 & 0 \\
0 & -m_{12} & 0 & 0 & 0 & 0 \\
0 & -M_{12}-\varepsilon & 0 & 0 & 0 & 0 \\
0 & 0 & -m_{13} & 0 & 0 & 0 \\
0 & 0 & -M_{13}-\varepsilon & 0 & 0 & 0 \\
0 & 0 & 0 & -m_{21} & 0 & 0 \\
0 & 0 & 0 & -M_{21}-\varepsilon & 0 & 0 \\
0 & 0 & 0 & 0 & -m_{22} & 0 \\
0 & 0 & 0 & 0 & -M_{22}-\varepsilon & 0 \\
0 & 0 & 0 & 0 & 0 & m_1-\varepsilon \\
0 & 0 & 0 & 0 & 0 & M_1 \\
-M_{11} & 0 & 0 & 0 & 0 & 0 \\
m_{11} & 0 & 0 & 0 & 0 & 0 \\
-m_{11} & 0 & 0 & 0 & 0 & 0 \\
M_{11} & 0 & 0 & 0 & 0 & 0 \\
0 & -M_{12} & 0 & 0 & 0 & 0 \\
0 & m_{12} & 0 & 0 & 0 & 0 \\
0 & -m_{12} & 0 & 0 & 0 & 0 \\
0 & M_{12} & 0 & 0 & 0 & 0 \\
0 & 0 & -M_{13} & 0 & 0 & 0 \\
0 & 0 & m_{13} & 0 & 0 & 0 \\
0 & 0 & -m_{13} & 0 & 0 & 0 \\
0 & 0 & M_{13} & 0 & 0 & 0 \\
0 & 0 & 0 & -M_{21} & 0 & 0 \\
0 & 0 & 0 & m_{21} & 0 & 0 \\
0 & 0 & 0 & -m_{21} & 0 & 0 \\
0 & 0 & 0 & M_{21} & 0 & 0 \\
0 & 0 & 0 & -M_{21} & 0 & \\
0 & 0 & 0 & 0 & m_{22} & 0 \\
0 & 0 & 0 & 0 & -m_{22} & 0 \\
0 & 0 & 0 & 0 & M_{22} & 0 \\
0 & 0 & 0 & 0 & 0 & -M_1 \\
0 & 0 & 0 & 0 & 0 & m_1 \\
0 & 0 & 0 & 0 & 0 & -m_1 \\
0 & 0 & 0 & 0 & 0 & M_1
\end{bmatrix}
\begin{bmatrix} \delta_{11}(k) \\ \delta_{12}(k) \\ \delta_{13}(k) \\ \delta_{21}(k) \\ \delta_{22}(k) \\ \delta_1(k) \end{bmatrix}
+
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix} z_{11}(k) \\ z_{12}(k) \\ z_{13}(k) \\ z_{21}(k) \\ z_{22}(k) \\ z_1(k) \end{bmatrix}
+ \leq
\begin{bmatrix}
-m_{11} \\ -\varepsilon \\ -m_{12} \\ -\varepsilon \\ -m_{13} \\ -\varepsilon \\ -m_{21} \\ -\varepsilon \\ -m_{22} \\ -\varepsilon \\ -\varepsilon \\ M_1 \\ 0 \\ 0 \\ -m_{11} \\ M_{11} \\ 0 \\ 0 \\ -m_{12} \\ M_{12} \\ 0 \\ 0 \\ -m_{13} \\ M_{13} \\ 0 \\ 0 \\ -m_{21} \\ M_{21} \\ 0 \\ 0 \\ -m_{22} \\ M_{22} \\ 0 \\ 0 \\ -m_1 \\ M_1
\end{bmatrix}
$$

For the general system (Equation 2-4), this can be generalized in Theorem 5-1.1.

Theorem 5-1.1 (Rewriting a conjunctive MMPS system as an MLD system)**.** *Provided that Assumption 5-1.1 holds, the conjunctive MMPS system in Definition 2-5.1 can be written as the following MLD system*

$$x(k) = \alpha_{1,1} p(k) + \beta_{1,1} + \sum_{j=1}^{n_1-1} z_{1j} + \sum_{i=1}^{K-1} z_i \tag{5-2}$$

subject to:

for each maximization

$$
\begin{aligned}
&-o_{ij}(k) + (\varepsilon - m_{ij})\delta_i \leq -m_{ij} \\
&\quad o_{ij}(k) - M_{ij}\delta_{ij} \leq 0 \\
&\quad z_{ij} - M_{ij}\delta_{ij} \leq 0 \\
&-z_{ij} + m_{ij}\delta_{ij} \leq 0 \\
&-o_{ij} + z_{ij} - m_{ij}\delta_{ij} \leq -m_{ij} \\
&\quad o_{ij} - z_{ij} + M_{ij}\delta_{ij} \leq M_{ij} \\
&\quad \textit{for } i = 1, \ldots, K-1,\ j = 1, \ldots, n_i
\end{aligned} \tag{5-3}
$$

for each minimization

$$
\begin{aligned}
&-o_i(k) + (m_i - \varepsilon)\delta_i \leq -\varepsilon \\
&\ o_i(k) + M_i\delta_i \leq M_i \\
&\ z_i - M_i\delta_i \leq 0 \\
&-z_i + m_i\delta_i \leq 0 \\
&-o_i + z_i - m_i\delta_i \leq -m_i \\
&\ o_i - z_i + M_i\delta_i \leq M_i \\
&\ \textit{for } i = 1, \ldots, K-1
\end{aligned}
\tag{5-4}
$$

Next to that the bounds on the state and the input need to be added if these apply.

For larger systems it can happen that several o_{ij} descriptions are equal among themselves or equal to o_i. Then these descriptions (and their corresponding $\delta_{i(j)}$ and $z_{i(j)}$ variables) can be merged. This will result in a smaller system with fewer constraints.

When $x(k)$ is not one-dimensional, the model can be looked at component-wise. So

$$
\begin{aligned}
x(k) &= \begin{bmatrix} x_1(k) & x_2(k) & x_3(k) & x_4(k) \end{bmatrix}^T \\
x_1(k) &= \min_{i=1,\ldots,K_1} \max_{j=1,\ldots,n_{1,i}} (\alpha_{i,j}^T p(k) + \beta_{i,j})
\end{aligned}
$$

5-2 Rewriting the MLD system as an MIQP Problem

To be able to control the system with MPC, the newly acquired MLD system needs to be reformulated as a Mixed integer programming problem. Based on the objective function, the problem will either turn out to be a Mixed integer quadratic programming (MIQP) or a Mixed integer linear programming (MILP) problem.

Definition 5-2.1 (MLD-MPC Problem [15])**.** The MLD-MPC problem can be recast as a MIQP problem.

$$
\begin{aligned}
&\min_{\hat{V}(k)} \hat{V}(k)^T S_1 \hat{V}(k) + 2(S_2 + x^T(k) S_3)\hat{V}(k) \\
&\textbf{subject to: } F_1\hat{V}(k) \leq F_2 + F_3 x(k)
\end{aligned}
\tag{5-5}
$$

If $S_1 = 0$, the MLD-MPC problem is a MILP problem.

This thesis focuses on a quadratic cost function. To write the MLD-MPC problem in the way defined above, multiple steps have to be completed. These steps will be illustrated in the next subsections.

5-2-1 Rewriting the system model

The first step to reach the formulation in Definition 5-2.1 is to use successive substitution:

$$
\begin{aligned}
x(k) &= Ax(k-1) + B_1u(k) + B_2\delta(k) + B_3z(k) + B_4 \\
x(k+1) &= A(Ax(k-1) + B_1u(k) + B_2\delta(k) + B_3z(k) + B_4) + B_1u(k+1) + B_2\delta(k+1) \\
&\quad + B_3z(k+1) + B_4 \\
&\vdots \\
x(k+N_p-1) &= A^{N_p}x(k-1) + \textstyle\sum_{i=0}^{N_p-1} A^{j-i-1}(B_1u(k+i) + B_2\delta(k+i) + B_3z(k+i) + B_4)
\end{aligned}
$$

This can also be written in matrix format:

$$
\begin{aligned}
\begin{bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+N_p-1) \end{bmatrix} &= \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix}}_{M_2} x(k-1) + \underbrace{\begin{bmatrix} B_1 & 0 & \cdots & 0 \\ AB_1 & B_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1}B_1 & A^{N_p-2}B_1 & \cdots & B_1 \end{bmatrix}}_{T_1} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} \\
&+ \underbrace{\begin{bmatrix} B_2 & 0 & \cdots & 0 \\ AB_2 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1}B_2 & A^{N_p-2}B_2 & \cdots & B_2 \end{bmatrix}}_{T_2} \begin{bmatrix} \delta(k) \\ \delta(k+1) \\ \vdots \\ \delta(k+N_p-1) \end{bmatrix} \\
&+ \underbrace{\begin{bmatrix} B_3 & 0 & \cdots & 0 \\ AB_3 & B_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1}B_3 & A^{N_p-2}B_3 & \cdots & B_3 \end{bmatrix}}_{T_3} \begin{bmatrix} z(k) \\ z(k+1) \\ \vdots \\ z(k+N_p-1) \end{bmatrix} \\
&+ \underbrace{\begin{bmatrix} B_4 \\ AB_4 + B_4 \\ \vdots \\ A^{N_p-1}B_4 + A^{N_p-2}B_4 + \cdots + B_4 \end{bmatrix}}_{M_3}
\end{aligned} \tag{5-6}
$$

This is equal to the more compact matrix notation:

$$\hat{x}(k) = M_2x(k-1) + T_1\hat{u}(k) + T_2\hat{\delta}(k) + T_3\hat{z}(k) + M_3$$

where

$$\hat{s}(k) = \begin{bmatrix} s(k) & s(k+1) & \cdots & s(k+N_p-1) \end{bmatrix}^T$$

The input and auxiliary variables can be merge into the free variable vector $V(k)$. Now in the same way

$$\hat{V}(k) = \begin{bmatrix} \hat{u}(k)^T & \hat{\delta}(k)^T & \hat{z}(k)^T \end{bmatrix}^T$$

Using this $\hat{V}(k)$ results in:

$$\hat{x}(k) = M_2x(k-1) + \begin{bmatrix} T_1 & T_2 & T_3 \end{bmatrix} \hat{V}(k) + M_3 = M_2x(k-1) + M_1\hat{V}(k) + M_3 \tag{5-7}$$

5-2-2 Rewriting the constraints

The constraints from the MLD model are given as:

$$E_1x(k-1) + E_2u(k) + E_3\delta(k) + E_4z(k) \leq g_5$$

These need to be satisfied at each time step. Next to that, the final state $x(k+N_p-1)$ needs to be within the bounds on the state. This comes down to:

$$\underbrace{\begin{bmatrix} E_1 & 0 & \dots & 0 & 0 \\ 0 & E_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & E_1 & 0 \\ 0 & 0 & \dots & 0 & I \\ 0 & 0 & \dots & 0 & -I \end{bmatrix}}_{\hat{E}_1} \begin{bmatrix} x(k-1) \\ x(k) \\ \vdots \\ x(k+N_p-2) \\ x(k+N_p-1) \end{bmatrix} + \underbrace{\begin{bmatrix} E_2 & 0 & \dots & 0 \\ 0 & E_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_2 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_2} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} E_3 & 0 & \dots & 0 \\ 0 & E_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_3 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_3} \begin{bmatrix} \delta(k) \\ \delta(k+1) \\ \vdots \\ \delta(k+N_p-1) \end{bmatrix} + \underbrace{\begin{bmatrix} E_4 & 0 & \dots & 0 \\ 0 & E_4 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_4 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_4} \begin{bmatrix} z(k) \\ z(k+1) \\ \vdots \\ z(k+N_p-1) \end{bmatrix} \leq \underbrace{\begin{bmatrix} g_5 \\ g_5 \\ \vdots \\ g_5 \\ x_{max} \\ -x_{min} \end{bmatrix}}_{\hat{g}_5}$$

So in short:

$$\hat{E}_1 \begin{bmatrix} x(k-1) \\ \hat{x}(k) \end{bmatrix} + \hat{E}_2\hat{u}(k) + \hat{E}_3\hat{\delta}(k) + \hat{E}_4\hat{z}(k) \leq \hat{g}_5 \tag{5-8}$$

$\begin{bmatrix} x(k-1) \\ \hat{x}(k) \end{bmatrix}$ needs to be written in terms of $x(k-1)$:

$$\begin{bmatrix} x(k-1) \\ \hat{x}(k) \end{bmatrix} = \begin{bmatrix} I \\ M_2 \end{bmatrix} x(k-1) + \begin{bmatrix} 0 \\ M_1 \end{bmatrix} \hat{V}(k) + \begin{bmatrix} 0 \\ M_3 \end{bmatrix}$$

Filling this in in Equation 5-8 gives:

$$\underbrace{\left(\hat{E}_1 \begin{bmatrix} 0 \\ M_1 \end{bmatrix} + \begin{bmatrix} \hat{E}_2 & \hat{E}_3 & \hat{E}_4 \end{bmatrix}\right)}_{F_1} \hat{V}(k) \leq \underbrace{\hat{g}_5 - \hat{E}_1 \begin{bmatrix} 0 \\ M_3 \end{bmatrix}}_{F_2} \underbrace{-\hat{E}_1 \begin{bmatrix} I \\ M_2 \end{bmatrix}}_{F_3} x(k-1) \tag{5-9}$$

Combining the model in Equation 5-7 and the constraints in Equation 5-9, a description of the constrained system up to the planning horizon N_p is formulation in Equation 5-10. This is done in terms of the previous state and the free variables up to the prediction horizon.

$$\begin{aligned} &x(k) = M_1\hat{V}(k) + M_2x(k-1) + M_3 \\ &\textbf{subject to: } F_1\hat{V}(k) \leq F_2 + F_3x(k-1) \end{aligned} \tag{5-10}$$

5-2-3 Implementing a quadratic cost function

The last step in writing the MLD system as an MIQP problem is implementing a cost function. Just like in Section 3-2, this section will use the quadratic cost function in Equation 4-1.

In matrix format, the cost function looks like:

$$\begin{aligned}
J_2(x_0,\mathbf{u}) =& \Big(\begin{bmatrix} x(k-1) \\ \hat{x}(k) \end{bmatrix} - \underbrace{\begin{bmatrix} x_{eq}(k-1) \\ x_{eq}(k) \\ \vdots \\ x_{eq}(k+N_p-1) \end{bmatrix}}_{\hat{x}_{eq}} \Big)^T \underbrace{\begin{bmatrix} Q & 0 & \cdots & 0 & 0 \\ 0 & Q & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Q & 0 \\ 0 & 0 & \cdots & 0 & P \end{bmatrix}}_{\hat{Q}} \Big(\begin{bmatrix} x(k-1) \\ \hat{x}(k) \end{bmatrix} - \underbrace{\begin{bmatrix} x_{eq}(k-1) \\ x_{eq}(k) \\ \vdots \\ x_{eq}(k+N_p-1) \end{bmatrix}}_{\hat{x}_{eq}} \Big) \\
&+ \Big(\hat{u}(k) - \underbrace{\begin{bmatrix} u_{eq}(k-1) \\ u_{eq}(k) \\ \vdots \\ u_{eq}(k+N_p-1) \end{bmatrix}}_{\hat{u}_{eq}} \Big)^T \underbrace{\begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix}}_{\hat{R}} \Big(\hat{u}(k) - \underbrace{\begin{bmatrix} u_{eq}(k-1) \\ u_{eq}(k) \\ \vdots \\ u_{eq}(k+N_p-1) \end{bmatrix}}_{\hat{u}_{eq}} \Big)
\end{aligned}$$

Now the state evolution can be filled in in this cost function.

$$\begin{aligned}
J_2(x_0,\mathbf{u}) =& \Big(\begin{bmatrix} I \\ M_2 \end{bmatrix} \begin{bmatrix} x(k-1) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ M_1 \end{bmatrix} \hat{V}(k) + \begin{bmatrix} 0 \\ M_3 \end{bmatrix} - \tilde{x}_{eq} \Big)^T \hat{Q} \Big(\begin{bmatrix} I \\ M_2 \end{bmatrix} \begin{bmatrix} x(k-1) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ M_1 \end{bmatrix} \hat{V}(k) + \begin{bmatrix} 0 \\ M_3 \end{bmatrix} - \tilde{x}_{eq} \Big) \\
&+ \Big(\hat{V}(k) - \begin{bmatrix} \hat{u}_{eq} \\ 0 \\ 0 \end{bmatrix} \Big)^T \begin{bmatrix} \hat{R} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \Big(\hat{V}(k) - \begin{bmatrix} \hat{u}_{eq} \\ 0 \\ 0 \end{bmatrix} \Big) \\
=& \hat{V}^T(k) \Big(\begin{bmatrix} 0 \\ M_1 \end{bmatrix}^T \hat{Q} \begin{bmatrix} 0 \\ M_1 \end{bmatrix} + \begin{bmatrix} \hat{R} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \Big) \hat{V}(k) \\
&+ 2 \Big(x^T(k-1) \begin{bmatrix} I \\ M_2 \end{bmatrix}^T \hat{Q} \begin{bmatrix} 0 \\ M_1 \end{bmatrix} + \big(\begin{bmatrix} 0 \\ M_3 \end{bmatrix} - \hat{x}_{eq} \big)^T \hat{Q} \begin{bmatrix} 0 \\ M_1 \end{bmatrix} - \begin{bmatrix} \hat{u}_{eq} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \hat{R} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \Big) \hat{V}(k) \\
&+ \Big(\begin{bmatrix} I \\ M_2 \end{bmatrix} x(k-1) + \begin{bmatrix} 0 \\ M_3 \end{bmatrix} - \hat{x}_{eq} \Big)^T \hat{Q} \Big(\begin{bmatrix} I \\ M_2 \end{bmatrix} x(k-1) + \begin{bmatrix} 0 \\ M_3 \end{bmatrix} - \hat{x}_{eq} \Big) + \begin{bmatrix} \hat{u}_{eq} \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} \hat{R} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{u}_{eq} \\ 0 \\ 0 \end{bmatrix}
\end{aligned} \tag{5-11}$$

When the cost function is minimized, only the terms that include the free variable $\hat{V}(k)$ have to be taken into account. So with the 2-norm cost function, the minimization will look like:

$$\begin{aligned}
&\min_{\hat{V}(k)} \hat{V}(k)^T S_1 \hat{V}(k) + 2(S_2 + x^T(k-1) S_3) \hat{V}(k) \\
&\textbf{subject to: } F_1 \hat{V}(k) \leq F_2 + F_3 x(k-1)
\end{aligned} \tag{5-12}$$

where

$$\begin{aligned}
S_1 &= \begin{bmatrix} 0 \\ M_1 \end{bmatrix}^T \hat{Q} \begin{bmatrix} 0 \\ M_1 \end{bmatrix} + \begin{bmatrix} \hat{R} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
S_2 &= \big(\begin{bmatrix} 0 \\ M_3 \end{bmatrix} - \hat{x}_{eq} \big)^T \hat{Q} \begin{bmatrix} 0 \\ M_1 \end{bmatrix} - \begin{bmatrix} \hat{u}_{eq} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \hat{R} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
S_3 &= \begin{bmatrix} I \\ M_2 \end{bmatrix}^T \hat{Q} \begin{bmatrix} 0 \\ M_1 \end{bmatrix}
\end{aligned} \tag{5-13}$$

This is indeed an MIQP problem.

How the cost matrices Q, R and P should be chosen to ensure a stable closed-loop system, has been discussed in Chapter 4.

5-3 Conclusion

This chapter gave an overview of the steps that need to be taken to transform an MMPS system into an MIQP problem, that can be solved using MPC. Every MMPS system can be recast as an conjunctive MMPS system. Using auxiliary variables, all bounded conjunctive MMPS systems can be transformed into an MLD system. This chapter used a quadratic cost function to transform perform the last transformation: from an MLD system to an MIQP problem. Here the altered cost function from Equation 4-1 was used, such that it works when the equilibrium value of the state and the input is not constant or equal to zero.

All steps that are taken in this chapter are not necessarily difficult, but especially rewriting the MMPS system as an MLD system (Section 5-1) can be quite a tedious process. Next to that, this is a step where errors can be made quite easily, which would mess up the optimization in the end. Caution is therefore recommended.

Chapter 6

Case Study: Urban Railway Line

This chapter applies the research in this Master thesis project to a real life example. The system that will be used is a partially homogeneous discrete-event (DE) Max-Min-Plus-Scaling (MMPS) system of an urban railway line. This chapter will run through the same steps as that were taken in Chapter 5. Next to that, the stability of the system will be investigated by a disturbance rejection analysis and by looking at the consequences of parametric uncertainty.

The chapter starts off by introducing the parameters that are present in the system in Table 6-1 and by initializing the state vector. Furthermore, the system description is presented.

Description	**Parameter**
trains	k
stations	j
arrival time at station k	$a_j(k)$
departure time at station k	$d_j(k)$
# passengers in train k when leaving station j	$\rho_j(k)$
# passengers at station j when train k leaves	$\sigma_j(k)$
maximum capacity train	ρ_{max}
running time from station j-1 to j	$\tau_{r,j}$
# passengers entering station j per minute	e_j
# passengers that can board the train per minute	b
# passengers that can disembark the train per minute	f
fraction of passengers in train k that leave the train k at station j	β_j
headway time	τ_H

Table 6-1: List of model parameters

There are several variables that depend on these parameters. These are summarized in Table 6-2.

α **values**	γ **values**
$\alpha_1 = \frac{b}{b-e_j}$	$\gamma_0 = \frac{1}{b}\rho_{max}$
$\alpha_2 = \frac{b}{b-e_j}\frac{\beta_j}{f}$	$\gamma_1 = 1$
$\alpha_3 = \frac{1}{b-e_j}$	$\gamma_2 = -\frac{1-\beta_j}{b} + \frac{\beta_j}{f}$
$\alpha_4 = -\frac{e_j}{b-e_j}$	

Table 6-2: Computation of α and γ values

The initial values of the system are equal to:

$$\begin{array}{ll} \textbf{for j = 1} & \textbf{for k = 0} \\ a_1(k) = \max(\tau_{r,1}\ ,\ k\bar{\tau}_1 + \tau_H) & a_j(0) = \max((j-1)\bar{\tau}_0 + \tau_{r,j}\ ,\ \tau_H) \\ d_1(k) = (k+1)\bar{\tau}_1 & d_j(0) = j\bar{\tau}_0 \\ \rho_1(k) = \bar{\rho}_1 & \rho_j(0) = \bar{\rho}_0 \\ \sigma_1(k) = 0 & \sigma_j(0) = 0 \end{array} \tag{6-1}$$

Using this initialization and the parameters in Table 6-1, the urban railway model can be described by Equation 6-2.

for j > 1 and k > 0:

$$a_j(k) = \max\left(d_{j-1}(k) + \tau_{r,j}\ ,\ d_j(k-1) + \tau_H\right) \tag{6-2a}$$

$$d_j(k) = \min\left(\alpha_1 a_j(k) + \alpha_2 \rho_{j-1}(k) + \alpha_3 \sigma_j(k-1) + \alpha_4 d_j(k-1)\ ,\ \gamma_0 + \gamma_1 a_j(k) + \gamma_2 \rho_{j-1}(k)\right) \tag{6-2b}$$

$$\rho_j(k) = (1-\beta_j)\rho_{j-1}(k) + b\big(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)\big) \tag{6-2c}$$

$$\sigma_j(k) = \sigma_j(k-1) + e_j\big(d_j(k) - d_j(k-1)\big) - b\big(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)\big) \tag{6-2d}$$

These equations can be interpreted in the following way. The arrival time $a_j(k)$ is the maximum of the time train k can be at station j and the time the previous train has enough headway time from station j. The departure time $d_j(k)$ is the minimum of the time that either the platform is empty or that the train is full. The people in the train when it leaves the station $\rho_j(k)$ is the amount of people in train at previous station $\rho_{j-1}(k)$ minus the amount of people getting out at the station plus the amount of people getting in. The amount of people at the platform when the train leaves $\sigma_j(k)$ is equal to the amount of people left on the platform when previous train left $\sigma_j(k-1)$ plus the amount of people that arrived in the meantime minus the amount of people getting in in train k.

The four states are combined in one state vector

$$x_j(k) = \begin{bmatrix} a_j(k) \\ d_j(k) \\ \rho_j(k) \\ \sigma_j(k) \end{bmatrix} \quad x(k) = \begin{bmatrix} x_2(k) \\ \vdots \\ x_{J-1}(k) \\ x_J(k) \end{bmatrix}$$

where J is the total amount of stations.

6-1 Rewriting the System

To be able to efficiently optimize the urban railway line with Model predictive control (MPC), it needs to be written as an Mixed integer quadratic programming (MIQP) problem. The first step in achieving this is to add an input to the system, such that control is possible. After that, this section carries out the rewriting process (as described in Chapter 5), where it starts with rewriting the state vector.

6-1-1 Adding an input vector

The system in Equation 6-2 does not include a control variable, this is why the first step in rewriting the system is to add this. The input that is chosen for this system is an extra waiting time at the station. This means it should be added to $d_j(k)$. However, it does not make sense to let the train wait longer if the train is already full. So the input is only added to the first term of the minimization, which corresponds to a situation where the train would leave because the platform is empty.

Next to that, it will be assumed that the amount of people that can board the train per minute b is higher than the amount of people that will enter the platform per minute e_j. This means that all people that entered the station in the extra waiting time $u_j(k)$ have actually boarded the train. So $\sigma_j(k)$ is not influenced by the input and $e_j u_j(k)$ should be added to the formula for $\rho_j(k)$. So we end up with the system in Equation 6-3.

$$
\begin{aligned}
a_j(k) &= \max\left(d_{j-1}(k) + \tau_{r,j}\ ,\ d_j(k-1) + \tau_H\right) \\
d_j(k) &= \min\left(\alpha_1 a_j(k) + \alpha_2 \rho_{j-1}(k) + \alpha_3 \sigma_j(k-1) + \alpha_4 d_j(k-1) + u_j(k)\ ,\ \gamma_0 + \gamma_1 a_j(k) + \gamma_2 \rho_{j-1}(k)\right) \\
\rho_j(k) &= (1-\beta_j)\rho_{j-1}(k) + b\big(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)\big) + e_j u_j(k) \\
\sigma_j(k) &= \sigma_j(k-1) + e_j\big(d_j(k) - d_j(k-1)\big) - b\big(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)\big)
\end{aligned}
\tag{6-3}
$$

6-1-2 Rewriting the state vector

To rewrite the state vector, the state vector first needs to be separated in time and quantitative variables:

$$
x_{j,t}(k) = \begin{bmatrix} a_j(k) \\ d_j(k) \end{bmatrix} \quad x_{j,q}(k) = \begin{bmatrix} \rho_j(k) \\ \sigma_j(k) \end{bmatrix} \tag{6-4}
$$

As it is not possible to describe the difference in the time variables ($a_j(k) - a_j(k-1)$ and $d_j(k) - d_j(k-1)$) without the original time variables, the new state vector per station will consist of the following six states:

$$x_j(k) = \begin{bmatrix} a_j(k) - a_j(k-1) \\ d_j(k) - d_j(k-1) \\ a_j(k) \\ d_j(k) \\ \rho_j(k) \\ \sigma_j(k) \end{bmatrix} \tag{6-5}$$

6-1-3 Rewriting to conjunctive canonical form

Following Theorem 2-5.1, the general formulation of an MMPS system can rewritten into a conjunctive canonical formulation.

The conjunctive canonical form is defined as

$$x(k) = \min_{i=1,\dots,K} \max_{l=1,\dots,n_i} (\alpha_{i,l}^T p(k) + \beta_{i,l})$$

Each of the six states in the state description can be rewritten into this format. As the size of some of these descriptions is rather large, the full description can be found in Appendix A-1. $\rho_j(k)$ and $\sigma_j(k)$ are described as a min-max-min formulation, because then the size of the system remains smaller and from there it is already possible to rewrite the system as an Mixed logical dynamical (MLD) system. Here, only the sizes of the six state descriptions are listed.

$$\begin{aligned} a_j(k) - a_j(k-1) &= \min_{i_1=1} \max_{l_1=1,2} (\eta_{i_1,l_1}^T x_j(k-1) + \lambda_{i_1,l_1} x_{j-1}(k) + \mu_{i_1,l_1}) \\ d_j(k) - d_j(k-1) &= \min_{i_2=1,2} \max_{l_2=1,2} (\eta_{i_2,l_2}^T x_j(k-1) + \lambda_{i_2,l_2} x_{j-1}(k) + \mu_{i_2,l_2}) \\ a_j(k) &= \min_{i_1=3} \max_{l_3=1,2} (\eta_{i_3,l_3}^T x_j(k-1) + \lambda_{i_3,l_3} x_{j-1}(k) + \mu_{i_3,l_3}) \\ d_j(k) &= \min_{i_4=1,2} \max_{l_4=1,2} (\eta_{i_4,l_4}^T x_j(k-1) + \lambda_{i_4,l_4} x_{j-1}(k) + \mu_{i_4,l_4}) \\ \rho_j(k) &= \min_{i_5=1,2} \max_{l_5=1,2} \min_{r_5=1,2} (\eta_{i_5,l_5}^T x_j(k-1) + \lambda_{i_5,l_5} x_{j-1}(k) + \mu_{i_5,l_5}) \\ \sigma_j(k) &= \min_{i_6=1,2} \max_{l_6=1,2} \min_{r_6=1,\dots,8} (\eta_{i_6,l_6}^T x_j(k-1) + \lambda_{i_6,l_6} x_{j-1}(k) + \mu_{i_6,l_6}) \end{aligned} \tag{6-6}$$

6-1-4 Rewriting to MLD system

An MMPS system can be recast as an MLD system if Assumption 5-1.1 holds. This means that no term in the maximizations should be equal to $-\infty$ and no term in the minimizations should be equal to $+\infty$. Since this condition is satisfied for all terms in the formulation in Equation 6-6, the MMPS system can indeed be recast as an MLD system.

To do this we follow the procedure in Section 5-1, for each of the 6 states separately, starting from the formulation in Equation 6-6. When doing this, one will notice that there is a large portion of identical o_{ij} and o_i functions. These can be replaced by only one o value and thus one pair of auxiliary variables δ and z. In the end, this results in the system per station described in Equation 6-7.

$$
\begin{aligned}
x_j(k) = {} & \underbrace{\begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\alpha_1 & 0 & \alpha_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_4 & 0 & \alpha_3 \\ 0 & 0 & 0 & b\alpha_4 & 0 & \alpha_1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{1,j}} x_j(k-1) + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ e_j \\ 0 \end{bmatrix}}_{B_{1,j}} u_j(k) + \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}}_{B_3} z_j(k) + \\
& \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & \alpha_1 & 0 \\ 0 & 1 & 0 \\ 0 & \alpha_1 & 0 \\ 0 & -b\alpha_4 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix}}_{B_{4,j}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha_1 & \alpha_2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha_1 & \alpha_2 & 0 \\ 0 & 0 & 0 & -b\alpha_4 & b(\alpha_2-\gamma_2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{11,j}} x_{j-1}(k)
\end{aligned}
\tag{6-7}
$$

subject to:

$$E_{1,j}x_j(k-1) + E_{2,j}u_j(k) + E_{3,j}\delta_j(k) + E_4 z_j(k) \leq g_{5,j} - E_{11,j}x_{j-1}(k)$$

The set of constraints is fully written out in Appendix A-2. Here the last nine constraints are the state and input constraints. Each variable has a lower bound of zero. $\rho_j(k)$ is the only state with a constraint on the maximal value (the train capacity). If there is a maximal extra waiting time, this can be incorporated by choosing $u_{max} < \infty$. So in short, the MLD model equivalent to the original MMPS system in Equation 6-2 is:

$$
\begin{aligned}
& x_j(k) = A_{1,j}x_j(k-1) + B_{1,j}u_j(k) + B_2 z_j(k) + B_{4,j} + A_{11,j}x_{j-1}(k) \\
& \text{s.t. } E_{1,j}x_j(k-1) + E_{2,j}u_j(k) + E_{3,j}\delta_j(k) + E_4 z_j(k) \leq g_{5,j} - E_{11,j}x_{j-1}(k)
\end{aligned}
\tag{6-8}
$$

It is desired to have a system for the entire network (not per station) and we want this system to only depend on the previous state $x(k-1)$, the input $u(k)$ and known variables. To achieve this, we need to get rid of $x_{j-1}(k)$. So each state description of the previous station is filled in in $x_j(k)$, until it depends on the the state at station 1: $x_1(k)$. This is an initial condition.

For the state description and the constraints, this results in the following description. Here all matrices, apart form the last one, have a lower block triangular structure.

$$
\begin{aligned}
x(k) = \begin{bmatrix} x_2(k) \\ x_3(k) \\ \vdots \\ x_J(k) \end{bmatrix} = {} & \begin{bmatrix} A_{1,2} & 0 & \cdots & 0 \\ A_{11,3}A_{1,2} & A_{1,3} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{11,J}\cdots A_{11,3}A_{1,2} & A_{11,J}\cdots A_{11,4}A_{1,3} & \cdots & A_{1,J} \end{bmatrix} x(k-1) + \begin{bmatrix} B_{1,2} & 0 & \cdots & 0 \\ A_{11,3}B_{1,2} & B_{1,3} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{11,J}\cdots A_{11,3}B_{1,2} & A_{11,J}\cdots A_{11,4}B_{1,3} & \cdots & B_{1,J} \end{bmatrix} u(k) \\
& + \begin{bmatrix} B_{3,2} & 0 & \cdots & 0 \\ A_{11,3}B_{3,2} & B_{3,3} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{11,J}\cdots A_{11,3}B_{3,2} & A_{11,J}\cdots A_{11,4}B_{3,3} & \cdots & B_{3,J} \end{bmatrix} z(k) + \begin{bmatrix} I & 0 & \cdots & 0 \\ A_{11,3} & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{11,J}\cdots A_{11,3} & A_{11,J}\cdots A_{11,4} & \cdots & I \end{bmatrix} \begin{bmatrix} B_{4,2} \\ B_{4,3} \\ \vdots \\ B_{4,J} \end{bmatrix} \\
& + \begin{bmatrix} A_{11,2} \\ A_{11,3}A_{11,2} \\ \vdots \\ A_{11,J}\cdots A_{11,2} \end{bmatrix} x_1(k)
\end{aligned}
$$

subject to:

$$\begin{bmatrix} E_{1,2} & 0 & 0 & \cdots & 0 \\ E_{11,3}A_{1,2} & E_{1,3} & 0 & \cdots & 0 \\ E_{11,4}A_{11,3}A_{1,2} & E_{11,4}A_{1,3} & E_{1,4} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E_{11,J}A_{11,J-1}\cdots A_{11,3}A_{1,2} & E_{11,J}A_{11,J-1}\cdots A_{11,4}A_{1,3} & E_{11,J}A_{11,J-1}\cdots A_{11,5}A_{1,4} & \cdots & E_{1,J} \end{bmatrix} x(k-1) + \begin{bmatrix} E_{2,2} & 0 & 0 & \cdots & 0 \\ E_{11,3}B_{1,2} & E_{2,3} & 0 & \cdots & 0 \\ E_{11,4}A_{11,3}B_{1,2} & E_{11,4}B_{1,3} & E_{2,4} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E_{11,J}A_{11,J-1}\cdots A_{11,3}B_{1,2} & E_{11,J}A_{11,J-1}\cdots A_{11,4}B_{1,3} & E_{11,J}A_{11,J-1}\cdots A_{11,5}B_{1,4} & \cdots & E_{2,J} \end{bmatrix} u(k)$$
$$+ \begin{bmatrix} E_{3,2} & 0 & 0 & \cdots & 0 \\ E_{11,3}B_{2,2} & E_{3,3} & 0 & \cdots & 0 \\ E_{11,4}A_{11,3}B_{2,2} & E_{11,4}B_{2,3} & E_{3,4} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E_{11,J}A_{11,J-1}\cdots A_{11,3}B_{2,2} & E_{11,J}A_{11,J-1}\cdots A_{11,4}B_{2,3} & E_{11,J}A_{11,J-1}\cdots A_{11,5}B_{2,4} & \cdots & E_{3,J} \end{bmatrix} \delta(k) + \begin{bmatrix} E_{4,2} & 0 & 0 & \cdots & 0 \\ E_{11,3}B_{3,2} & E_{4,3} & 0 & \cdots & 0 \\ E_{11,4}A_{11,3}B_{3,2} & E_{11,4}B_{3,3} & E_{4,4} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E_{11,J}A_{11,J-1}\cdots A_{11,3}B_{3,2} & E_{11,J}A_{11,J-1}\cdots A_{11,4}B_{3,3} & E_{11,J}A_{11,J-1}\cdots A_{11,5}B_{3,4} & \cdots & E_{4,J} \end{bmatrix} z(k)$$
$$\leq \begin{bmatrix} g_{5,2} \\ g_{5,3} \\ \vdots \\ g_{5,J} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ E_{11,3} & 0 & 0 & \cdots & 0 \\ E_{11,4}A_{11,3}A_{1,2} & E_{11,4} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E_{11,J}A_{11,J-1}\cdots A_{11,3} & E_{11,J}A_{11,J-1}\cdots A_{11,4} & E_{11,J}A_{11,J-1}\cdots A_{11,5} & \cdots & 0 \end{bmatrix} \begin{bmatrix} B_{4,2} \\ B_{4,3} \\ \vdots \\ B_{4,J} \end{bmatrix} - \begin{bmatrix} E_{11,2} \\ E_{11,3}A_{11,2} \\ \vdots \\ E_{11,J}A_{11,J-1}\cdots A_{11,2} \end{bmatrix} x_1(k)$$

Or in short:

$$\begin{aligned} &x(k) = A_1 x(k-1) + B_{1,}u(k) + B_3 z(k) + B_4 + A_{11}x_1(k) \\ &\text{s.t. } \; E_1 x(k-1) + E_2 u(k) + E_3 \delta(k) + E_4 z(k) \leq g_5 - E_{11}x_1(k) \end{aligned} \tag{6-9}$$

6-1-5 Rewriting to MIQP problem

The description of the MLD system (Equation 6-9) is very similar to the description in the general procedure that was followed in Section 5-2. Only the terms related to $x_1(k)$ are added. This is why in this section only the main steps in the transformation to an MIQP problem will be stated.

By successive substitution the model can be written as:

$$\begin{bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+N_p-1) \end{bmatrix} = \underbrace{\begin{bmatrix} A_1 \\ A_1^2 \\ \vdots \\ A_1^{N_p} \end{bmatrix}}_{M_2} x(k-1) + \underbrace{\begin{bmatrix} B_1 & 0 & \cdots & 0 \\ A_1B_1 & B_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_1^{N_p-1}B_1 & A_1^{N_p-2}B_1 & \cdots & B_1 \end{bmatrix}}_{T_1} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} + \underbrace{\begin{bmatrix} B_3 & 0 & \cdots & 0 \\ A_1B_3 & B_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_1^{N_p-1}B_3 & A_1^{N_p-2}B_3 & \cdots & B_3 \end{bmatrix}}_{T_3} \begin{bmatrix} z(k) \\ z(k+1) \\ \vdots \\ z(k+N_p-1) \end{bmatrix}$$
$$+ \underbrace{\begin{bmatrix} B_4 \\ A_1B_4 + B_4 \\ \vdots \\ A_1^{N_p-1}B_4 + A_1^{N_p-2}B_4 + \cdots + B_4 \end{bmatrix}}_{M_3} + \underbrace{\begin{bmatrix} A_{11} & 0 & \cdots & 0 \\ A_1A_{11} & A_{11} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_1^{N_p-1}A_{11} & A_1^{N_p-2}A_{11} & \cdots & A_{11} \end{bmatrix}}_{M_{x_1}} \begin{bmatrix} x_1(k) \\ x_1(k+1) \\ \vdots \\ x_1(k+N_p-1) \end{bmatrix}$$

This is equal to the more compact matrix notation:

$$\begin{aligned} \hat{x}(k) =& M_2 x(k-1) + T_1 \hat{u}(k) + T_3 \hat{z}(k) + M_3 + M_{x_1}\hat{x}_1(k) \\ =& M_2 x(k-1) + M_1 \hat{V}(k) + M_3 + M_{x_1}\hat{x}_1(k) \end{aligned} \tag{6-10}$$

The constraints can be written as:

$$\underbrace{\begin{bmatrix} E_1 & 0 & \dots & 0 & 0 \\ 0 & E_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & E_1 & 0 \\ 0 & 0 & \dots & 0 & I \\ 0 & 0 & \dots & 0 & -I \end{bmatrix}}_{\hat{E}_1} \begin{bmatrix} x(k-1) \\ x(k) \\ \vdots \\ x(k+N_p-2) \\ x(k+N_p-1) \end{bmatrix} + \underbrace{\begin{bmatrix} E_2 & 0 & \dots & 0 \\ 0 & E_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_2 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_2} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} + \underbrace{\begin{bmatrix} E_3 & 0 & \dots & 0 \\ 0 & E_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_3 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_3} \begin{bmatrix} \delta(k) \\ \delta(k+1) \\ \vdots \\ \delta(k+N_p-1) \end{bmatrix}$$

$$+\underbrace{\begin{bmatrix} E_4 & 0 & \dots & 0 \\ 0 & E_4 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_4 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_4} \begin{bmatrix} z(k) \\ z(k+1) \\ \vdots \\ z(k+N_p-1) \end{bmatrix} \leq \underbrace{\begin{bmatrix} g_5 \\ g_5 \\ \vdots \\ g_5 \\ x_{max} \\ -x_{min} \end{bmatrix}}_{\hat{g}_5} - \underbrace{\begin{bmatrix} E_{11} & 0 & \dots & 0 \\ 0 & E_{11} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_{11} \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_{11}} \begin{bmatrix} x_1(k) \\ x_1(k+1) \\ \vdots \\ x_1(k+N_p-1) \end{bmatrix}$$

So in short:

$$\hat{E}_1 \begin{bmatrix} x(k-1) \\ \hat{x}(k) \end{bmatrix} + \hat{E}_2\hat{u}(k) + \hat{E}_3\hat{\delta}(k) + \hat{E}_4\hat{z}(k) \leq \hat{g}_5 - \hat{E}_{11}\hat{x}_1(k) \tag{6-11}$$

which is equal to:

$$\underbrace{\left(\hat{E}_1 \begin{bmatrix} 0 \\ M_1 \end{bmatrix} + \begin{bmatrix} \hat{E}_2 & \hat{E}_3 & \hat{E}_4 \end{bmatrix}\right)}_{F_1} \hat{V}(k) \leq \underbrace{\hat{g}_5 - (\hat{E}_{11} - \hat{E}_1 \begin{bmatrix} 0 \\ M_{x_1} \end{bmatrix})\hat{x}_1(k) - \hat{E}_1 \begin{bmatrix} 0 \\ M_3 \end{bmatrix}}_{F_2} \underbrace{-\hat{E}_1 \begin{bmatrix} I \\ M_2 \end{bmatrix}}_{F_3} x(k-1) \tag{6-12}$$

When combining this with a quadratic cost function, this can be recast as an MIQP problem. For the urban railway line, this means that the MIQP problem can be written as:

$$\begin{aligned} &\min_{\hat{V}(k)} \hat{V}(k)^T S_1 \hat{V}(k) + 2(S_2 + x^T(k-1)S_3)\hat{V}(k) \\ &\textbf{subject to: } F_1\hat{V}(k) \leq F_2 + F_3 x(k-1) \end{aligned} \tag{6-13}$$

where

$$\begin{aligned} S_1 &= \begin{bmatrix} 0 \\ M_1 \end{bmatrix}^T \hat{Q} \begin{bmatrix} 0 \\ M_1 \end{bmatrix} + \begin{bmatrix} \hat{R} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ S_2 &= (\begin{bmatrix} 0 \\ M_3 \end{bmatrix} + \begin{bmatrix} 0 \\ M_{x_1} \end{bmatrix} \hat{x}_1(k) - \hat{x}_{eq})^T \hat{Q} \begin{bmatrix} 0 \\ M_1 \end{bmatrix} - \begin{bmatrix} \hat{u}_{eq} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \hat{R} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ S_3 &= \begin{bmatrix} I \\ M_2 \end{bmatrix}^T \hat{Q} \begin{bmatrix} 0 \\ M_1 \end{bmatrix} \\ F_1 &= \hat{E}_1 \begin{bmatrix} 0 \\ M_1 \end{bmatrix} + \begin{bmatrix} \hat{E}_2 & \hat{E}_3 & \hat{E}_4 \end{bmatrix} \\ F_2 &= \hat{g}_5 - (\hat{E}_{11} - \hat{E}_1 \begin{bmatrix} 0 \\ M_{x_1} \end{bmatrix})\hat{x}_1(k) - \hat{E}_1 \begin{bmatrix} 0 \\ M_3 \end{bmatrix} \\ F_3 &= -\hat{E}_1 \begin{bmatrix} I \\ M_2 \end{bmatrix} \end{aligned}$$

A quadratic cost function makes sense for the urban railway line model, because large delays will the penalized more severely. For an urban railway line there is not a clear timetable. This means that small delays are not a very large problem. Next to that, it is not possible to overtake in this model. So trains with a large delay can really clog the system.

6-2 The Cost function

This section follows the procedure from Chapter 3, together with the alterations proposed in Chapter 4.

6-2-1 Determining the equilibrium values

To determine the equilibrium values of the urban railway line, one has to take a look at the original MMPS system including the input, together with the conditions that should hold at the equilibrium.

$$\begin{aligned}
a_j(k) &= \max\left(d_{j-1}(k) + \tau_{r,j}\ ,\ d_j(k-1) + \tau_H\right) \\
d_j(k) &= \min\left(\alpha_1 a_j(k) + \alpha_2 \rho_{j-1}(k) + \alpha_3 \sigma_j(k-1) + \alpha_4 d_j(k-1) + u_j(k)\ ,\ \gamma_0 + \gamma_1 a_j(k) + \gamma_2 \rho_{j-1}(k)\right) \\
\rho_j(k) &= (1-\beta_j)\rho_{j-1}(k) + b(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k)) + e_j u_j(k) \\
\sigma_j(k) &= \sigma_j(k-1) + e_j(d_j(k) - d_j(k-1)) - b(d_j(k) - a_j(k) - \frac{\beta_j}{f}\rho_{j-1}(k))
\end{aligned}$$

The conditions at the equilibrium are:

$$\begin{aligned}
a_j(k) - a_j(k-1) &= a_j(k-1) - a_j(k-2) \\
d_j(k) - d_j(k-1) &= d_j(k-1) - d_j(k-2) \\
\rho_j(k) &= \rho_j(k-1) \\
\sigma_j(k) &= \sigma_j(k-1)
\end{aligned} \tag{6-14}$$

Apart from these constraints we will also assume: $\rho_j(k) = \rho_{j-1}(k)$, because this will make the computation easier and in practice it would be preferable to keep the amount of people in a train equal over the whole network.

If we use the description of $\sigma_j(k)$ together with the fourth condition, we can see that: $d_j(k) = \alpha_1 a_j(k) + \alpha_2 \rho_{eq} + \alpha_4 d_j(k-1)$. This means that $\sigma_{eq} = 0$ and $u_{eq} = 0$.

The description of $\rho_j(k)$ together with the extra constraint shows us that: $a_j(k) = d_j(k) - (\frac{1}{b} + \frac{1}{f})\beta_j \rho_{eq}$. From this we can conclude that $a_j(k) - a_j(k-1) = d_j(k) - d_j(k-1)$. When combining this with the previous expression, it can be concluded that $d_j(k) - d_j(k-1) = \frac{\beta_j}{e_j}\rho_{eq}$.

This means that the equilibrium values for the whole state vector will be

$$\begin{aligned}
x_{j,eq} &= \begin{bmatrix} \frac{\beta_j}{e_j}\rho_{eq} \\ \frac{\beta_j}{e_j}\rho_{eq} \\ a_j(0) + k\frac{\beta_j}{e_j}\rho_{eq} \\ d_j(0) + k\frac{\beta_j}{e_j}\rho_{eq} \\ \rho_{eq} \\ 0 \end{bmatrix} \\
u_{j,eq} &= 0
\end{aligned} \tag{6-15}$$

It is obvious that the equilibrium value for $a_j(k)$ and $d_j(k)$ are not constant for a certain station, as they depend on the event counter k, but they are known.

6-2-2 Determining the terminal cost

To determine the terminal cost, the first step is to linearize the system around its equilibrium, which has been computed in the previous subsection. This linearization will have the following form:

$$x(k) = A_{1,eq}x(k-1) + B_{1,eq}u(k)$$

The choice for the parameters of the system has an influence on the position of the equilibrium and thus on the part of the MMPS system that is used in that equilibrium (as described in Section 4-4). This means that we need to make sure that it holds that: $x_{j,eq}(k) = A_{1j,eq}x_j(k-1) + B_{1j,eq}u_j(k) + A_{11j,eq}x_{j-1}(k) + B_{4j,eq}$ for $j = 2, \ldots, J$.

After a few iterations we find out that the part of the conjunctive MMPS formulation that is responsible for the value of $x_j(k)$ at the equilibrium is:

$$x_j(k) = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\alpha_1 & 0 & \alpha_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_4 & 0 & \alpha_3 \\ 0 & 0 & 0 & b\alpha_4 & 0 & \alpha_1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha_1 & \alpha_2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha_1 & \alpha_2 & 0 \\ 0 & 0 & 0 & -b\alpha_4 & b(\alpha_2 - \gamma_2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_{j-1}(k) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ e_j \\ 0 \end{bmatrix} u_j(k) + \begin{bmatrix} 0 & 1 & 0 \\ 0 & \alpha_1 & 0 \\ 0 & 1 & 0 \\ 0 & \alpha_1 & 0 \\ 0 & -b\alpha_4 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} \tag{6-16}$$

For the parameters, it then needs to hold that:

$$\begin{aligned} \frac{\beta_j}{e_j} &= \frac{\beta_i}{e_i} \\ \bar{\rho}_0 &= \bar{\rho}_1 = \rho_{eq} \\ \bar{\tau}_0 &= \bar{\tau}_1 = \frac{\beta_j}{e_j}\rho_{eq} \\ \tau_{r,j} &= \beta_j(\frac{1}{e_j} - \frac{1}{b} - \frac{1}{f})\rho_{eq} \end{aligned} \tag{6-17}$$

These constraints on the parameters can be quite restrictive. This will be looked at more closely when studying the robustness of the system.

In a similar way as in Section 6-1-4, the expression in Equation 6-16 can be rewritten as a system for the whole network that only depends on the states at the first station $x_1(k)$ (together with $x(k-1)$ and $u(k)$). This will have the form: $x(k) = A_{1,eq}x(k-1) + B_{1,eq}u(k) + A_{11,eq}x_1(k) + B_{4,eq}$.

For $J = 3$ the matrices will have the following structure:

$$A_{1,eq} = \left[\begin{array}{cccccc|cccccc} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\alpha_{1,2} & 0 & \alpha_{3,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{4,2} & 0 & \alpha_{3,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b\alpha_{4,2} & 0 & \alpha_{1,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \alpha_{4,2} & 0 & \alpha_{3,2} & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{1,3}\alpha_{4,2} + b\alpha_{2,3}\alpha_{4,2} & 0 & \alpha_{1,3}\alpha_{3,2} + \alpha_{2,3}\alpha_{1,2} & 0 & 0 & 0 & -\alpha_{1,3} & 0 & \alpha_{3,3} \\ 0 & 0 & 0 & \alpha_{4,2} & 0 & \alpha_{3,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{1,3}\alpha_{4,2} + b\alpha_{2,3}\alpha_{4,2} & 0 & \alpha_{1,3}\alpha_{3,2} + \alpha_{2,3}\alpha_{1,2} & 0 & 0 & 0 & \alpha_{4,3} & 0 & \alpha_{3,3} \\ 0 & 0 & 0 & -b\alpha_{4,3}\alpha_{4,2} + b^2(\alpha_{2,3} - \gamma_{2,3})\alpha_{4,2} & 0 & -b\alpha_{4,3}\alpha_{3,2} + b(\alpha_{2,3} - \gamma_{2,3})\alpha_{1,2} & 0 & 0 & 0 & b\alpha_{4,3} & 0 & \alpha_{1,3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$B_{1,eq} = \left[\begin{array}{cc} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ e_2 & 0 \\ 0 & 0 \\ \hline 1 & 0 \\ \alpha_{1,3}+\alpha_{2,3}e_2 & 1 \\ 1 & 0 \\ \alpha_{1,3}+\alpha_{2,3}e_2 & 1 \\ -b\alpha_{4,3}+b(\alpha_{2,3}-\gamma_{2,3})e_2 & e_3 \\ 0 & 0 \end{array}\right] \quad B_{4,eq} = \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ 0 & \alpha_{1,2} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \alpha_{1,2} & 0 & 0 \\ 0 & -b\alpha_{4,2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & \alpha_{1,2} & 1 & 0 \\ 0 & \alpha_{1,3}\alpha_{1,2}-b\alpha_{2,3}\alpha_{4,2} & \alpha_{1,3} & 0 \\ 0 & \alpha_{1,2} & 1 & 0 \\ 0 & \alpha_{1,3}\alpha_{1,2}-b\alpha_{2,3}\alpha_{4,2} & \alpha_{1,3} & 0 \\ 0 & -b\alpha_{4,3}\alpha_{1,2}-b^2(\alpha_{2,3}-\gamma_{2,3})\alpha_{4,2} & -b\alpha_{4,3} & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} \rho_{max} \\ \tau_{r,2} \\ \tau_{r,3} \\ \tau_H \end{bmatrix}$$

$$A_{11,eq} = \left[\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{1,2} & \alpha_{2,2} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{1,2} & \alpha_{2,2} & 0 \\ 0 & 0 & 0 & -b\alpha_{4,2} & b(\alpha_{2,2}-\gamma_{2,2}) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \alpha_{1,2} & \alpha_{2,2} & 0 \\ 0 & 0 & 0 & \alpha_{1,3}\alpha_{1,2}-b\alpha_{2,3}\alpha_{4,2} & \alpha_{1,3}\alpha_{2,2}+b\alpha_{2,3}(\alpha_{2,2}-\gamma_{2,2}) & 0 \\ 0 & 0 & 0 & \alpha_{1,2} & \alpha_{2,2} & 0 \\ 0 & 0 & 0 & \alpha_{1,3}\alpha_{1,2}-b\alpha_{2,3}\alpha_{4,2} & \alpha_{1,3}\alpha_{2,2}+b\alpha_{2,3}(\alpha_{2,2}-\gamma_{2,2}) & 0 \\ 0 & 0 & 0 & -b\alpha_{4,3}\alpha_{1,2}-b^2(\alpha_{2,3}-\gamma_{2,3})\alpha_{4,2} & -b\alpha_{4,3}\alpha_{2,2}+b^2(\alpha_{2,3}-\gamma_{2,3})(\alpha_{2,2}-\gamma_{2,2}) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{6-18}$$

The linear version of this system is:

$$x(k) = A_{1,eq}x(k-1) + B_{1,eq}u(k)$$

We should choose $Q, R \succ 0$ and (A, B) stabilizable (which is the case if e_2 & $b(\alpha_{2,3} - \gamma_{2,3})e_2$ or e_3 are not zero). Then the state feedback gain K ($u = Kx$) should be chosen as the infinite horizon LQ gain. Now, P is the solution to the Lyapunov equation in Equation 3-6. Then the terminal cost is equal to $V_f(x) = \frac{1}{2}(x(k) - x_{eq}(k))^T P(x(k) - x_{eq}(k))$.

6-2-3 Determining the terminal constraint set

In the terminal set for the Urban railway line there is one condition that needs to hold: $\rho_j(k) \leq \rho_{max}$. As the terminal set is positive definite, it should hold that if the system starts in $\mathbb{X}_f$, it should stay in $\mathbb{X}_f$. This means that the terminal set should only consist of positive state values ($x(k) \geq 0$), because both a (difference in a) time instance and an amount of people cannot be negative.

So this means that the terminal set can be summarized as

$$x(k) \geq 0 \quad \bigcap \quad \begin{bmatrix} \rho_2(k) \\ \rho_3(k) \end{bmatrix} \leq \rho_{max}$$

However, this is not a very robust choice. If there is some parameter uncertainty, $\begin{bmatrix} \rho_2(k) \\ \rho_3(k) \end{bmatrix} \leq \rho_{max}$ will lead to instability rather easily. This is why some caution is built into the terminal set. This means that the final terminal set is:

$$x(k) \geq 0 \quad \bigcap \quad \begin{bmatrix} \rho_2(k) \\ \rho_3(k) \end{bmatrix} \leq \mu\rho_{max} \tag{6-19}$$

Where $\mu \leq 1$. How to choose μ will be described in Section 6-4-4

6-3 Assumptions and Theorems

This section will prove that the assumptions in Section 3-2-1 hold for the urban railway line. In this way in can be ensured that the reasoning from Section 3-2-2 holds and that using the 2-norm cost function will result in a stable MPC controller.

The functions f (which is the state description), ℓ and V_f are continuous. When the conditions in Equation 6-17 hold, $f(x_{eq}, u_{eq}) = x_{eq}$. Next to that, $\ell(x_{eq}, u_{eq}) = 0$ and $V_f(x_{eq}) = 0$. This means that Assumption 3-2.1 holds.

$\mathbb{X} = [0,\infty)\times[0,\infty)\times[0,\infty)\times[0,\infty)\times[0,\rho_{max}]\times[0,\infty)$. $\mathbb{U} = [0,\infty)\times[0,\infty)$ and it is coercive. Both these sets contain the equilibrium. The terminal set $\mathbb{X}_f$ in Equation 6-19 is a subset of $\mathbb{X}$. For the set $\mathbb{Z}$ it holds that $\mathbb{Z} \subseteq \mathbb{X}\times\mathbb{U}$. This set has the shape of a polyhedron, as it is bounded by the constraint in Equation 6-13. This means $\mathbb{Z}$ is closed and Assumption 3-2.2 holds.

Lastly, we take a look at the conditions in Assumption 3-2.3. It needs to hold for all $x \in \mathbb{X}_f$ that

$$V_f(f(x,u)) \leq V_f(x) - \ell(x,u)$$

This can also be written as $V_f(A_K(x-x_{eq})) \leq V_f(x-x_{eq}) - \frac{1}{2}(x-x_{eq})^T Q_K (x-x_{eq})$. Because it holds that $A_K^T P A_K = P - \mu Q_K$, $\mu > 1$ (Equation 3-6), the equation can be rewritten as

$$\begin{aligned} V_f(A_K(x-x_{eq})) &= \frac{1}{2}(x-x_{eq})^T P(x-x_{eq}) - \frac{1}{2}\mu(x-x_{eq})^T Q_K(x-x_{eq}) \\ &\leq \frac{1}{2}(x-x_{eq})^T P(x-x_{eq}) - \frac{1}{2}(x-x_{eq})^T Q_K(x-x_{eq}) \\ &= V_f(x-x_{eq}) - \ell(x-x_{eq}, A_K(x-x_{eq})) \end{aligned} \tag{6-20}$$

Next to that, the second part of the assumption holds because $\ell(x,u) \geq q||x||^2 + r||u||^2 \geq q||x||^2$ and $V_f(x) \leq p||x||^2$ for $q = \min eig(Q)$ and $p = \max eig(P)$.

This means that all assumptions from Section 3-2-1 are satisfied. Since $\mathbb{X}_f$ contains the equilibrium in its interior, Theorem 3-2.1 is valid. Furthermore, the hypotheses in Theorem 3-2.2 are satisfied as well. So the equilibrium is exponentially stable in $\mathcal{X}_N$.

6-4 Results

This section shows the results for the MIQP problem. The parameter values that were used for this optimization can be found in Appendix A-3. The Matlab scripts can be found in Appendix B. The simulation uses a homogeneous system. So when there are no disturbances, $\beta_j = \beta_i = \beta$ and $e_j = e_i = e$. In this way the α and γ variables (Table 6-2) will be equal for each station as well. The results are gathered for 100 trains ($k = 100$), but only the part of the graphs that gives information will be depicted. The prediction horizon N_p is equal to 5 trains.

First the results will be analyzed on its own. After this it will be analyzed how the system behaves in the presence of disturbances and parametric uncertainty.

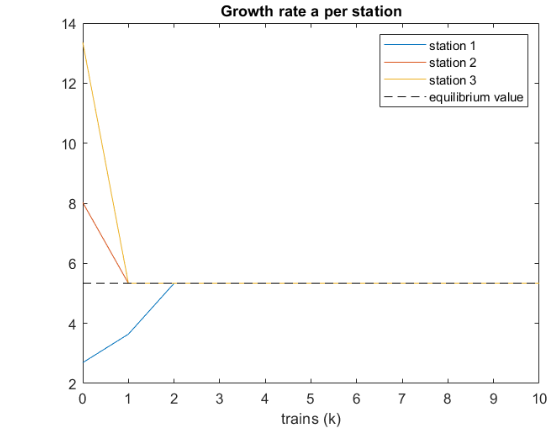
6-4-1 Analysis unperturbed system

When optimizing the MIQP problem in Equation 6-13, the results in Figure 6-1 are acquired.

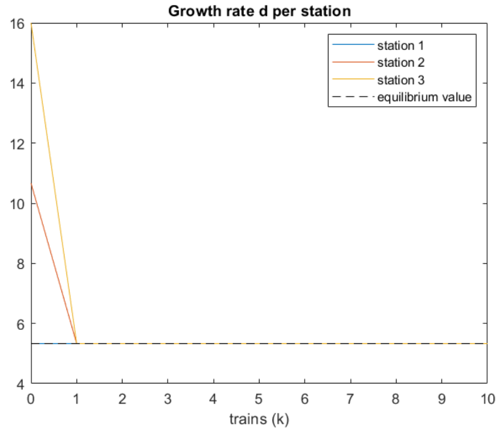
Here the following stage cost matrices are used:

$$Q_j = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad R_j = 3$$

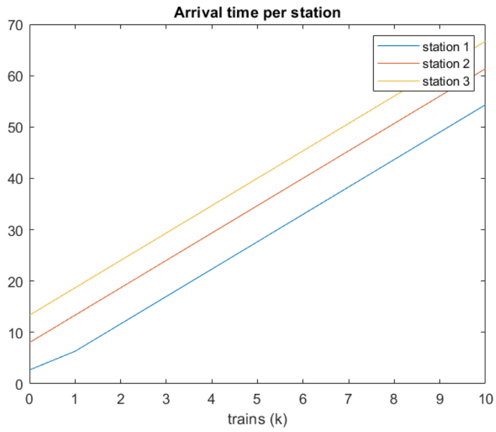
The first thing to notice is that the growth rate in $a_j(k)$ and $d_j(k)$ both go to the constant value that was expected by the equilibrium value calculation (Figures 6-1a and 6-1b, respectively). Next to that, the amount of people in the train (Figure 6-1e) stays at the assigned equilibrium value as well (aside from a small computational error). Furthermore, the amount of people left on the platform $\sigma_j(k)$ (Figure 6-1f) is indeed zero. So without disturbances or parameter uncertainty, the system behaves as expected.



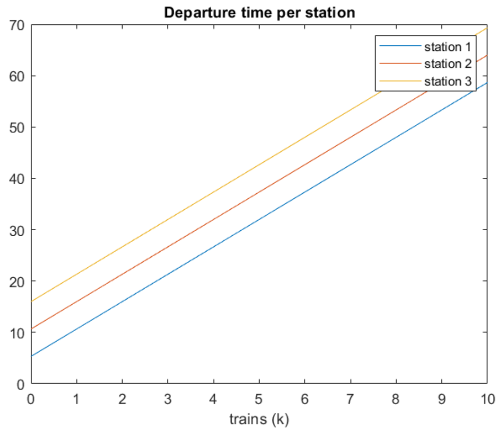
(a) Results for $a_j(k) - a_j(k-1)$



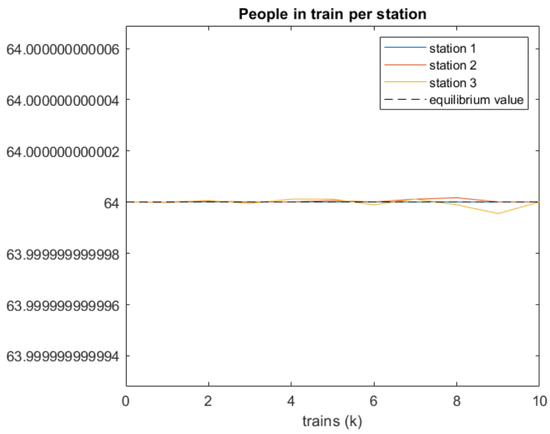
(b) Results for $d_j(k) - d_j(k-1)$



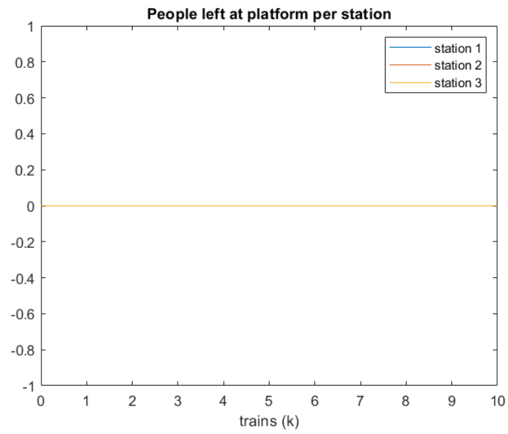
(c) Results for $a_j(k)$



(d) Results for $d_j(k)$



(e) Results for $\rho_j(k)$



(f) Results for $\sigma_j(k)$

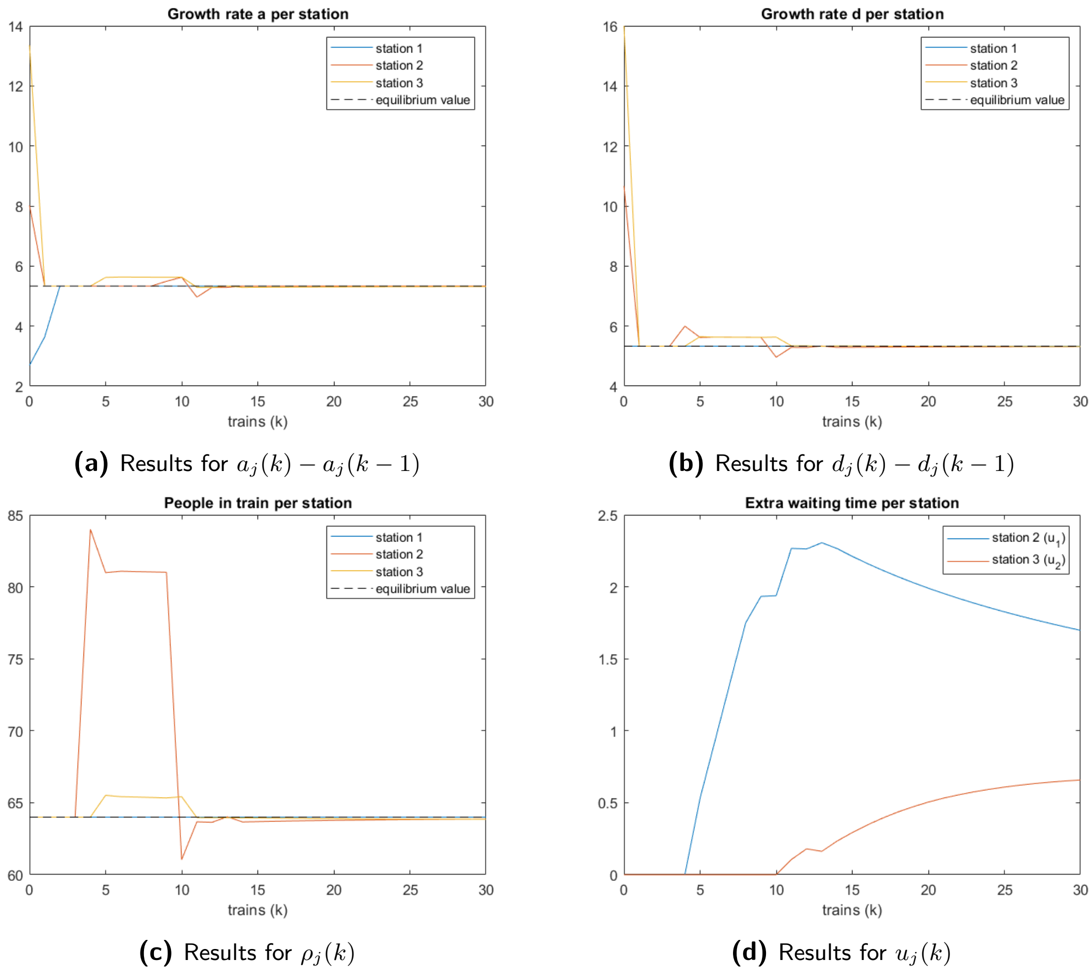
Figure 6-1: Results MPC state vector with 6 states

6-4-2 Stability

The poles of $A_{1,eq}$ when $J = 3$ are $\alpha_{4,2}$, $\alpha_{4,3}$ and 10 times 0. This means that the linearized plant is stable if $|\alpha_{4,2}| < 1$ and $|\alpha_{4,3}| < 1$. Since $\alpha_4 = \frac{-e}{b-e}$, the plant is stable when $b > 2e$.

To show that the controlled closed loop system is stable as well, we will take a look at the disturbance rejection properties of the system. Two different output disturbances will be added to the system. These disturbances represent the following two situations:

1. There are **more people on the station** than expected when the train arrives. This can be represented as a pulse of size p_ρ on $\rho_j(k)$ and a pulse of size $\frac{1}{b}p_\rho$ on $d_j(k)$ and $d_j(k) - d_j(k-1)$ (if $\rho_j(k) + p_\rho \leq \rho_{max}$).
2. There is an animal on the train track, so the **train needs to lower its speed**. This means the train arrives at the station later than expected, that there are more people that need to enter the train and the train will leave later as well. So this can be represented as a pulse of size p_a on $a_j(k)$ and $a_j(k) - a_j(k-1)$, a pulse of size $\alpha_1 \cdot p_a$ on $d_j(k)$ and $d_j(k) - d_j(k-1)$ and a pulse of size $b\alpha_4 \cdot p_a$ on $\rho_j(k)$.



(a) Results for $a_j(k) - a_j(k-1)$

(b) Results for $d_j(k) - d_j(k-1)$

(c) Results for $\rho_j(k)$

(d) Results for $u_j(k)$

Figure 6-2: Case 1: Results MPC output disturbance p_ρ

For both cases the disturbance is present at station 2 and trains 5 up until 10 are influenced by it. After that, the system goes back to the normal (disturbance-free) situation. In case 1 the pulses represent the situation when all extra people can fit in the train. Otherwise $p_{\rho,real} = p_{max} - p_{eq}$ and there would also be a pulse on $\sigma_j(k)$ of size $p_\rho - p_{\rho,real}$. The design of the system is depicted in Figure 6-3.

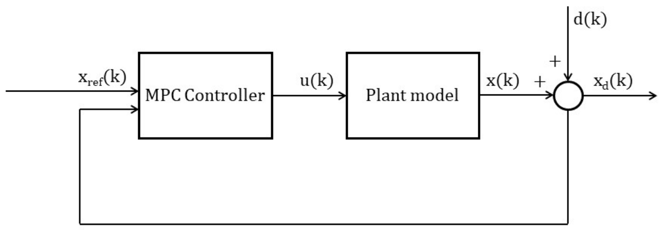
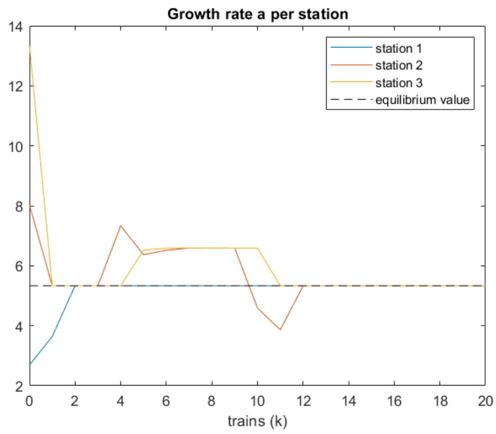
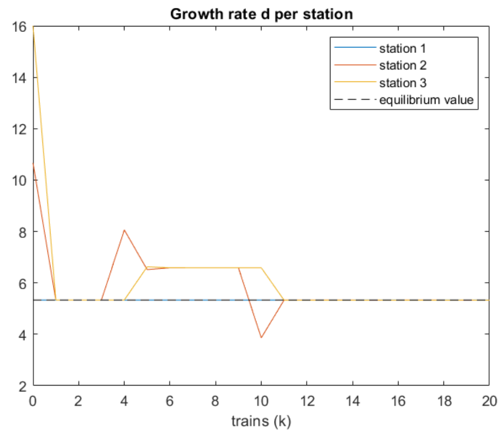


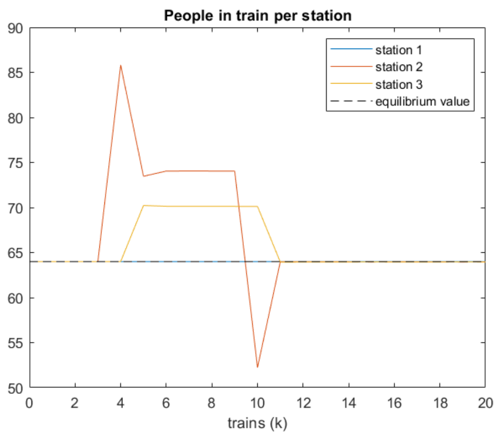
Figure 6-3: Setup output disturbance MPC control



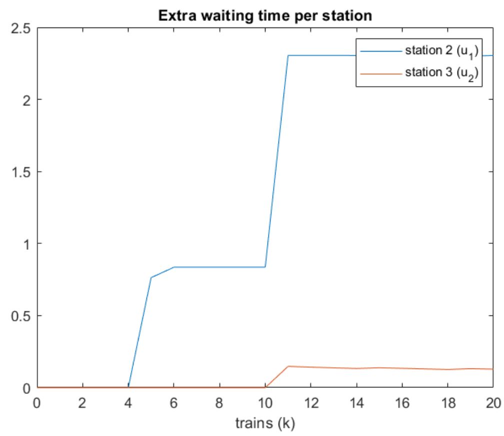
(a) Results for $a_j(k) - a_j(k-1)$



(b) Results for $d_j(k) - d_j(k-1)$



(c) Results for $\rho_j(k)$



(d) Results for $u_j(k)$

Figure 6-4: Case 2: Results MPC output disturbance p_a

For case 1 and $p_\rho = 20$, the results are summarized in Figure 6-2. Here the results for $a_j(k)$ and $d_j(k)$ are skipped, since the disturbances are not very visible. The results for $\sigma_j(k)$ are skipped as well, because it is still equal to zero everywhere. From Figures 6-2a-6-2c, it is clear that, indeed, after train 10 the disturbance is rejected in only a few events. After this, the value is again at the equilibrium value. The input does stay above zero after the disturbance has ended, because the trains that did not have to deal with extra people did not get delayed and thus have to wait for the delayed trains to be able to leave station 2. The extra waiting time is already a lot smaller at station 3. Next to that, the amount of people in the train immediately after the disruption ended is lower than the equilibrium, because the train before that left later than usual. So there were less people on the station when this train arrived.

For case 2 and $p_a = 2$, the results are summarized in Figure 6-4. These graphs show a similar picture to the graphs in Figure 6-2. Again, the disturbance is rejected only a few trains after the pulse disturbance has ended and $u_2(k)$ stays large after the disruption has ended.

So the stability of the system is supported by these two cases of output disturbance rejection.

6-4-3 Uncertainty and robustness

This section checks whether the system has robust stability. For the system to be robustly stable, the system needs to remain stable under the uncertainty that can be present in the system.[16] This uncertainty can be caused by parametric uncertainty or unmodelled/neglected dynamics uncertainty. In this section we will assume that there is only parametric uncertainty present. For this situation, the uncertain parameters are assumed to be bounded within some region $[\alpha_{min}, \alpha_{max}]$.

In the urban railway line, the parametric uncertainty mainly lays in the parameters that consider passengers. This means that the amount of passengers that enter the station per minute e_j, the fraction of people leaving the train β_j and the initial amount of people ρ_0 and ρ_1 are subject to uncertainty. Next to this, the running time $\tau_{r,j}$ has an uncertain factor, as the train might need to stop or slow down on its way to the next station.

In this analysis, it is assumed that the parametric uncertainty in these parameters is equal to:

- $[e - 2, e + 2]$
- $[\beta - 0.2, \beta + 0.2]$
- $[\bar{\rho}_0 - 20, \bar{\rho}_0 + 20]$
- $[\bar{\rho}_1 - 20, \bar{\rho}_1 + 20]$
- $[\tau_r - 2, \tau_r + 2]$

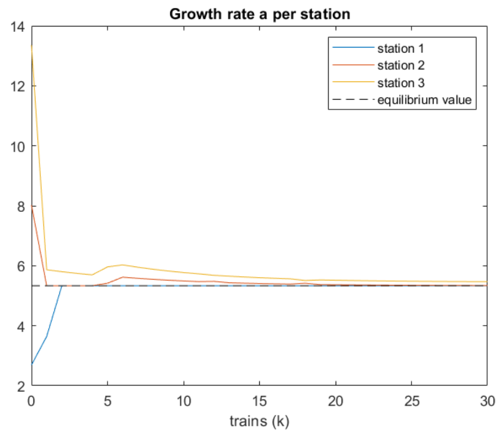
To show the consequences of the parametric uncertainty, we will start by looking at the same two situations as in the previous subsection. After this the consequences of all parametric uncertainties will be shown by analyzing their system description.

Now these situations are represented by an alteration in the parameters:

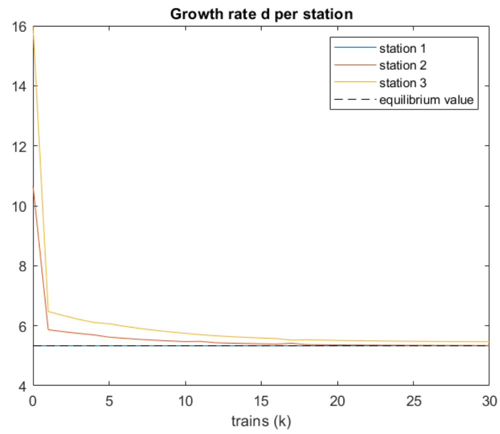
1. There are **more people** than expected **on the station** when the train arrives. This is represented as $e_2 = e + 2$.
2. There is an animal on the train track, so the **train needs to lower its speed**. This is represented as $\tau_{r,2} = \tau_r + 2$.

The parametric uncertainty holds for all stations.

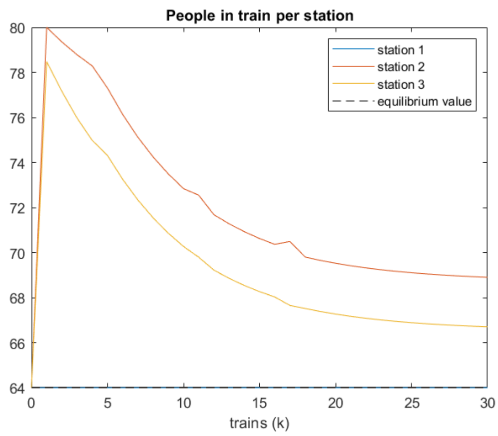
In Figure 6-5 the first situation is depicted. Even when the actual value of e_2 is 2 people/min higher, the MPC controller based on the original system is able to steer the growth rates back to the right equilibrium values. The amount of people in the train is higher than the original equilibrium value, but they do both go to a stable equilibrium. This shows that the system has robust stability for the maximal parameter uncertainty of e_2.



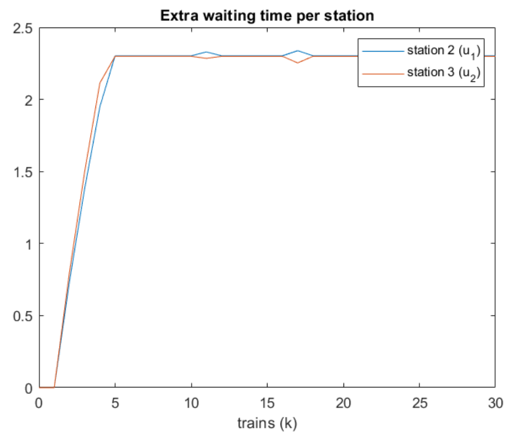
(a) Results for $a_j(k) - a_j(k-1)$



(b) Results for $d_j(k) - d_j(k-1)$



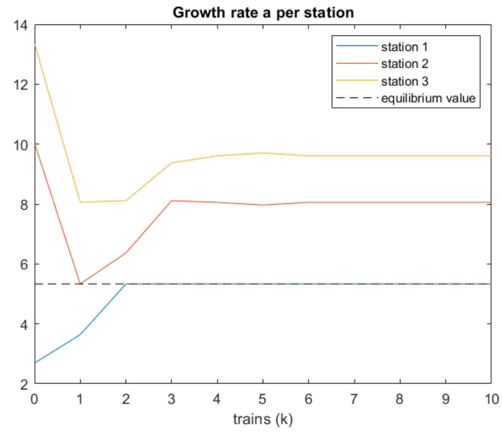
(c) Results for $\rho_j(k)$



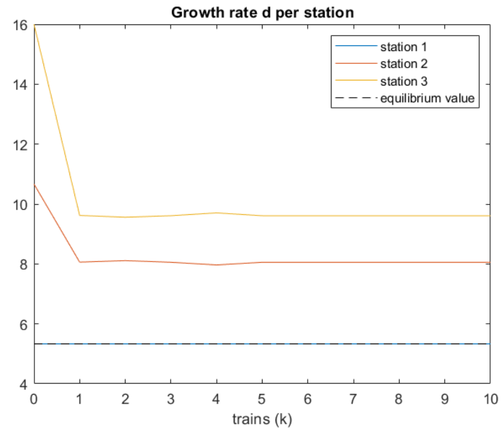
(d) Results for $u_j(k)$

Figure 6-5: Case 1: Results MPC with parameter uncertainty in e_2

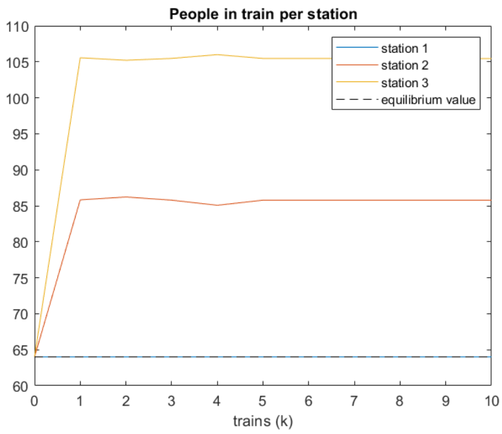
For situation 2, all states and the input will return to a stable value again. However, for the states this value is not constant to the original equilibrium value. This makes sense, since a larger $\tau_{r,2}$ means that there is more time for people to arrive at the station and it takes more time until all those people will have entered the train ($d_2(k) - a_2(k)$ is larger). The headway time between the train does stay the same. For this headway time stay respected during the simulation, we see that with some delay the values for the growth rate of a (Figure 6-6a) go to the same (higher) value of the growth rate of d (Figure 6-6b). As this shift does not mean that the system loses stability, robust stability is satisfied when the maximal running time uncertainty is present.



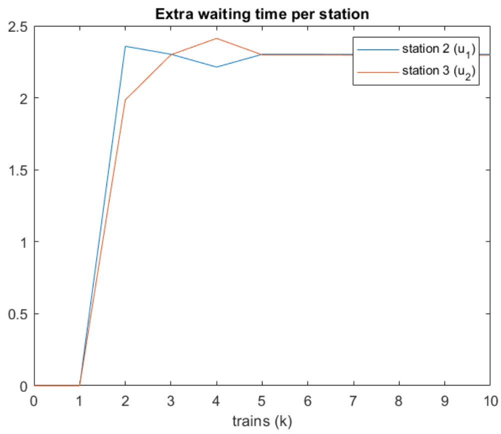
(a) Results for $a_j(k) - a_j(k-1)$



(b) Results for $d_j(k) - d_j(k-1)$



(c) Results for $\rho_j(k)$



(d) Results for $u_j(k)$

Figure 6-6: Case 2: Results MPC with parameter uncertainty in $\tau_{r,2}$

Summary influence parameter uncertainty

Based on the system in Equation 6-18, the influence of the uncertainty set can be determined. Here, the values in the situation where there is no uncertainty are depicted as e, β, α_1, etc, because in the homogeneous system these values are equal for each station. For convenience we will abbreviate $a_j(k) - a_j(k-1)$ as $aa_j(k)$ and $d_j(k) - d_j(k-1)$ as $dd_j(k)$. Next to that, we leave out the $\sigma_j(k)$ terms, as these are always equal to zero.

The influence of uncertainty in τ_r can be characterized as:

$$\begin{aligned}
\Delta aa_2(k) &= \Delta\tau_{r,2} - \Delta a_2(k-1) \\
\Delta dd_2(k) &= -\alpha_1\Delta d_2(k-1) + \alpha_1\Delta\tau_{r,2} + \Delta u_2 \\
\Delta\rho_2(k) &= e_2\Delta d_2(k-1) \\
\Delta aa_3(k) &= \Delta dd_2(k) \\
\Delta dd_3(k) &= (\alpha_1 + \alpha_2 e_2)\Delta dd_2(k) + \alpha_1\Delta d_2(k-1) - \alpha_1\Delta d_3(k-1) + \Delta u_3 \\
\Delta\rho_3(k) &= b(-\alpha_4 + e_2(\alpha_3 - \gamma_2))\Delta dd_2(k) + e_3(\alpha_1\Delta d_2(k-1) - \alpha_1(3)\Delta d_3(k-1) - \Delta u_3)
\end{aligned}$$

where Δ represents the deviation for the original equilibrium value caused by the uncertainty.

The first thing to notice is that $\tau_{r,2}$ has an influence on the initial value of $a_2(k)$ ($a_j(0) = \max((j-1)\bar{\tau}_0 + \tau_{r,j}\ ,\ \tau_H)$). However, when $\tau_{r,2}$ is lowered, this influence can be partially compensated at the beginning by τ_H. So then $\Delta aa_2(1)$ will be negative. If $\tau_{r,2}$ is heightened $\Delta aa_2(k) = \Delta\tau_{r,2} - \Delta a_2(k-1)$ will be zero. However, the value of $aa_2(k)$ needs to respect the constraints caused by $a_j(k) = \max(d_{j-1}(k) + \tau_{r,j}\ ,\ d_j(k-1) + \tau_H)$. So this is why the value of $aa_2(k)$ gradually goes to the same value as $dd_2(k)$. A similar reasoning holds for $aa_3(k)$. At first it is equal to $dd_2(k)$, but as it cannot stay at this value while satisfying the constraints, it gradually transitions to the value of $dd_3(k)$ as well.

The extra waiting time at station 2 causes $\Delta dd_2(k)$ to stay at a constant value. This constant value is zero when $\tau_{r,2}$ is lowered and larger than zero when $\tau_{r,2}$ is heightened. u_3 is able to compensate the $\Delta d_2(k-1)$ and $\Delta d_3(k-1)$ term in $\Delta dd_3(k)$ and $\Delta\rho_3(k)$. So the values of $\Delta dd_3(k)$ and $\Delta\rho_3(k)$ in the equilibrium will be equal to a multiple of $\Delta dd_2(k)$. So these will be zero when $\tau_{r,2}$ is lowered and larger than zero when $\tau_{r,2}$ is heightened.

In the end, it can be concluded that the uncertainty in $\tau_{r,2}$ does not lead to instability in the system.

In a similar way influence of uncertainty in β_2 can be described as:

$$\begin{aligned}
&\Delta aa_2(k) = 0 \\
&\Delta dd_2(k) = -\alpha_1\Delta d_2(k-1) + \frac{\alpha_1}{f}\bar{\rho}_1\Delta\beta_2 + \Delta u_2 \\
&\Delta\rho_2(k) = e\Delta dd_2(k) - \bar{\rho}_1\Delta\beta_2 \\
&\Delta aa_3(k) = \Delta dd_2(k) \\
&\Delta dd_3(k) = (\alpha_1 - e\alpha_2)\Delta dd_2(k) - \alpha_2\bar{\rho}_1\Delta\beta_2 + \alpha_1(\Delta d_2(k-1) - \Delta d_3(k-1)) + \Delta u_3 \\
&\Delta\rho_3(k) = (-b\alpha_4 + b(\alpha_2 - \gamma_2))\Delta dd_2(k) - b(\alpha_2 - \gamma_2)\bar{\rho}_1\Delta\beta_2 + e(\alpha_1\Delta d_2(k-1) - \alpha_1\Delta d_3(k-1)) + e\Delta u_3
\end{aligned}$$

The results for $\beta_2 = \beta_2 - 0.2$ can be found in Appendix A-4. Altering β_2 does not influence the initial values of the system. When β_2 is lowered, the system is able to return to the original growth rates by itself without the input needing to interfere. However, the amount of people in the train will end up in a higher equilibrium value, because there a negative input would have been needed. This is not possible in the current system description.

When β_2 is heightened it is harder to compensate, because if the train waits longer at the second station, there will be more people at the third station as well. So the controller will try to find a middle ground, but is unable to fully solve all of the deviations in the equilibrium values.

The influence of uncertainty e is harder to quantify, because e is present in all α variables and there it can be part of the denominator as well as the numerator. This is why we will show the actual equilibrium values rather than the change in these values. In general the equilibrium values can be described as:

$$\begin{aligned}
aa_2(k) &= -a_2(k-1) + \tau_r + (k+1)\tau_1 \\
dd_2(k) &= \frac{b}{b-e_2}(\tau_r + \frac{\beta}{f}\bar{\rho}_1 + (k+1)\bar{\tau}_1 - d_2(k-1)) + u_2 \\
\rho_2(k) &= e_2 dd_2(k) + (1-\beta)\bar{\rho}_1 \\
aa_3(k) &= (a_2(k) - a_2(k-1))_{eq} + u_3 \\
dd_3(k) &= (1 + e_2\frac{\beta}{f})\alpha_1 dd_2(k) + \alpha_1(d_2(k-1) - d_3(k-1) + \tau_r) + u_3 \\
\rho_3(k) &= edd_3(k) + (1-\beta)\rho_2(k) + e\alpha_2(1-\beta)\rho_1
\end{aligned}$$

An example of the heightened e_2 value was already presented in Figure 6-5d. For an example of the lowered e_2 value, one can take a look at Appendix A-5. The first thing to notice is that $aa_2(k)$ is calculated in the normal way. So its description is not influenced by uncertainty. Next to that, in $dd_2(k)$ the influence of $(k+1)\bar{\tau}_1$ and $-d_2(k-1)$ counteract each other, so this has a stabilizing effect. If e_2 is lowered, the system can stabilize itself without input, because $\frac{b}{b-e_2}$ will be lowered so the effect will die out. If e_2 is heightened $\frac{b}{b-e_2}$ will grow, as well as $d_2(k-1)$. So this effect needs to be counteracted with a positive $u_2(k)$. This is what is shown in figure A-2d in Appendix A-5. In this Appendix you can also see that, even though $dd_2(k)$ and $dd_3(k)$ can (almost) be brought back to the original equilibrium, $\rho_3(k)$ and $\rho_2(k)$ will not. This is because of the $\bar{\rho}_1$ terms in those descriptions.

There is no influence of the uncertainty in $\bar{\rho}_0$ on the location of the equilibria. Altering $\bar{\rho}_0$ only changes the location of the initial value.

Uncertainty in $\bar{\rho}_1$ can influence the equilibria. Its effect can be described as:

$$\begin{aligned}
\Delta aa_2(k) &= \Delta\alpha_2(k-1) \\
\Delta dd_2(k) &= -\alpha_1\Delta d_2(k-1) + \Delta u_2 + \alpha_2\Delta\bar{\rho}_1 \\
\Delta\rho_2(k) &= e\Delta dd_2(k) + (1-\beta)\Delta\bar{\rho}_1 \\
\Delta aa_3(k) &= \Delta dd_2(k) \\
\Delta dd_3(k) &= (\alpha_1 + \alpha_2 e_2)\Delta dd_2(k) + \alpha_2(1-\beta)\Delta\bar{\rho}_1 + \alpha_1\Delta d_2(k-1) - \alpha_1\Delta d_3(k-1) + \Delta u_3 \\
\Delta\rho_3(k) &= b(\alpha_4 + e(\alpha_2 - \gamma_2))\Delta dd_2(k) + b(\alpha_2 - \gamma_2)(1-\beta)\Delta\bar{\rho}_1 + e(\alpha_1\Delta d_2(k-1) - \alpha_1\Delta d_3(k-1) + \Delta u_3)
\end{aligned}$$

An example for $\bar{\rho}_1 + 20$ can be found in Appendix A-6. If there are more people in the train at station 1 this will automatically be compensated over time, as a larger portion of those people will get out at later stations as well. If there are fewer people in the train at the beginning we will see graphs that are very similar in shape to the situation where β_2 is heightened. This makes sense since the descriptions for the deviation of those variables is very similar.

As none of the parameter uncertainties in this section will lead to instability, it can be concluded that the system has robust stability for uncertainties in the parameter for station 2.

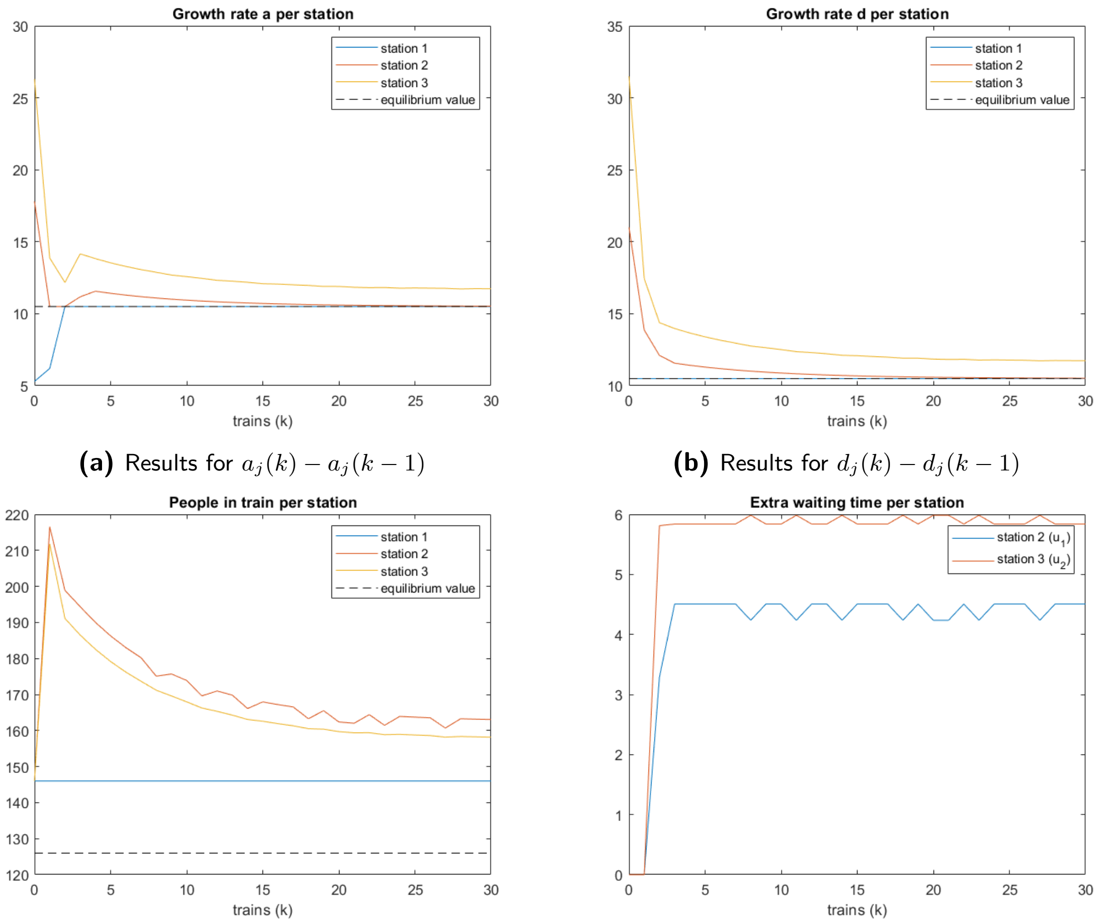
6-4-4 Sensitivity Terminal Set

In Section 6-2-3, it was already pointed out that a terminal set with the constraint $\rho_j(k) \leq \rho_{max}$ would not be a robust choice. When the system is at $\rho_j(k) = \rho_{max}$, uncertainty in the parameters can very easily lead to instability. In this section it will be shown which buffer needs to be build into the description of the terminal set to ensure that the terminal set is indeed positive definite for all systems in the uncertainty set. So if we apply all worst-case uncertainties on the parameters of station 2 at the same time, what is the maximal value of $\rho_j(k)$ that will ensure stability.

The peak that $\rho_2(k)$ and $\rho_3(k)$ reach under the uncertainty can be calculated as:

$$\rho_2(k) = (\frac{be_2\beta_2}{(b-e_2)f} + 1 - \beta_2)\bar{\rho}_1 + \frac{be_2}{b-e_2}\tau_{r,2}$$

$$\rho_3(k) = b\alpha_4\beta(\frac{1}{b} + \frac{1}{f})\rho_{eq} - (b\alpha_4\alpha_{1,2} + b^2(\alpha_2 - \gamma_2)\alpha_{4,2})(\tau_{r,2} + \frac{\beta_2}{f}\bar{\rho}_1) + b(\alpha_2 - \gamma_2)(1 - \beta_2)\bar{\rho}_1$$



(a) Results for $a_j(k) - a_j(k-1)$

(b) Results for $d_j(k) - d_j(k-1)$

(c) Results for $\rho_j(k)$

(d) Results for $u_j(k)$

Figure 6-7: Results MPC sensitivity of terminal set

So we need to make sure that the choice of ρ_{eq} ensures that both $\rho_2(k) \leq \rho_{max}$ and $\rho_3(k) \leq$

ρ_{max}. If we fill in $\bar{\rho}_1 = \rho_{eq} + 20$ and $\tau_{r,2} = \beta(\frac{1}{e} - \frac{1}{b} - \frac{1}{f})\rho_{eq} + 2$, we get the following conditions:

$$\rho_{eq} \leq \frac{(b-e_2)f\rho_{max} - 20(be_2\beta_2 + (1-\beta_2)(b-e_2)f) - 2be_2 f}{be_2(\beta_2 - \beta) + (1-\beta_2)(b-e_2)f + \frac{e_2}{e_3}(b-e)\beta f}$$

$$\rho_{eq} \leq \frac{\rho_{max} - 20\big(-b\alpha_4\alpha_{2,2} + b^2(\alpha_2-\gamma_2)(\alpha_{2,2}-\gamma_{2,2})\big) + 2(b\alpha_4\alpha_{1,2} + b^2(\alpha_2-\gamma_2)\alpha_{4,2})}{b\alpha_4(\beta(\frac{1}{b}+\frac{1}{f}) - \alpha_{2,2} - \alpha_{1,2}\beta(\frac{1}{e}-\frac{1}{b}-\frac{1}{f})) + b^2(\alpha_2-\gamma_2)(\alpha_{2,2}-\gamma_{2,2}-\alpha_{4,2}\beta(\frac{1}{e}-\frac{1}{b}-\frac{1}{f}))}$$

With the parameter values in Appendix A-3 and the uncertainty from Section 6-4-3, this comes down to $\rho_{eq} \leq 126$ and $\rho_{eq} \leq 130$. So if the terminal set is chosen as

$$x(k) \geq 0 \quad \bigcap \quad \begin{bmatrix}\rho_2(k)\\ \rho_3(k)\end{bmatrix} \leq 126 \tag{6-21}$$

the terminal set will be a robust choice for uncertainty in the parameters of station 2. The results for $\rho_{eq} = 126$ and all worst-case uncertainties combined is depicted in Figure 6-7. One can see that indeed the peak at station 2 is just slightly below ρ_{max} (=217) and a larger value would lead to instability.

6-5 Conclusion

This chapter implemented the procedure that has been composed in this master thesis on a real life example: an urban railway line. The urban railway line is a partially additive homogeneous DE MMPS system. The input that was used to control the system was an extra waiting time at the station. The system has two time variables and two quantitative variables. So for the time variables the difference in these states was added to the state vector.

In several steps the control on the MMPS system could be recast as an MIQP problem. Here, the optimization variables were captured in the free variables $\hat{V}(k)$, which consists of the auxiliary variables $\hat{\delta}(k)$ and $\hat{z}(k)$ and the input $\hat{u}(k)$.

After the problem was fully described, it was confirmed that the assumptions and theorems that were established in Chapter 3 indeed hold for the urban railway line. Next, the system was simulated for several situations. For this simulation a homogeneous system was used. So the parameters are equal for each station. It was concluded that the system is stable when $b > 2e$. Next to that, the controlled system is able to successfully reject multiple disturbances.

When the equilibrium values were determined, several conditions on the parameters were composed. As these conditions were quite restrictive, it made sense to study what the effect would be of these values were being altered. This was done by studying the robust stability of the system under parameter uncertainty. Here, it could be concluded that the system did have robust stability. It was not always possible to return to the original equilibrium values, though. In some situations it might help if the input is also allowed to be negative. So the train could leave earlier than when it is full or all people have gotten in. This would mean that, the input $u_j(k)$ should also be added in the second term of the description of $d_j(k)$. So the description for the departure time at a station would become:

$$d_j(k) = \min\left(\alpha_1 a_j(k) + \alpha_2\rho_{j-1}(k) + \alpha_3\sigma_j(k-1) + \alpha_4 d_j(k-1) + u_j(k),\ \gamma_0 + \gamma_1 a_j(k) + \gamma_2\rho_{j-1}(k) + u_j(k)\right)$$

Furthermore, the terminal set for the urban railway line was computed. This is the set of state values where the states are all bigger than zero and the amount of people in the train is smaller than ρ_{max}. However, this is not a robust choice. When the uncertainty set for the parameter at station 2 are taken into account, the second constraints should be altered. The new condition that should hold for the terminal set, when using the parameters in Appendix A-3, is $\rho_j(k) \leq 126$.

So this example supports that the proposed procedure on how to design an MPC controller for a (partially) additive homogenous DE MMPS system, is indeed able to guarantee a stable outcome.

Chapter 7

Conclusions and Contributions

This chapter will summarize the main conclusions that can be drawn from the research in this master thesis. Next to that, this chapter will also give concise overview of the contribution of this master thesis to the field of systems and control.

7-1 Conclusions

In this section an overview is given of the main conclusions in this master thesis report. This will be done based on the research questions that were discussed in Section 1-2-1. First the subquestions will be answered separately. Thereafter, everything will be combined to show that the main research question of this master thesis was resolved.

7-1-1 Choice of state vector

The first subquestion that was investigated is:

How should the state vector of a time-invariant Max-Min-Plus-Scaling (MMPS) system be defined?

A conventional Model predictive control (MPC) controller aims to steer a state and input to a constant equilibrium value. For the MMPS system this is not possible for the time variables. In this master thesis this was solved by adding an extra state for each time variable $x_t(k)$ that was the difference between the current and the previous time variable $x_t(k) - x_t(k-1)$. Now this difference can be steered to a constant equilibrium value. However, it is not always possible to describe the difference in $x_t(k)$ without using the state $x_t(k)$ itself. This means that $x_t(k)$ needs to stay in the state vector as well. So the altered state vector will look like

$$x(k) = \begin{bmatrix} x_t(k) - x_t(k-1) \\ x_t(k) \\ x_q(k) \end{bmatrix}$$

In the case study this solution was applied to a model of an urban railway line. Here it was showed that the method works well for the undisturbed and disturbed case, as well as the case where parameter uncertainty is present.

7-1-2 Linearization of the state description

The section subquestion that was examined is:

How can a time-variant MMPS system be linearized?

In the process of determining the terminal cost that will ensure stability, a linearization of the MMPS system around the equilibrium is needed. This can be a challenge for a system that consists of the operations minimization and maximization.

This problem is solved by first rewriting a general MMPS system into the canonical conjunctive MMPS format. This is always possible. Now, one can determine which part of the conjunctive MMPS system is responsible for determining the new state value when the system is in the equilibrium. This part of the conjunctive formulation can be easily linearized. In this way one has determined the linearization of the original MMPS system around the equilibrium.

7-1-3 Choice of objective function

The third subquestion was:

How should the objective function be defined to guarantee stability for a time-variant MMPS system?

In this master thesis the procedure for controlling a nonlinear discrete-time (DT) system with a quadratic objective function was used as a basis. From there the procedure was altered such that is was valid for a (nonlinear) discrete-event (DE) MMPS system as well.

In this procedure it is necessary that the weighting matrix on the state in the stage cost is positive definite. This means that the weight on the original time variables $x_t(k)$ cannot be put at zero. This meant that an alteration to the original quadratic objective function was needed, because a time variable cannot be steered to a constant equilibrium value. To fix this, a variable equilibrium value was added in the stage cost $\ell(x, u)$ and terminal cost V_f that increases for the $x_t(k)$ state(s) as k increases.

Next to this, it could be concluded that the assumptions, that needed to hold for the nonlinear DT system such that the objective function could control it in a stable way, were also valid for the DE MMPS system.

7-1-4 Shape of terminal set

The third subquestion was the following:

What will the terminal set for a time-variant MMPS system look like?

The terminal constraint set of a (partially) additive homogeneous MMPS system is the sublevel set of the terminal cost. This is an ellipsoid with $x_{eq}(k)$ at its center. Since the value of

$x_{t,eq}(k)$ depends on the initial value and the value of k, the location of the terminal set is not always the same. Next to this $x_t(k)$ and $x_t(k) - x_t(k-1)$ are always larger than zero (since they represent times/time differences). So in general, the terminal set is a (multidimensional) ellipsoid, where for $x_t(k)$ and $x_t(k) - x_t(k-1)$ only the positive values are included.

7-1-5 Main question

The main research question that needed to be answered in this master thesis is:

How can the process of designing a stabilizing MPC controller for a time-invariant MMPS system be fully described?

For a DE system time-invariance comes down the system being (partially) additive homogeneous. So the question could be rephrased as:

How can the process of designing a stabilizing MPC controller for a (partially) additive homogeneous MMPS system be fully described?

This master thesis has successfully described the full process of how to design an MPC controller for such a system. To apply the procedure that was proposed in Chapter 3, several obstacles needed to be overcome. The first challenge was how to handle the equilibrium for time variables. Next to this, we needed to find out how to linearize the MMPS system around this acquired equilibrium. Furthermore, it needed to be checked whether the stability assumptions for the objective function for nonlinear DT systems were still valid for nonlinear DE systems. This included figuring out the shape of the terminal set.

7-2 Contributions

This master thesis has provided new insights in the field of system and control. These insights can be summarized as:

- It provides a clear overview of how to recast MPC control on an MMPS system as an Mixed integer quadratic programming (MIQP) problem.
- It provides a procedure on how to alter the state vector and how to choose a cost function such that the MMPS system is stabilized.
- It shows how it can be proven that the procedure works by investigating the stability.

Chapter 8

Recommendations for Future Work

As the Max-Min-Plus-Scaling (MMPS) system is a relatively new system description and therefore not that thoroughly researched yet, there are still a lot of areas that can be investigated further. To continue on the work that I have conducted in this thesis, I would recommend focusing on the next four areas:

- **Develop a more compact way to write the conjunctive MMPS equation.** It might be possible to rewrite $x_t(k) - x_t(k-1)$ in a way that it is not necessary to have the original $x_t(k)$ in the state vector as well. When you take a look at the system of the urban railway model, this would mean that you would have to fill in for example $d_j(k)$ in $a_j(k)$ multiple times. This would make the expression for $a_j(k)$ very large. However, multiple of these terms will most likely be redundant.

  Next to this, going from a maximization to a minimization (and vise versa) also adds a lot of terms. So if this has to be done multiple times in a row, this will most likely lead to redundant terms as well.

  So it would be interesting to investigate if it is possible to develop an algorithm where these redundant terms can be filtered out more easily.

- **Use a new weight on the original state such that $x_t(k)$ does not need to be weighted in the cost function.** Right now both $x_t(k)$ and $x_t(k) - x_t(k-1)$ are present in the state vector, because it is not possible to describe the states without $x_t(k)$. In the cost function of the Model predictive control (MPC) problem, the weight matrix on the state needs to be positive definite. So there has to be a weight on this original $x_t(k)$ value. This is not ideal, because if there are some perturbation in the system, it might not be possible to return to the wanted $x_t(k)$ value. The aim should be to get the growth rate $x_t(k) - x_t(k-1)$ back to the equilibrium value.

  A suggestion to solve this issue this is using the original state vector

$$x(k) = \begin{bmatrix} x_t(k) \\ x_q(k) \end{bmatrix}$$

and using a non-diagonal weighting matrix on these states, such that in that way only the difference in $x_t(k)$ can be charged.

For the stage cost on the state, this would look like:

$$(\begin{bmatrix} x(k-1) \\ x(k) \end{bmatrix} - \begin{bmatrix} x_{eq}(k-1) \\ x_{eq}(k) \end{bmatrix})^T E^T Q E (\begin{bmatrix} x(k-1) \\ x(k) \end{bmatrix} - \begin{bmatrix} x_{eq}(k-1) \\ x_{eq}(k) \end{bmatrix}) \tag{8-1}$$

where the new weight on the state is $E^T Q E$ and

$$E = \left[\begin{array}{cc|cc} -I_{n_t} & 0 & I_{n_t} & 0 \\ 0 & 0 & 0 & I_{n_q} \end{array} \right]$$

One can observe that it holds that

$$E \begin{bmatrix} x(k-1) \\ x(k) \end{bmatrix} = \begin{bmatrix} x_t(k) - x_t(k-1) \\ x_q(k) \end{bmatrix}$$

So indeed, only $x_t(k) - x_t(k-1)$ and $x_q(k)$ will be weighted. In a same way the event counter dependence in $x_{eq}(k)$ will be cancelled out as well. This can be extended to all states from $x(k-1)$ up until $x(k+N_p-1)$.

As this new weight matrix uses both $x(k)$ and $x(k-1)$, the weight matrices for the whole prediction horizon will partially overlap. It is interesting to investigate whether this new weight matrix will indeed work or whether new problems will arise.

- **Repeat the design process for other cost functions.** This research focused on the quadratic cost function, because it is the most commonly used in conventional MPC. Next to that, it also made the most sense for the Urban railway line. There are other cost function that might be interesting to investigate further, for example the 1-norm or ∞-norm. Furthermore, the Hilbert's projective norm could be interesting for the time variables, since this would circumvent the problem that the variable should go to an equilibrium value.

- **Design a new MMPS model.** There are not that many real life examples of MMPS systems yet. This is why it could be interesting to construct a new one. Then the procedure in this master thesis can be tested on this new MMPS system as well.

Appendix A

Urban Railway Network

This appendix expands on the details of the Urban railway network.

A-1 Conjunctive canonical formulation

This section elaborates on the conjunctive canonical formulation of the Urban railway network examined in this master thesis.

$$a_j(k) - a_j(k-1) = \min\Big(\max(\begin{bmatrix}0 & 0 & -1 & 0 & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}0 & 1 & 0\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix},$$

$$\begin{bmatrix}0 & 0 & -1 & 1 & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}0 & 0 & 1\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix})\Big)$$

$$d_j(k) - d_j(k-1) = \min\Big(\max(\begin{bmatrix}0 & 0 & 0 & -\alpha_1 & 0 & \alpha_3\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & \alpha_1 & \alpha_2 & 0\end{bmatrix} x_{j-1}(k) + u_j(k) + \begin{bmatrix}0 & \alpha_1 & 0\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix},$$

$$\begin{bmatrix}0 & 0 & 0 & 0 & 0 & \alpha_3\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & \alpha_2 & 0\end{bmatrix} x_{j-1}(k) + u_j(k) + \begin{bmatrix}0 & 0 & \alpha_1\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix}),$$

$$\max(\begin{bmatrix}0 & 0 & 0 & -1 & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 1 & \gamma_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}1/b & 1 & 0\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix},$$

$$\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & \gamma_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}1/b & 0 & 1\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix})\Big)$$

$$a_j(k) = \min\Big(\max(\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}0 & 1 & 0\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix},$$

$$\begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}0 & 0 & 1\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix})\Big)$$

$$
\begin{aligned}
d_j(k) = \min\Big(& \max(\begin{bmatrix} 0 & 0 & 0 & \alpha_4 & 0 & \alpha_3 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & \alpha_1 & \alpha_2 & 0 \end{bmatrix} x_{j-1}(k) + u_j(k) + \begin{bmatrix} 0 & \alpha_1 & 0 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} , \\
& \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & \alpha_3 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & 0 & \alpha_2 & 0 \end{bmatrix} x_{j-1}(k) + u_j(k) + \begin{bmatrix} 0 & 0 & \alpha_1 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix}) , \\
& \max(\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & 1 & \gamma_2 & 0 \end{bmatrix} x_{j-1}(k) + \begin{bmatrix} 1/b & 1 & 0 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} , \\
& \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & 0 & \gamma_2 & 0 \end{bmatrix} x_{j-1}(k) + \begin{bmatrix} 1/b & 0 & 1 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix}) \Big)
\end{aligned}
$$

$$
\begin{aligned}
\rho_j(k) = \min\Big(& \max\big(\min(\begin{bmatrix} 0 & 0 & 0 & b\alpha_4 & 0 & \alpha_1 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & -b\alpha_4 & b(\alpha_2-\gamma_2) & 0 \end{bmatrix} x_{j-1}(k) + e_j u_j(k) + \begin{bmatrix} 0 & -b\alpha_4 & 0 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} , \\
& \begin{bmatrix} 0 & 0 & 0 & -b\alpha_1 & 0 & \alpha_1 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & b\alpha_1 & b(\alpha_2-\gamma_2) & 0 \end{bmatrix} x_{j-1}(k) + e_j u_j(k) + \begin{bmatrix} 0 & b\alpha_1 & -b \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix}) , \\
& \min(\begin{bmatrix} 0 & 0 & 0 & b & 0 & \alpha_1 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & -b & b(\alpha_2-\gamma_2) & 0 \end{bmatrix} x_{j-1}(k) + e_j u_j(k) + \begin{bmatrix} 0 & -b & b\alpha_1 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} , \\
& \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \alpha_1 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & 0 & b(\alpha_2-\gamma_2) & 0 \end{bmatrix} x_{j-1}(k) + e_j u_j(k) + \begin{bmatrix} 0 & 0 & -b\alpha_4 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix})) , \\
& \max\big(\min(\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_{j-1}(k) + e_j u_j(k) + \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} , \\
& \begin{bmatrix} 0 & 0 & 0 & -b & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & b & 0 & 0 \end{bmatrix} x_{j-1}(k) + e_j u_j(k) + \begin{bmatrix} 1 & b & -b \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix}) , \\
& \min(\begin{bmatrix} 0 & 0 & 0 & b & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & -b & 0 & 0 \end{bmatrix} x_{j-1}(k) + e_j u_j(k) + \begin{bmatrix} 1 & -b & b \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} , \\
& \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_{j-1}(k) + e_j u_j(k) + \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix}))\Big)
\end{aligned}
$$

$$
\begin{aligned}
\sigma_j(k) = \min\Big(& \max\big(\min(\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_{j-1}(k) + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} , \\
& \begin{bmatrix} 0 & 0 & 0 & -b\alpha_1 & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & b\alpha_1 & 0 & 0 \end{bmatrix} x_{j-1}(k) + \begin{bmatrix} 0 & b\alpha_1 & -b\alpha_1 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix}) , \\
& \min(\begin{bmatrix} 0 & 0 & 0 & b\alpha_4 & 0 & \alpha_1 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & -b\alpha_4 & b(\alpha_2-\gamma_2) & 0 \end{bmatrix} x_{j-1}(k) + \begin{bmatrix} -1 & -b\alpha_4 & 0 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} , \\
& \begin{bmatrix} 0 & 0 & 0 & -b\alpha_1 & 0 & \alpha_1 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & b\alpha_1 & b(\alpha_2-\gamma_2) & 0 \end{bmatrix} x_{j-1}(k) + \begin{bmatrix} -1 & b\alpha_1 & -b \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix}) , \\
& \min(\begin{bmatrix} 0 & 0 & 0 & b & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & 0 & -b & 0 \end{bmatrix} x_{j-1}(k) + \begin{bmatrix} 0 & -b & b \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix} , \\
& \begin{bmatrix} 0 & 0 & 0 & b\alpha_4 & 0 & 0 \end{bmatrix} x_j(k-1) + \begin{bmatrix} 0 & 0 & 0 & -b\alpha_4 & 0 & 0 \end{bmatrix} x_{j-1}(k) + \begin{bmatrix} 0 & -b\alpha_4 & b\alpha_4 \end{bmatrix} \begin{bmatrix} \rho_{max} \\ \tau_{r,j} \\ \tau_H \end{bmatrix}) ,
\end{aligned}
$$

$$
\begin{aligned}
&\min\Big(\begin{bmatrix}0 & 0 & 0 & b\alpha_4+b & 0 & \alpha_1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & -b\alpha_4-b & b(\alpha_2-\gamma_2) & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}-1 & -b\alpha_4-b & b\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & b\alpha_4 & 0 & \alpha_1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & -b\alpha_4 & b(\alpha_2-\gamma_2) & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}-1 & -b\alpha_4 & 0\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big), \\
&\min\Big(\begin{bmatrix}0 & 0 & 0 & -b\alpha_4 & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & b\alpha_4 & 0 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}0 & b\alpha_4 & -b\alpha_4\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & -b & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & b & 0 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}0 & b & -b\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big), \\
&\min\Big(\begin{bmatrix}0 & 0 & 0 & 0 & 0 & \alpha_1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & b(\alpha_2-\gamma_2) & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}-1 & 0 & -b\alpha_4\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & -b & 0 & \alpha_1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & b & b(\alpha_2-\gamma_2) & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}-1 & b & -b\alpha_4-b\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big), \\
&\min\Big(\begin{bmatrix}0 & 0 & 0 & b\alpha_1 & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & -b\alpha_1 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}0 & -b\alpha_1 & b\alpha_1\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}0 & 0 & 0\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big), \\
&\min\Big(\begin{bmatrix}0 & 0 & 0 & b & 0 & \alpha_1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & -b & b(\alpha_2-\gamma_2) & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}-1 & -b & b\alpha_1\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & 0 & 0 & \alpha_1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & b(\alpha_2-\gamma_2) & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}-1 & 0 & -b\alpha_4\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big)\Big), \\
&\max\Big(\min\Big(\begin{bmatrix}0 & 0 & 0 & -e_j\alpha_4 & 0 & \alpha_4\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & e_j\alpha_4 & e_j\gamma_2+b\beta_j/f-b\alpha_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b & e_j\alpha_4 & 0\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & -b-e_j & 0 & \alpha_4\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & b+e_j & e_j\gamma_2+b\beta_j/f-b\alpha_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b & b+e_j & -b\alpha_1\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big), \\
&\min\Big(\begin{bmatrix}0 & 0 & 0 & -e_j & 0 & 1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & e_j & e_j\gamma_2+1-\beta_j & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b-1 & e_j & 0\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & -b-e_j & 0 & 1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & b+e_j & e_j\gamma_2+1-\beta_j & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b-1 & b+e_j & -b\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big), \\
&\min\Big(\begin{bmatrix}0 & 0 & 0 & b\alpha_1-e_j & 0 & \alpha_4\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & -b\alpha_1+e_j & e_j\gamma_2+b\beta_j/f-b\alpha_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b & -b\alpha_1+e_j & b\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & -e_j & 0 & \alpha_4\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & e_j & e_j\gamma_2+b\beta_j/f-b\alpha_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b & e_j & -b\alpha_4\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big), \\
&\min\Big(\begin{bmatrix}0 & 0 & 0 & b-e_j & 0 & 1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & -b+e_j & e_j\gamma_2+1-\beta_j & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b-1 & -b+e_j & b\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & -e_j & 0 & 1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & e_j & e_j\gamma_2+1-\beta_j & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b-1 & e_j & 0\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big), \\
&\min\Big(\begin{bmatrix}0 & 0 & 0 & -b\alpha_4 & 0 & \alpha_4\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & b\alpha_4 & e_j\gamma_2+b\beta_j/f-b\alpha_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b & b\alpha_4 & e_j\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & -b & 0 & \alpha_4\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & b & e_j\gamma_2+b\beta_j/f-b\alpha_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b & b & e_j-b\alpha_1\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big), \\
&\min\Big(\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & e_j\gamma_2+1-\beta_j & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b-1 & 0 & e_j\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}, \\
&\quad \begin{bmatrix}0 & 0 & 0 & -b & 0 & 1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & b & e_j\gamma_2+1-\beta_j & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b-1 & b & e_j-b\end{bmatrix} \begin{bmatrix}\rho_{max}\\ \tau_{r,j}\\ \tau_H\end{bmatrix}\Big),
\end{aligned}
$$

$$\min(\begin{bmatrix}0 & 0 & 0 & b\alpha_1 & 0 & \alpha_4\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & -b\alpha_1 & e_j\gamma_2 + b\beta_j/f - b\alpha_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b & -b\alpha_1 & b+e_j\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix},$$

$$\begin{bmatrix}0 & 0 & 0 & 0 & 0 & \alpha_4\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & e_j\gamma_2 + b\beta_j/f - b\alpha_2 & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b & 0 & -e_j\alpha_4\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix}),$$

$$\min(\begin{bmatrix}0 & 0 & 0 & b & 0 & 1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & -b & e_j\gamma_2 + 1 - \beta_j & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b - 1 & -b & b+e_j\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix},$$

$$\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} x_j(k-1) + \begin{bmatrix}0 & 0 & 0 & 0 & e_j\gamma_2 + 1 - \beta_j & 0\end{bmatrix} x_{j-1}(k) + \begin{bmatrix}e_j/b - 1 & 0 & e_j\end{bmatrix} \begin{bmatrix}\rho_{max} \\ \tau_{r,j} \\ \tau_H\end{bmatrix})))$$

A-2 Constaints MLD system

In this section the constraints are written out that are formed when the conjunctive Max-Min-Plus-Scaling (MMPS) system is written as a Mixed logical dynamical (MLD) system. This means that the compact notation

$$E_{1,j}x_j(k-1) + E_{2,j}u_j(k) + E_{3,j}\delta_j(k) + E_4 z_j(k) \leq g_{5,j} - E_{11,j}x_{j-1}(k)$$

is equal to:

$$\underbrace{\begin{bmatrix}
0&0&0&-1&0&0\\
0&0&0&1&0&0\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&-1&0&0\\
0&0&0&1&0&0\\
0&0&0&-\alpha_1&0&0\\
0&0&0&\alpha_1&0&0\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&-\alpha_1&0&0\\
0&0&0&\alpha_1&0&0\\
0&0&0&\alpha_4&0&\alpha_3\\
0&0&0&-\alpha_4&0&-\alpha_3\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&\alpha_4&0&\alpha_3\\
0&0&0&-\alpha_4&0&-\alpha_3\\
0&0&0&b&0&0\\
0&0&0&-b&0&0\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&b&0&0\\
0&0&0&-b&0&0\\
0&0&0&-b\alpha_1&0&0\\
0&0&0&b\alpha_1&0&0\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&-b\alpha_1&0&0\\
0&0&0&b\alpha_1&0&0\\
0&0&0&-b&0&0\\
0&0&0&b&0&0\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&-b&0&0\\
0&0&0&b&0&0\\
0&0&0&b\alpha_4&0&\alpha_1\\
0&0&0&-b\alpha_4&0&-\alpha_1\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&b\alpha_4&0&\alpha_1\\
0&0&0&-b\alpha_4&0&-\alpha_1\\
0&0&0&b\alpha_1&0&0\\
0&0&0&-b\alpha_1&0&0\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&b\alpha_1&0&0\\
0&0&0&-b\alpha_1&0&0\\
0&0&0&-b\alpha_4&0&-\alpha_1\\
0&0&0&b\alpha_4&0&\alpha_1\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&-b\alpha_4&0&-\alpha_1\\
0&0&0&b\alpha_4&0&\alpha_1\\
0&0&0&b\alpha_4&0&0\\
0&0&0&-b\alpha_4&0&0\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&b\alpha_4&0&0\\
0&0&0&-b\alpha_4&0&0\\
0&0&0&-e_j&0&0\\
0&0&0&e_j&0&0\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&-e_j&0&0\\
0&0&0&e_j&0&0\\
0&0&0&e_j\alpha_4&0&-\alpha_4\\
0&0&0&-e_j\alpha_4&0&\alpha_4\\
0&0&0&0&0&0\\
0&0&0&0&0&0\\
0&0&0&e_j\alpha_4&0&-\alpha_4\\
0&0&0&-e_j\alpha_4&0&\alpha_4\\
-1&0&0&0&0&0\\
0&-1&0&0&0&0\\
0&0&-1&0&0&0\\
0&0&0&-1&0&0\\
0&0&0&0&-1&0\\
0&0&0&0&0&-1\\
0&0&0&0&1&0\\
0&0&0&0&0&0\\
0&0&0&0&0&0
\end{bmatrix}}_{E_1(j)} x_j(k-1) + \underbrace{\begin{bmatrix}
0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\1\\-1\\0\\0\\1\\-1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\-1\\1
\end{bmatrix}} u_j(k) + \underbrace{\begin{bmatrix}
\varepsilon-m_1&0&0&0&0&0&0&0&0&0&0&0\\
-M_1&0&0&0&0&0&0&0&0&0&0&0\\
-M_1&0&0&0&0&0&0&0&0&0&0&0\\
m_1&0&0&0&0&0&0&0&0&0&0&0\\
-m_1&0&0&0&0&0&0&0&0&0&0&0\\
M_1&0&0&0&0&0&0&0&0&0&0&0\\
0&\varepsilon-m_2&0&0&0&0&0&0&0&0&0&0\\
0&-M_2&0&0&0&0&0&0&0&0&0&0\\
0&-M_2&0&0&0&0&0&0&0&0&0&0\\
0&m_2&0&0&0&0&0&0&0&0&0&0\\
0&-m_2&0&0&0&0&0&0&0&0&0&0\\
0&M_2&0&0&0&0&0&0&0&0&0&0\\
0&0&m_3-\varepsilon&0&0&0&0&0&0&0&0&0\\
0&0&M_3&0&0&0&0&0&0&0&0&0\\
0&0&-M_3&0&0&0&0&0&0&0&0&0\\
0&0&m_3&0&0&0&0&0&0&0&0&0\\
0&0&-m_3&0&0&0&0&0&0&0&0&0\\
0&0&M_3&0&0&0&0&0&0&0&0&0\\
0&0&0&m_4-\varepsilon&0&0&0&0&0&0&0&0\\
0&0&0&M_4&0&0&0&0&0&0&0&0\\
0&0&0&-M_4&0&0&0&0&0&0&0&0\\
0&0&0&m_4&0&0&0&0&0&0&0&0\\
0&0&0&-m_4&0&0&0&0&0&0&0&0\\
0&0&0&M_4&0&0&0&0&0&0&0&0\\
0&0&0&0&\varepsilon-m_5&0&0&0&0&0&0&0\\
0&0&0&0&-M_5&0&0&0&0&0&0&0\\
0&0&0&0&-M_5&0&0&0&0&0&0&0\\
0&0&0&0&m_5&0&0&0&0&0&0&0\\
0&0&0&0&-m_5&0&0&0&0&0&0&0\\
0&0&0&0&M_5&0&0&0&0&0&0&0\\
0&0&0&0&0&\varepsilon-m_6&0&0&0&0&0&0\\
0&0&0&0&0&-M_6&0&0&0&0&0&0\\
0&0&0&0&0&-M_6&0&0&0&0&0&0\\
0&0&0&0&0&m_6&0&0&0&0&0&0\\
0&0&0&0&0&-m_6&0&0&0&0&0&0\\
0&0&0&0&0&M_6&0&0&0&0&0&0\\
0&0&0&0&0&0&m_7-\varepsilon&0&0&0&0&0\\
0&0&0&0&0&0&M_7&0&0&0&0&0\\
0&0&0&0&0&0&-M_7&0&0&0&0&0\\
0&0&0&0&0&0&m_7&0&0&0&0&0\\
0&0&0&0&0&0&-m_7&0&0&0&0&0\\
0&0&0&0&0&0&M_7&0&0&0&0&0\\
0&0&0&0&0&0&0&m_8-\varepsilon&0&0&0&0\\
0&0&0&0&0&0&0&M_8&0&0&0&0\\
0&0&0&0&0&0&-M_8&0&0&0&0&0\\
0&0&0&0&0&0&0&m_8&0&0&0&0\\
0&0&0&0&0&0&0&-m_8&0&0&0&0\\
0&0&0&0&0&0&0&M_8&0&0&0&0\\
0&0&0&0&0&0&0&0&\varepsilon-m_9&0&0&0\\
0&0&0&0&0&0&0&0&-M_9&0&0&0\\
0&0&0&0&0&0&0&0&-M_9&0&0&0\\
0&0&0&0&0&0&0&0&m_9&0&0&0\\
0&0&0&0&0&0&0&0&-m_9&0&0&0\\
0&0&0&0&0&0&0&0&M_9&0&0&0\\
0&0&0&0&0&0&0&0&0&\varepsilon-m_{10}&0&0\\
0&0&0&0&0&0&0&0&0&-M_{10}&0&0\\
0&0&0&0&0&0&0&0&0&-M_{10}&0&0\\
0&0&0&0&0&0&0&0&0&m_{10}&0&0\\
0&0&0&0&0&0&0&0&0&-m_{10}&0&0\\
0&0&0&0&0&0&0&0&0&M_{10}&0&0\\
0&0&0&0&0&0&0&0&0&0&\varepsilon-m_{11}&0\\
0&0&0&0&0&0&0&0&0&0&-M_{11}&0\\
0&0&0&0&0&0&0&0&0&0&-M_{11}&0\\
0&0&0&0&0&0&0&0&0&0&m_{11}&0\\
0&0&0&0&0&0&0&0&0&0&-m_{11}&0\\
0&0&0&0&0&0&0&0&0&0&M_{11}&0\\
0&0&0&0&0&0&0&0&0&0&0&m_{12}-\varepsilon\\
0&0&0&0&0&0&0&0&0&0&0&M_{12}\\
0&0&0&0&0&0&0&0&0&0&0&-M_{12}\\
0&0&0&0&0&0&0&0&0&0&0&m_{12}\\
0&0&0&0&0&0&0&0&0&0&0&-m_{12}\\
0&0&0&0&0&0&0&0&0&0&0&M_{12}\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}}_{E_3} \delta_j(k)$$

$$
+\underbrace{\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & & 0 & 0 & -1 & -1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}}_{E_4} z_j(k) \leq
\underbrace{\begin{bmatrix}
0 & -1 & 1 & -m_1\\
0 & 1 & -1 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & -1 & 1 & -m_1\\
0 & 1 & -1 & M_1\\
0 & -\alpha_1 & \alpha_1 & -m_2\\
0 & \alpha_1 & -\alpha_1 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & -\alpha_1 & \alpha_1 & -m_2\\
0 & \alpha_1 & -\alpha_1 & M_2\\
\frac{1}{b} & \alpha_4 & 0 & -\varepsilon\\
-\frac{1}{b} & -\alpha_4 & 0 & M_3\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
\frac{1}{b} & \alpha_4 & 0 & -m_3\\
-\frac{1}{b} & -\alpha_4 & 0 & M_3\\
0 & b & -b & -\varepsilon\\
0 & -b & b & M_4\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & b & -b & -m_4\\
0 & -b & b & M_4\\
0 & -b\alpha_1 & b\alpha_1 & -m_5\\
0 & b\alpha_1 & -b\alpha_1 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & -b\alpha_1 & b\alpha_1 & -m_5\\
0 & b\alpha_1 & -b\alpha_1 & M_5\\
0 & -b & b & -m_6\\
0 & b & -b & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & -b & b & -m_6\\
0 & b & -b & M_6\\
1 & b\alpha_4 & 0 & -\varepsilon\\
-1 & -b\alpha_4 & 0 & M_7\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
1 & b\alpha_4 & 0 & -m_7\\
-1 & -b\alpha_4 & 0 & M_7\\
0 & b\alpha_1 & -b\alpha_1 & -m_8\\
0 & -b\alpha_1 & b\alpha_1 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & b\alpha_1 & -b\alpha_1 & -m_8\\
0 & -b\alpha_1 & b\alpha_1 & M_8\\
-1 & -b\alpha_4 & 0 & -m_9\\
1 & b\alpha_4 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
-1 & -b\alpha_4 & 0 & -m_9\\
1 & b\alpha_4 & 0 & M_9\\
0 & b\alpha_4 & -b\alpha_4 & -m_9\\
0 & -b\alpha_4 & b\alpha_4 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & b\alpha_4 & -b\alpha_4 & -m_{10}\\
0 & -b\alpha_4 & b\alpha_4 & M_{10}\\
0 & -e_j & e_j & -m_{11}\\
0 & e_j & -e_j & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & -e_j & e_j & -m_{11}\\
0 & e_j & -e_j & M_{11}\\
\frac{e_j}{b} & e_j\alpha_4 & 0 & -\varepsilon\\
-\frac{e_j}{b} & -e_j\alpha_4 & 0 & M_{12}\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
\frac{e_j}{b} & e_j\alpha_4 & 0 & -m_{12}\\
-\frac{e_j}{b} & -e_j\alpha_4 & 0 & M_{12}\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
1 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & u_{max}
\end{bmatrix}
\begin{bmatrix}
\rho_{max}\\
\tau_{r,j}\\
\tau_H\\
1
\end{bmatrix}
-
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0\\
0 & 0 & 0 & \alpha_1 & 0 & 0\\
0 & 0 & 0 & -\alpha_1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & \alpha_1 & 0 & 0\\
0 & 0 & 0 & -\alpha_1 & 0 & 0\\
0 & 0 & 0 & -\alpha_4 & \alpha_2-\gamma_2 & 0\\
0 & 0 & 0 & \alpha_4 & \gamma_2-\alpha_2 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -\alpha_4 & \alpha_2-\gamma_2 & 0\\
0 & 0 & 0 & \alpha_4 & \gamma_2-\alpha_2 & 0\\
0 & 0 & 0 & -b & 0 & 0\\
0 & 0 & 0 & b & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -b & 0 & 0\\
0 & 0 & 0 & b & 0 & 0\\
0 & 0 & 0 & b\alpha_1 & 0 & 0\\
0 & 0 & 0 & -b\alpha_1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & b\alpha_1 & 0 & 0\\
0 & 0 & 0 & -b\alpha_1 & 0 & 0\\
0 & 0 & 0 & b & 0 & 0\\
0 & 0 & 0 & -b & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & b & 0 & 0\\
0 & 0 & 0 & -b & 0 & 0\\
0 & 0 & 0 & -b\alpha_4 & b(\alpha_2-\gamma_2) & 0\\
0 & 0 & 0 & b\alpha_4 & b(\gamma_2-\alpha_2) & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -b\alpha_4 & b(\alpha_2-\gamma_2) & 0\\
0 & 0 & 0 & b\alpha_4 & b(\gamma_2-\alpha_2) & 0\\
0 & 0 & 0 & -b\alpha_1 & 0 & 0\\
0 & 0 & 0 & b\alpha_1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -b\alpha_1 & 0 & 0\\
0 & 0 & 0 & b\alpha_1 & 0 & 0\\
0 & 0 & 0 & b\alpha_4 & b(\gamma_2-\alpha_2) & 0\\
0 & 0 & 0 & -b\alpha_4 & b(\alpha_2-\gamma_2) & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & b\alpha_4 & b(\gamma_2-\alpha_2) & 0\\
0 & 0 & 0 & -b\alpha_4 & b(\alpha_2-\gamma_2) & 0\\
0 & 0 & 0 & -b\alpha_4 & 0 & 0\\
0 & 0 & 0 & b\alpha_4 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -b\alpha_4 & 0 & 0\\
0 & 0 & 0 & b\alpha_4 & 0 & 0\\
0 & 0 & 0 & e_j & 0 & 0\\
0 & 0 & 0 & -e_j & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & e_j & 0 & 0\\
0 & 0 & 0 & -e_j & 0 & 0\\
0 & 0 & 0 & -e_j\alpha_4 & -e_j\gamma_2-\frac{b\beta_j(j)}{f}+b\alpha_2 & 0\\
0 & 0 & 0 & e_j\alpha_4 & e_j\gamma_2+\frac{b\beta_j(j)}{f}-b\alpha_2 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -e_j\alpha_4 & -e_j\gamma_2-\frac{b\beta_j(j)}{f}+b\alpha_2 & 0\\
0 & 0 & 0 & e_j\alpha_4 & e_j\gamma_2+\frac{b\beta_j(j)}{f}-b\alpha_2 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}}_{g_5(j)} x_{j-1}(k)
$$

A-3 Parameter values

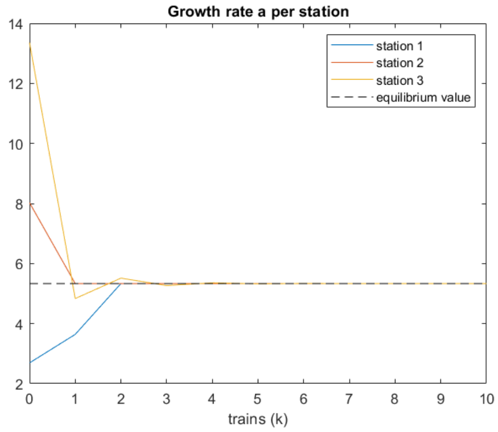
This appendix lists the parameter values that are used in the case study. These parameters satisfy the conditions in Equation 6-17.

Parameter value	Description
$N_p = 5$	prediction horizon
$J = 3$	total # stations
$K = 100$	total # trains
$\rho_{max} = 217$	[#] max. people in a train
$\tau_{r,j} = 2.692$	[min] running times
$e_j = 8$	[#/min] passenger entering station per minute
$b = 30$	[#/min] passengers that can board the train per minute
$f = 35$	[#/min] passengers that can disembark the train per minute
$\beta_j = 0.667$	[-] fraction of passengers in train k leaving train at station j
$\tau_H = 1$	[min] headway time
$\bar{\rho}_0 = 64$	[#] $\rho_j(0) = \bar{\rho}_0$
$\bar{\rho}_1 = 64$	[#] $\rho_1(k) = \bar{\rho}_1$
$\bar{\tau}_1 = 5.333$	[min] $d_1(k) = (k-1)\bar{\tau}_1$
$\bar{\tau}_0 = 5.333$	[min] $d_j(0) = j\bar{\tau}_0$
$\rho_{eq} = 64$	[#] amount of people in train at equilibrium

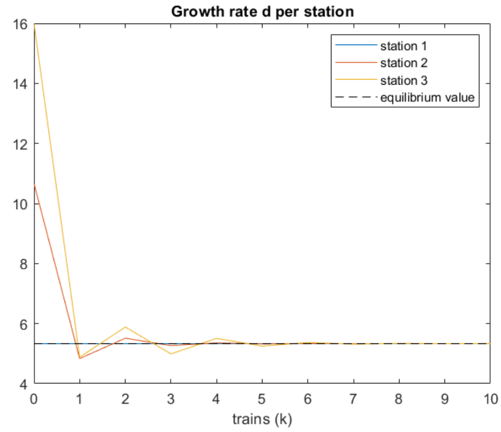
Table A-1: Parameter values Urban railway model

A-4 Results uncertainty on lowered β_2 value

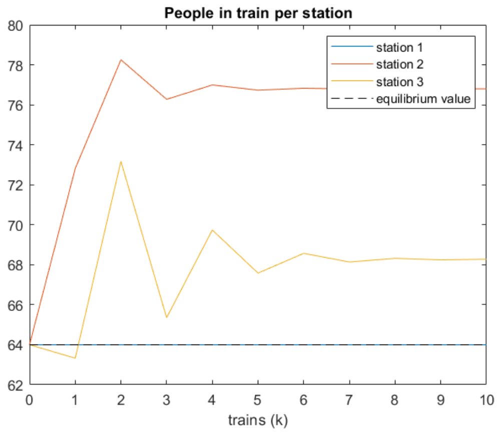
This appendix gives the results for a parametric uncertainty in β_2 which is equal to $\beta_j - 0.2$.



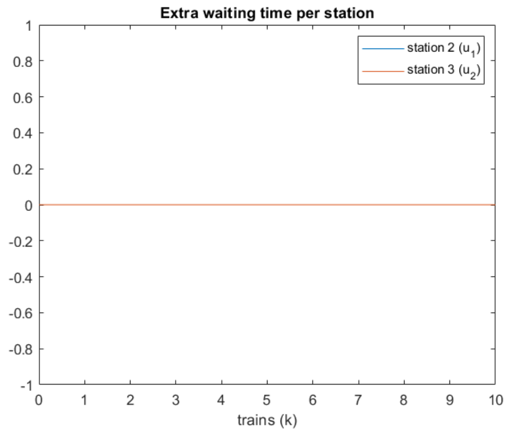
(a) Results for $a_j(k) - a_j(k-1)$



(b) Results for $d_j(k) - d_j(k-1)$



(c) Results for $\rho_j(k)$

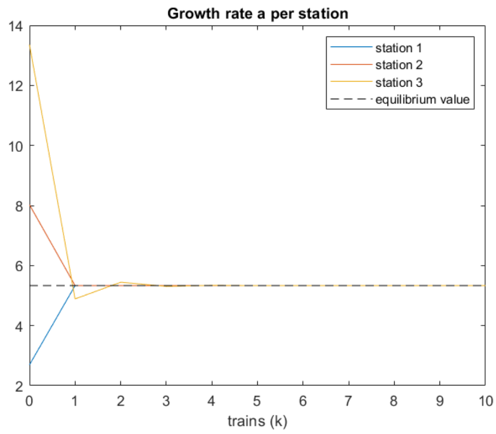


(d) Results for $u_j(k)$

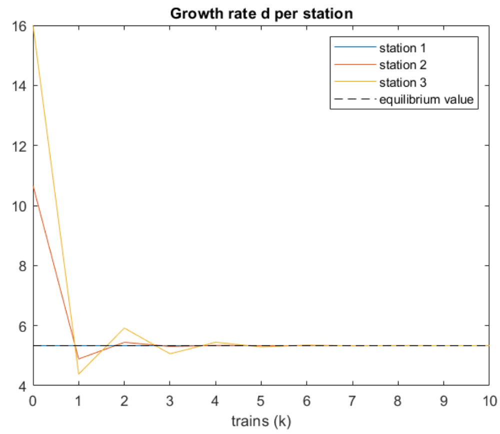
Figure A-1: Case 3: Results MPC with parameter uncertainty in β_2

A-5 Results uncertainty on lowered e_2 value

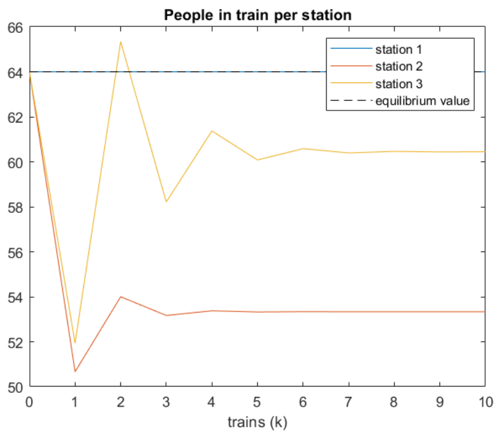
This appendix gives the results for a parametric uncertainty in e_2 which is equal to $e_2 - 2$.



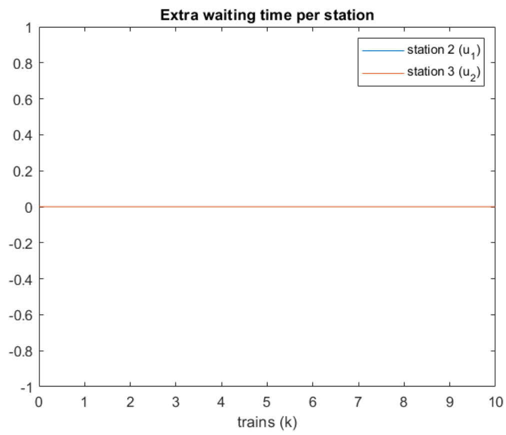
(a) Results for $a_j(k) - a_j(k-1)$



(b) Results for $d_j(k) - d_j(k-1)$



(c) Results for $\rho_j(k)$

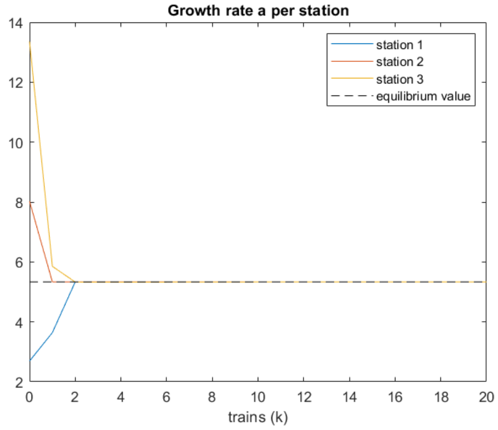


(d) Results for $u_j(k)$

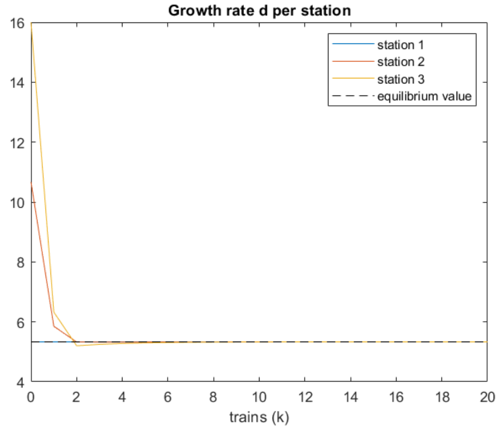
Figure A-2: Case 1b: Results MPC with parameter uncertainty for lowered e_2

A-6 Results uncertainty on heightened ρ_1 value

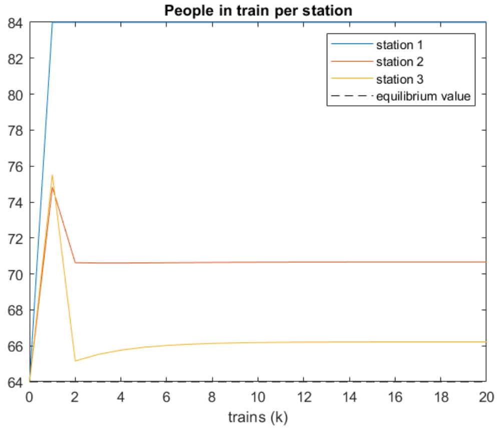
This appendix gives the results for a parametric uncertainty in $\bar{\rho}_1$ which is equal to $\bar{\rho}_1 + 20$.



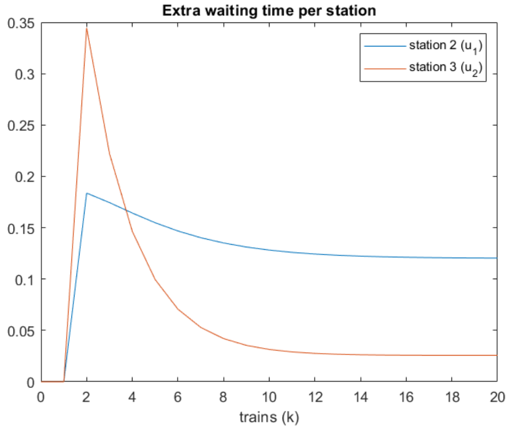
(a) Results for $a_j(k) - a_j(k-1)$



(b) Results for $d_j(k) - d_j(k-1)$



(c) Results for $\rho_j(k)$



(d) Results for $u_j(k)$

Figure A-3: Results MPC with parameter uncertainty for heightened $\bar{\rho}_1$

Appendix B

Matlab code

B-1 Main code

```
clear all
close all
clc

% the states: [a_j(k)-a_j(k-1) d_j(k)-d_j(k-1) a_j(k) d_j(k) rho_j(k)
    sigma_j(k)]
%% Variables
load equilibriumvar_full6st.mat
load Variables_6basic.mat

A111=zeros(6,6,J);
A1111=zeros(6,6,J);
B111=zeros(6,1,J);
B411=zeros(6,1,J);
for j=2:J
    A111(:,3:end,j)=[-1 0 0 0;0 -alpha_1(j) 0 alpha_3(j);0 0 0 0;0
        alpha_4(j) 0 alpha_3(j);0 b*alpha_4(j) 0 alpha_1(j);0 0 0 0];
    A1111(:,3:end,j)=[0 1 0 0;0 alpha_1(j) alpha_2(j) 0;0 1 0 0;0 alpha_1
        (j) alpha_2(j) 0;0 -b*alpha_4(j) b*(alpha_2(j)-gamma_2(j)) 0;0 0 0
        0];
    B111(:,:,j)=[0;1;0;1;e_j(j);0];
    B411(:,:,j)=[0 1 0;0 alpha_1(j) 0;0 1 0;0 alpha_1(j) 0;0 -b*alpha_4(j
        ) 0;0 0 0]*[rho_max;tau_rj(j);tau_H];
end

for j=2:3
    A1(:,:,j)=A111(:,:,j);
    A11(:,:,j)=A1111(:,:,j);
    B1(:,:,j)=B111(:,:,j);
    B4(:,:,j)=B411(:,:,j);
```

```
end

matr1=A1(:,:,2);
matr2=A11(:,:,2);
matr3=B1(:,:,2);
matr4=eye(size(A11(:,:,2),2));
A1tot=[matr1 zeros(size(A1(:,:,2),1),size(A1(:,:,2),2)*(J-2))];
A11tot=matr2;
B1tot=[matr3 zeros(size(B1(:,:,2),1),size(B1(:,:,2),2)*(J-2))];
B4hor=[matr4 zeros(size(A11(:,:,2),1),size(A11(:,:,2),2)*(J-2))];
B4vert=B4(:,:,2);
for j=3%:J
    matr1= [A11(:,:,j)*matr1 A1(:,:,j)];
    A1tot= [A1tot;matr1 zeros(size(A1(:,:,j),1),size(A1(:,:,j),2)*(J-j))];
    matr2= A11(:,:,j)*matr2;
    A11tot= [A11tot;matr2];
    matr3= [A11(:,:,j)*matr3 B1(:,:,j)];
    B1tot= [B1tot;matr3 zeros(size(B1(:,:,j),1),size(B1(:,:,j),2)*(J-j))];
    matr4= [A11(:,:,j)*matr4 eye(size(A11(:,:,j),2))];
    B4hor= [B4hor;matr4 zeros(size(A11(:,:,j),1),size(A11(:,:,j),2)*(J-j))];
    B4vert= [B4vert;B4(:,:,j)]
end
B4tot=B4hor*B4vert;
eq.A1tot=A1tot;
eq.A11tot=A11tot;
eq.B1tot=B1tot;
eq.B4tot=B4tot;

%% State feedback gain & weight matrix
Qx=kron(eye(J-1),diag([10 10 0.01 0.01 10 1]));
Qu=kron(eye(J-1),3);
mu=2;

K_L=dlqr(eq.A1tot,eq.B1tot,Qx,Qu);
Q_K=Qx+K_L'*Qu*K_L;
P=dlyap((eq.A1tot-eq.B1tot*K_L)',mu*Q_K);

%% maxima & minima
maximum=24*60*100;                          % minutes in a day
minimum=-maximum;
Max1=maximum;
Max2=maximum;
Max3=maximum;
Max4=maximum;
Max5=maximum;
Max6=maximum;
Max7=maximum;
Max8=maximum;
Max9=maximum;
Max10=maximum;
```

```
Max11=maximum;
Max12=maximum;

min1=minimum;
min2=minimum;
min3=minimum;
min4=minimum;
min5=minimum;
min6=minimum;
min7=minimum;
min8=minimum;
min9=minimum;
min10=minimum;
min11=minimum;
min12=minimum;

%% State initialization
sys_small=buildsysk_6states(Max1,min1,Max2,min2,Max3,min3,Max4,min4,Max5,
    min5,Max6,min6,Max7,min7,Max8,min8,Max9,min9,Max10,min10,Max11,min11,
    Max12,min12,J,1);
addpath 'C:\Users\justi\Documents\Werktuigbouw\SC\Afstuderen\Matlab code\
    Full model'
sys_full=buildsysk_fullmodel(sys_small,J);

%% Disturbance
% model error
sys_small_dis=buildsysk_6states(Max1,min1,Max2,min2,Max3,min3,Max4,min4,
    Max5,min5,Max6,min6,Max7,min7,Max8,min8,Max9,min9,Max10,min10,Max11,
    min11,Max12,min12,J,2);
sys_full_dis=buildsysk_fullmodel(sys_small_dis,J);


%% MPC using YALMIP and Gurobi
%%
choice_case=2;       % no disturbance: case 0, for output disturbance:
    choose case 1 or case 2, for uncertainty: case 3
% parameter uncertainty
if choice_case==3
    load Variables_6basic_uncertain.mat
end

% output disturbances
pulse_rho=20;
pulse_a=2;
dis=zeros(6*J,K+1);

% initialize free variables
Size_u=(J-1)*1;
Size_d=(J-1)*12;
Size_z=(J-1)*12;

V_rec=zeros(Size_u+Size_d+Size_z,K);
u_rec=zeros(Size_u,K);
```

```matlab
delta_rec=zeros(Size_d,K);
z_rec=zeros(Size_z,K);
J_rec=zeros(Size_u,K);

% initialize state vector
x=zeros(6*J,K+1);
x(1:6,:)=x1_k;
xjzero=xj_0(:,2:J);
x(7:end,1)=xjzero(:);


for k=2:K+1 % k=1 is k=0
    Np=min(5,K+1-k+1);
    Nc=Np;

    [MILPsysk,M2k,M3k,Mx1k]=MILP_sysk_fullmodel6st(sys_full,Np,Nc,x(7:end,k-1),x1_k(:,k:k+Np-1),rho_max,x_eq6(7:end,k-1:k+Np-1),u_eq(:,k:k+Np-1),delta_eq(:,k:k+Np-1),z_eq(:,k:k+Np-1),J,P,Qx,Qu);

    % Solve the constrained optimization problem (with YALMIP)
    V_con = [sdpvar((Size_u)*(Np),1);binvar((Size_d)*(Np),1);sdpvar((Size_z)*(Np),1)];                 % define optimization variable
    Constraint = [MILPsysk.A*V_con<=MILPsysk.b];              % define constraints
    Objective = V_con'*MILPsysk.Q*V_con+(MILPsysk.c)*V_con; % define cost function
    options = sdpsettings('solver','gurobi','verbose', 1);%,'gurobi.MIPGap',.1,'gurobi.MIPGapAbs',0.01);
    optimize(Constraint,Objective,options)                    %solve the problem

    % Select the first input only
    V_rec(:,k) =       [V_con(1:Size_u);V_con(Size_u*(Np)+1:Size_u*(Np)+Size_d);V_con(Size_u*(Np)+Size_d*(Np)+1:Size_u*(Np)+Size_d*(Np)+Size_z)];
    u_rec(:,k) =       V_con(1:Size_u);
    delta_rec(:,k) =   V_con(Size_u*(Np)+1:Size_u*(Np)+Size_d);
    z_rec(:,k) =       V_con(Size_u*(Np)+Size_d*(Np)+1:Size_u*(Np)+Size_d*(Np)+Size_z);

    % Compute the state/output evolution
    % case 1: no disturbance/noise
    x(7:end,k) =   sys_full.A1*x(7:end,k-1)+sys_full.B1*u_rec(:,k)+sys_full.B2*delta_rec(:,k)+sys_full.B3*z_rec(:,k)+sys_full.B4+sys_full.A11*x1_k(:,k);
    % case 2: add output disturbance
    if (k>=5) && (k<=10)
        if choice_case==1
            % ten extra people (extra waiting time)
            if x(1*6+5,k)+pulse_rho<=rho_max
                pulse_rho_real=pulse_rho;
            else
```

```
                    pulse_rho_real=rho_max−x(1*6+5,k);
                end
                dis(6*1+2,k)=pulse_rho_real/b;
                dis(6*1+4,k)=pulse_rho_real/b;
                dis(6*1+5,k)=pulse_rho_real;
                dis(6*1+6,k)=pulse_rho−pulse_rho_real;

            elseif choice_case==2
                % lower speed
                dis(6*1+1,k)=pulse_a;
                dis(6*1+2,k)=pulse_a*alpha_1(2);
                dis(6*1+3,k)=pulse_a;
                dis(6*1+4,k)=pulse_a*alpha_1(2);
                dis(6*1+5,k)=−pulse_a*b*alpha_4(2);
            end
        end
        x(7:end,k) =    x(7:end,k)+dis(7:end,k);

        % case 3: model error
%           x(7:end,k) =   sys_full_dis.A1*x(7:end,k-1)+sys_full_dis.B1*
    u_rec(:,k)+sys_full_dis.B2*delta_rec(:,k)+sys_full_dis.B3*z_rec(:,k)+
    sys_full_dis.B4+sys_full_dis.A11*x1_k(:,k);

        clear V_con

end

%% plot results
close all

% crop figures
K_plot=20;


k=0:K_plot;                              %  used for plotting equilibrium value

figure (1)
plot (0:K_plot,x(1,1:K_plot+1))
hold on
plot (0:K_plot,x(7,1:K_plot+1))
plot (0:K_plot,x(13,1:K_plot+1))
plot (0:K_plot,beta_j(1)/e_j(1)*rho_eq*ones(1,length(k)),'k--')
% legend('station 1','station 2','station 3')
legend('station 1','station 2','station 3','equilibrium value')
xlabel('trains (k)')
title('Growth rate a per station')

figure (2)
plot (0:K_plot,x(2,1:K_plot+1))
hold on
plot (0:K_plot,x(8,1:K_plot+1))
plot (0:K_plot,x(14,1:K_plot+1))
plot (0:K_plot,beta_j(1)/e_j(1)*rho_eq*ones(1,length(k)),'k--')
```

```
legend('station 1','station 2','station 3','equilibrium value')
xlabel('trains (k)')
title('Growth rate d per station')

figure (3)
plot (0:K_plot,x(3,1:K_plot+1))
hold on
plot (0:K_plot,x(9,1:K_plot+1))
plot (0:K_plot,x(15,1:K_plot+1))
legend('station 1','station 2','station 3')
xlabel('trains (k)')
title('Arrival time per station')

figure (4)
plot (0:K_plot,x(4,1:K_plot+1))
hold on
plot (0:K_plot,x(10,1:K_plot+1))
plot (0:K_plot,x(16,1:K_plot+1))
legend('station 1','station 2','station 3')
xlabel('trains (k)')
title('Departure time per station')

figure (5)
plot (0:K_plot,x(5,1:K_plot+1))
hold on
plot (0:K_plot,x(11,1:K_plot+1))
plot (0:K_plot,x(17,1:K_plot+1))
plot (0:K_plot,rho_eq*ones(1,length(k)),'k--')
legend('station 1','station 2','station 3','equilibrium value')
xlabel('trains (k)')
title('People in train per station')

figure (6)
plot (0:K_plot,x(6,1:K_plot+1))
hold on
plot (0:K_plot,x(12,1:K_plot+1))
plot (0:K_plot,x(18,1:K_plot+1))
legend('station 1','station 2','station 3')
xlabel('trains (k)')
title('People left at platform per station')

figure (7)
plot (0:K_plot,u_rec(1,1:K_plot+1))
hold on
plot (0:K_plot,u_rec(2,1:K_plot+1))
legend('station 2 (u_1)','station 3 (u_2)')
xlabel('trains (k)')
title('Extra waiting time per station')
```

B-2 Build MLD model per station

```
function sys=buildsysk_6states(Max1,min1,Max2,min2,Max3,min3,Max4,min4,Max5,min5,Max6,min6,Max7,min7,Max8,min8,Max9,min9,Max10,min10,Max11,min11,Max12,min12,J,var)
if var==1
    load Variables_6basic.mat
elseif var==2
    load Variables_6basic_uncertain.mat
end

sys.A1=     zeros(6,6,J);     % x_j(k-1)
sys.A11=    zeros(6,6,J);     % x_j-1(k)
sys.B1=     zeros(6,1,J);     % u_j(k)
sys.B2=     zeros(6,12,J);    % delta_j(k)
sys.B3=     zeros(6,12);      % z_j(k)
sys.B4_mat= zeros(6,3,J);
sys.B4=     zeros(6,1,J);

% forming the matrices
for j=1:J
    sys.A1(:,3:end,j)= [-1 0 0 0;0 -alpha_1(j) 0 alpha_3(j);0 0 0 0;0 alpha_4(j) 0 alpha_3(j);0 b*alpha_4(j) 0 alpha_1(j);0 0 0 0];
    sys.A11(:,3:end,j)=[0 1 0 0;0 alpha_1(j) alpha_2(j) 0;0 1 0 0;0 alpha_1(j) alpha_2(j) 0;0 -b*alpha_4(j) b*(alpha_2(j)-gamma_2(j)) 0;0 0 0 0];
    sys.B1(:,:,j)=     [0;1;0;1;e_j(j);0];
    sys.B4_mat(:,:,j)= [0 1 0;0 alpha_1(j) 0;0 1 0;0 alpha_1(j) 0;0 -b*alpha_4(j) 0;0 0 0];

    sys.B4(:,:,j)=     sys.B4_mat(:,:,j)*[rho_max;tau_rj(j);tau_H];
end

sys.B3(1,1)=           1;
sys.B3(2,2:3)=         [1 1];
sys.B3(3,1)=           1;
sys.B3(4,2:3)=         [1 1];
sys.B3(5,4:7)=         [1 1 0 1];
sys.B3(6,6:12)=        [1 0 1 1 1 0 1];

% matrix sizes constaints
sys.E1=        zeros(6*12,6,J);        % x_j(k-1)
sys.E11=       zeros(6*12,6,J);        % x_k-1(k)
sys.E2=        zeros(6*12,1,J);        % u_j(k)
```

```
sys.E3=         zeros(6*12,12);                                       % delta_j(k)
sys.E4=         zeros(6*12,12);                                       % z_j(k)
sys.g5_mat=     zeros(6*12,4,J);
sys.g5=         zeros(6*12,1,J);

% forming the matrices (incl. u=>0)
for j=1:J
    sys.E1(1:6,4,j)=        [-1;1;0;0;-1;1];
    sys.E1(7:12,4,j)=       [-alpha_1(j);alpha_1(j);0;0;-alpha_1(j);alpha_1(j)];
    sys.E1(13:18,4:end,j)=  [alpha_4(j) 0 alpha_3(j);-alpha_4(j) 0 -alpha_3(j);0 0 0;0 0 0;alpha_4(j) 0 alpha_3(j);-alpha_4(j) 0 -alpha_3(j)];
    sys.E1(19:24,4,j)=      [b;-b;0;0;b;-b];
    sys.E1(25:30,4,j)=      [-b*alpha_1(j);b*alpha_1(j);0;0;-b*alpha_1(j);b*alpha_1(j)];
    sys.E1(31:36,4,j)=      [-b;b;0;0;-b;b];
    sys.E1(37:42,4:end,j)=    [b*alpha_4(j) 0 alpha_1(j);-b*alpha_4(j) 0 -alpha_1(j);0 0 0;0 0 0;b*alpha_4(j) 0 alpha_1(j);-b*alpha_4(j) 0 -alpha_1(j)];
    sys.E1(43:48,4,j)=      [b*alpha_1(j);-b*alpha_1(j);0;0;b*alpha_1(j);-b*alpha_1(j)];
    sys.E1(49:54,4:end,j)=  [-b*alpha_4(j) 0 -alpha_1(j);b*alpha_4(j) 0 alpha_1(j);0 0 0;0 0 0;-b*alpha_4(j) 0 -alpha_1(j);b*alpha_4(j) 0 alpha_1(j)];
    sys.E1(55:60,4,j)=      [b*alpha_4(j);-b*alpha_4(j);0;0;b*alpha_4(j);-b*alpha_4(j)];
    sys.E1(61:66,4,j)=      [-e_j(j);e_j(j);0;0;-e_j(j);e_j(j)];
    sys.E1(67:72,4:end,j)=  [e_j(j)*alpha_4(j) 0 -alpha_4(j);-e_j(j)*alpha_4(j) 0 alpha_4(j);0 0 0;0 0 0;e_j(j)*alpha_4(j) 0 -alpha_4(j);-e_j(j)*alpha_4(j) 0 alpha_4(j)];

    sys.E11(1:6,4,j)=         [1;-1;0;0;1;-1];
    sys.E11(7:12,4,j)=        [alpha_1(j);-alpha_1(j);0;0;alpha_1(j);-alpha_1(j)];
    sys.E11(13:18,4:5,j)=     [-alpha_4(j) alpha_2(j)-gamma_2(j);alpha_4(j) gamma_2(j)-alpha_2(j);0 0;0 0;-alpha_4(j) alpha_2(j)-gamma_2(j);alpha_4(j) gamma_2(j)-alpha_2(j)];
    sys.E11(19:24,4,j)=       [-b;b;0;0;-b;b];
    sys.E11(25:30,4,j)=       [b*alpha_1(j);-b*alpha_1(j);0;0;b*alpha_1(j);-b*alpha_1(j)];
    sys.E11(31:36,4,j)=       [b;-b;0;0;b;-b];
    sys.E11(37:42,4:5,j)=     [-b*alpha_4(j) b*(alpha_2(j)-gamma_2(j));b*alpha_4(j) b*(gamma_2(j)-alpha_2(j));0 0;0 0;-b*alpha_4(j) b*(alpha_2(j)-gamma_2(j));b*alpha_4(j) b*(gamma_2(j)-alpha_2(j))];
    sys.E11(43:48,4,j)=       [-b*alpha_1(j);b*alpha_1(j);0;0;-b*alpha_1(j);b*alpha_1(j)];
    sys.E11(49:54,4:5,j)=     [b*alpha_4(j) b*(gamma_2(j)-alpha_2(j));-b*alpha_4(j) b*(gamma_2(j)-alpha_2(j));0 0;0 0;b*alpha_4(j) b*(gamma_2(j)-alpha_2(j));-b*alpha_4(j) b*(alpha_2(j)-gamma_2(j))];
```

```
    sys.E11(55:60,4,j)=      [-b*alpha_4(j);b*alpha_4(j);0;0;-b*alpha_4(j);b*alpha_4(j)];
    sys.E11(61:66,4,j)=      [e_j(j);-e_j(j);0;0;e_j(j);-e_j(j)];
    sys.E11(67:72,4:5,j)=    [-e_j(j)*alpha_4(j) -e_j(j)*gamma_2(j)-b*beta_j(j)/f+b*alpha_2(j);e_j(j)*alpha_4(j) e_j(j)*gamma_2(j)+b*beta_j(j)/f-b*alpha_2(j);0 0;0 0;-e_j(j)*alpha_4(j) -e_j(j)*gamma_2(j)-b*beta_j(j)/f+b*alpha_2(j);e_j(j)*alpha_4(j) e_j(j)*gamma_2(j)+b*beta_j(j)/f-b*alpha_2(j)];

    sys.E2(13:18,:,j)=       [1; -1; 0; 0; 1; -1];

    sys.g5_mat(13:18,1,j)=  [1/b;-1/b;0;0;1/b;-1/b];
    sys.g5_mat(37:42,1,j)=  [1;-1;0;0;1;-1];
    sys.g5_mat(49:54,1,j)=  [-1;1;0;0;-1;1];
    sys.g5_mat(67:72,1,j)=  [e_j(j)/b;-e_j(j)/b;0;0;e_j(j)/b;-e_j(j)/b];
    sys.g5_mat(1:6,2:4,j)=  [-1 1 -min1;1 -1 0;0 0 0;0 0 0;-1 1 -min1;1 -1 Max1];
    sys.g5_mat(7:12,2:4,j)= [-alpha_1(j) alpha_1(j) -min2;alpha_1(j) -alpha_1(j) 0;0 0 0;0 0 0;-alpha_1(j) alpha_1(j) -min2;alpha_1(j) -alpha_1(j) Max2];
    sys.g5_mat(13:18,2:4,j)=[alpha_4(j) 0 -eps;-alpha_4(j) 0 Max3;0 0 0;0 0 0;alpha_4(j) 0 -min3;-alpha_4(j) 0 Max3];
    sys.g5_mat(19:24,2:4,j)=[b -b -eps;-b b Max4;0 0 0;0 0 0;b -b -min4;-b b Max4];
    sys.g5_mat(25:30,2:4,j)=[-b*alpha_1(j) b*alpha_1(j) -min5;b*alpha_1(j) -b*alpha_1(j) 0;0 0 0;0 0 0;-b*alpha_1(j) b*alpha_1(j) -min5;b*alpha_1(j) -b*alpha_1(j) Max5];
    sys.g5_mat(31:36,2:4,j)=[-b b -min6;b -b 0;0 0 0;0 0 0;-b b -min6;b -b Max6];
    sys.g5_mat(37:42,2:4,j)=[b*alpha_4(j) 0 -eps;-b*alpha_4(j) 0 Max7;0 0 0;0 0 0;b*alpha_4(j) 0 -min7;-b*alpha_4(j) 0 Max7];
    sys.g5_mat(43:48,2:4,j)=[b*alpha_1(j) -b*alpha_1(j) -eps;-b*alpha_1(j) b*alpha_1(j) Max8;0 0 0;0 0 0;b*alpha_1(j) -b*alpha_1(j) -min8;-b*alpha_1(j) b*alpha_1(j) Max8];
    sys.g5_mat(49:54,2:4,j)=[-b*alpha_4(j) 0 -min9;b*alpha_4(j) 0 0;0 0 0; 0 0 0;-b*alpha_4(j) 0 -min9;b*alpha_4(j) 0 Max9];
    sys.g5_mat(55:60,2:4,j)=[b*alpha_4(j) -b*alpha_4(j) -min10;-b*alpha_4(j) b*alpha_4(j) 0;0 0 0;0 0 0;b*alpha_4(j) -b*alpha_4(j) -min10;-b*alpha_4(j) b*alpha_4(j) Max10];
    sys.g5_mat(61:66,2:4,j)=[-e_j(j) e_j(j) -min11;e_j(j) -e_j(j) 0;0 0 0;0 0 0;-e_j(j) e_j(j) -min11;e_j(j) -e_j(j) Max11];
    sys.g5_mat(67:72,2:4,j)=[e_j(j)*alpha_4(j) 0 -eps;-e_j(j)*alpha_4(j) 0 Max12;0 0 0;0 0 0;e_j(j)*alpha_4(j) 0 -min12;-e_j(j)*alpha_4(j) 0 Max12];
    sys.g5(:,:,j)=sys.g5_mat(:,:,j)*[rho_max;tau_rj(j);tau_H;1];
end

sys.E3(1:6,1)=       [eps-min1;-Max1;-Max1;min1;-min1;Max1];
sys.E3(7:12,2)=      [eps-min2;-Max2;-Max2;min2;-min2;Max2];
sys.E3(13:18,3)=     [min3-eps;Max3;-Max3;min3;-min3;Max3];
sys.E3(19:24,4)=     [min4-eps;Max4;-Max4;min4;-min4;Max4];
sys.E3(25:30,5)=     [eps-min5;-Max5;-Max5;min5;-min5;Max5];
sys.E3(31:36,6)=     [eps-min6;-Max6;-Max6;min6;-min6;Max6];
```

```
sys.E3(37:42,7)=     [min7-eps;Max7;-Max7;min7;-min7;Max7];
sys.E3(43:48,8)=     [min8-eps;Max8;-Max8;min8;-min8;Max8];
sys.E3(49:54,9)=     [eps-min9;-Max9;-Max9;min9;-min9;Max9];
sys.E3(55:60,10)=    [eps-min10;-Max10;-Max10;min10;-min10;Max10];
sys.E3(61:66,11)=    [eps-min11;-Max11;-Max11;min11;-min11;Max11];
sys.E3(67:72,12)=    [min12-eps;Max12;-Max12;min12;-min12;Max12];

sys.E4(3:6,1)=           [1;-1;1;-1];
sys.E4(9:12,2)=          [1;-1;1;-1];
sys.E4(13:18,1:3)=       [-1 1 0;1 -1 0;0 0 1;0 0 -1;-1 1 1;1 -1 -1];
sys.E4(21:24,4)=         [1;-1;1;-1];
sys.E4(27:30,5)=         [1;-1;1;-1];
sys.E4(33:36,6)=         [1;-1;1;-1];
sys.E4(37:42,5:7)=       [1 -1 0;-1 1 0;0 0 1;0 0 -1;1 -1 1;-1 1 -1];
sys.E4(45:48,8)=         [1;-1;1;-1];
sys.E4(49:54,4:9)=       [-1 0 0 0 1 0;1 0 0 0 -1 0;0 0 0 0 0 1;0 0 0 0 0 -1;-1 0 0 0 1 1;1 0 0 0 -1 -1];
sys.E4(57:60,10)=        [1;-1;1;-1];
sys.E4(63:66,11)=        [1;-1;1;-1];
sys.E4(67:72,10:12)=     [1 -1 0;-1 1 0;0 0 1;0 0 -1;1 -1 1;-1 1 -1];

%% add state & input constraints:
% d-a>=0
% a>=0
% 0<=rho<=rho_max
% sigma>=0
% u>=0

for j=1:J
    E1tot(:,:,j)=[sys.E1(:,:,j);-eye(6);0 0 0 0 1 0;0 0 0 0 0 0;0 0 0 0 0 0];
    E11tot(:,:,j)=[sys.E11(:,:,j);zeros(6+1+2,6)];
    E2tot(:,:,j)=[sys.E2(:,:,j);zeros(6+1,1);-1;1];
    g5tot(:,:,j)=[sys.g5(:,:,j);zeros(6,1);rho_max;0;10];
end
sys.E1=E1tot;
sys.E11=E11tot;
sys.E2=E2tot;
sys.g5=g5tot;

sys.E3=[sys.E3;zeros(6+1+2,12)];
sys.E4=[sys.E4;zeros(6+1+2,12)];


end
```

B-3 Combine MLD model for all stations

```
function sys=buildsysk_fullmodel(sys_small,J)

% full state matrices
matr1=sys_small.A1(:,:,2);
matr2=sys_small.B1(:,:,2);
matr3=sys_small.B2(:,:,2);
matr4=sys_small.B3;%(:,:,2);
matr5=eye(size(sys_small.A11(:,:,2)));
matr6=sys_small.A11(:,:,2);

sys.A1=[matr1 zeros(size(sys_small.A1(:,:,2), 1), size(sys_small.A1(:,:,2), 2)*(J-2))];
sys.B1=[matr2 zeros(size(sys_small.B1(:,:,2), 1), size(sys_small.B1(:,:,2), 2)*(J-2))];
sys.B2=[matr3 zeros(size(sys_small.B2(:,:,2), 1), size(sys_small.B2(:,:,2), 2)*(J-2))];
sys.B3=[matr4 zeros(size(sys_small.B3, 1), size(sys_small.B3, 2)*(J-2))];
B4mat=[matr5 zeros(size(sys_small.A11(:,:,2), 1), size(sys_small.A11(:,:,2), 2)*(J-2))];
B4_vert=sys_small.B4(:,:,2);
sys.A11=matr6;
for j=3:J
    matr1=[sys_small.A11(:,:,j)*matr1 sys_small.A1(:,:,j)];
    sys.A1=[sys.A1;matr1 zeros(size(sys_small.A1(:,:,j), 1), size(sys_small.A1(:,:,j), 2)*(J-j))];
    matr2=[sys_small.A11(:,:,j)*matr2 sys_small.B1(:,:,j)];
    sys.B1=[sys.B1;matr2 zeros(size(sys_small.B1(:,:,j), 1), size(sys_small.B1(:,:,j), 2)*(J-j))];
    matr3=[sys_small.A11(:,:,j)*matr3 sys_small.B2(:,:,j)];
    sys.B2=[sys.B2;matr3 zeros(size(sys_small.B2(:,:,j), 1), size(sys_small.B2(:,:,j), 2)*(J-j))];
    matr4=[sys_small.A11(:,:,j)*matr4 sys_small.B3];
    sys.B3=[sys.B3;matr4 zeros(size(sys_small.B3, 1), size(sys_small.B3, 2)*(J-j))];
    matr5=[sys_small.A11(:,:,j)*matr5 eye(size(sys_small.A11(:,:,j)))];
    B4mat=[B4mat;matr5 zeros(size(sys_small.A11(:,:,j), 1), size(sys_small.A11(:,:,j), 2)*(J-j))];
    B4_vert=[B4_vert;sys_small.B4(:,:,j)];

    matr6=sys_small.A11(:,:,j)*matr6;
    sys.A11=[sys.A11;matr6];
end
sys.B4=B4mat*B4_vert;

% full constraint matrices
matr7=sys_small.A1(:,:,2);
matr8=sys_small.B1(:,:,2);
matr9=sys_small.B2(:,:,2);
matr10=sys_small.B3;%(:,:,2);
matr11=eye(size(sys_small.E11(:,:,3),2));
```

```
sys.E1=[zeros(size(sys_small.E1(:,:,2), 1), size(sys_small.E1(:,:,2), 2)*(J-1)); sys_small.E11(:,:,3)*matr7 zeros(size(sys_small.E1(:,:,2), 1), size(sys_small.E1(:,:,2), 2)*(J-2))];
sys.E2=[zeros(size(sys_small.E2(:,:,2), 1), size(sys_small.E2(:,:,2), 2)*(J-1)); sys_small.E11(:,:,3)*matr8 zeros(size(sys_small.E2(:,:,2), 1), size(sys_small.E2(:,:,2), 2)*(J-2))];
sys.E3=[zeros(size(sys_small.E3, 1), size(sys_small.E3, 2)*(J-1)); sys_small.E11(:,:,3)*matr9 zeros(size(sys_small.E3, 1), size(sys_small.E3, 2)*(J-2))];
sys.E4=[zeros(size(sys_small.E4, 1), size(sys_small.E4, 2)*(J-1)); sys_small.E11(:,:,3)*matr10 zeros(size(sys_small.E4, 1), size(sys_small.E4, 2)*(J-2))];
EB4_mat=[zeros(size(sys_small.E11(:,:,3), 1), size(sys_small.E11(:,:,3), 2)*(J-1)); sys_small.E11(:,:,3)*matr11 zeros(size(sys_small.E11(:,:,3), 1), size(sys_small.E11(:,:,3), 2)*(J-2))];
EB4_vert=[sys_small.B4(:,:,2);sys_small.B4(:,:,3)];
for j=4:J
    matr7=[sys_small.A11(:,:,j-1)*matr7 sys_small.A1(:,:,j)];
    sys.E1=[sys.E1;sys_small.E11(:,:,j)*matr7 zeros(size(sys_small.E1(:,:,j), 1), size(sys_small.E1(:,:,j), 2)*(J+1-j))];
    matr8=[sys_small.A11(:,:,j-1)*matr8 sys_small.B1(:,:,j)];
    sys.E2=[sys.E2;sys_small.E11(:,:,j)*matr8 zeros(size(sys_small.E2(:,:,j), 1), size(sys_small.E2(:,:,j), 2)*(J+1-j))];
    matr9=[sys_small.A11(:,:,j-1)*matr9 sys_small.B2(:,:,j)];
    sys.E3=[sys.E3;sys_small.E11(:,:,j)*matr9 zeros(size(sys_small.E3, 1), size(sys_small.E3, 2)*(J+1-j))];
    matr10=[sys_small.A11(:,:,j-1)*matr10 sys_small.B3];
    sys.E4=[sys.E4;sys_small.E11(:,:,j)*matr10 zeros(size(sys_small.E4, 1), size(sys_small.E4, 2)*(J+1-j))];
    matr11=[sys_small.A11(:,:,j-1)*matr11 eye(size(sys_small.E11(:,:,j),2))];
    EB4_mat=[EB4_mat;sys_small.E11(:,:,j)*matr11 zeros(size(sys_small.E11(:,:,j), 1), size(sys_small.E11(:,:,j), 2)*(J+1-j))];
    EB4_vert=[EB4_vert;sys_small.B4(:,:,j)];
end
sys.EB4=EB4_mat*EB4_vert;

diagE1=sys_small.E1(:,:,2);
diagE2=sys_small.E2(:,:,2);
diagE3=sys_small.E3;%(:,:,2);
diagE4=sys_small.E4;%(:,:,2);
for j=3:J
    diagE1=blkdiag(diagE1,sys_small.E1(:,:,j));
    diagE2=blkdiag(diagE2,sys_small.E2(:,:,j));
    diagE3=blkdiag(diagE3,sys_small.E3);%(:,:,j));
    diagE4=blkdiag(diagE4,sys_small.E4);%(:,:,j));
end
sys.E1=sys.E1+diagE1;
sys.E2=sys.E2+diagE2;
sys.E3=sys.E3+diagE3;
sys.E4=sys.E4+diagE4;

```

```
sys.g5=sys_small.g5(:,:,2);
matr12=eye(size(sys_small.E11(:,:,2),2));
sys.Ex1=sys_small.E11(:,:,2)*matr12;
for j=3:J
    sys.g5=[sys.g5;sys_small.g5(:,:,j)];
    matr12=sys_small.A11(:,:,j-1)*matr12;
    sys.Ex1=[sys.Ex1;sys_small.E11(:,:,j)*matr12];
end

end
```

B-4 Build MIQP model

```
function [MILPsys,M2,MB4,Mx1]=MIQP_sysk_fullmodel6st(sys,Np,Nc,xkmin1,
    x1_k,rho_max,x_eq,u_eq,delta_eq,z_eq,J,P,Qx,Qu)
%% 2-norm
Nx=size(sys.A1,2);
Nx1=size(x1_k,1);
Nu=size(sys.B1,2);                                                            % B1 not
    dependent on j
Ndelta=size(sys.E3,2);                                                    % E3 not
    dependent on j
Nz=size(sys.B3,2);                                                            % B3 and E4
    not dependent on j
Nrest=size(sys.B4,2);

M2 = [];                                                                                   %
    A1
for i=1:Np
    M2=[M2;sys.A1^i];
end

MB4 = [];                                                                                 %
    B4
sum=0;
for i=1:Np
    sum=sum+sys.A1^(i-1);
    MB4=[MB4; sum];
end
MB4=MB4*sys.B4;

matrt1 = [];
matrt3 = [];
matrt4 = [];

T1 = [];
T3 = [];
Mx1 = [];

for i=1:Np
    matrt1=[sys.A1^(i-1)*sys.B1 matrt1];
    T1=[T1;matrt1 zeros(size(sys.B1, 1), size(sys.B1, 2)*(Np-i))];
                                             %T1=Tu
    matrt3=[sys.A1^(i-1)*sys.B3 matrt3];
    T3=[T3;matrt3 zeros(size(sys.B3, 1), size(sys.B3, 2)*(Np-i))];
                                             %T3=Tz
    matrt4=[sys.A1^(i-1)*sys.A11 matrt4];
```

```
    Mx1=[Mx1;matrt4 zeros(size(sys.A11, 1), size(sys.A11, 2)*(Np-i))]; %Mx_j-1(k)
end

% M1=[T1*Ku T2*Kdelta T3*Kz] % M1
M1=[T1 zeros(size(T3)) T3];


% constraints:
E1hat=[];
Ex1hat=[];
E2hat=[];
E3hat=[];
E4hat=[];
for i=1:Np
    E1hat=blkdiag(E1hat, sys.E1);
    E2hat=blkdiag(E2hat, sys.E2);
    E3hat=blkdiag(E3hat, sys.E3);
    E4hat=blkdiag(E4hat, sys.E4);
    Ex1hat=blkdiag(Ex1hat, sys.Ex1);
end

E1hat=[E1hat, zeros(size(E1hat,1),Nx);zeros(Nx+(J-1),size(E1hat,2)),kron(eye(J-1),[0 0 0 0 1 0;-eye(Nx1)])]; % E1hat (5=J-1)

E2hat=[E2hat; zeros(Nx+1*(J-1),size(E2hat,2))]; % E2hat
E3hat=[E3hat; zeros(Nx+1*(J-1),size(E3hat,2))]; % E3hat
E4hat=[E4hat; zeros(Nx+1*(J-1),size(E4hat,2))]; % E4hat
Ex1hat=[Ex1hat; zeros(Nx+1*(J-1),size(Ex1hat,2))]; % E11hat

g5hat=[];
EB4hat=[];
for i=1:Np
    g5hat=[g5hat; sys.g5];
    EB4hat=[EB4hat;sys.EB4];
end
g5hat=[g5hat; repmat([rho_max;zeros(Nx1,1)],J-1,1)]; % g5hat
EB4hat=[EB4hat; zeros(Nx+1*(J-1),1)]; % F3

% Cost function 2-norm
Qu_hat=1/2*kron(eye(Np),Qu);
Qdelta_hat=1/2*kron(eye(Np),zeros(Ndelta,Ndelta));
Qz_hat=1/2*kron(eye(Np),zeros(Nz,Nz));
Qx_hat=blkdiag(1/2*kron(eye((Np)),Qx),1/2*zeros(size(P)));
u_eq_hat=u_eq(:);
delta_eq_hat=delta_eq(:);
z_eq_hat=z_eq(:);
x_eq_hat=x_eq(:);
```

```
% Full contraints
F1=E1hat*[zeros(Nx, size(M1, 2)); M1]+[E2hat E3hat E4hat];
                                                     % F1, removed Ku,Kz,Kdelta
F2=g5hat-EB4hat-E1hat*[zeros(Nx,1); MB4]-(Ex1hat+E1hat*[zeros(Nx,Nx1*Np);
      Mx1])*x1_k(:);     % F2
F3=-E1hat*[eye(Nx); M2];                                     % F3

% Full minimization matrices
S1=[zeros(Nx,size(M1,2)); M1]'*Qx_hat*[zeros(Nx,size(M1,2)); M1]+blkdiag(
     Qu_hat, Qdelta_hat, Qz_hat);
S2=([zeros(Nx,1);MB4]+[zeros(Nx,Nx1*Np);Mx1]*x1_k(:)-x_eq_hat)'*Qx_hat*[
     zeros(Nx,size(M1,2));M1]-[u_eq_hat'*Qu_hat delta_eq_hat'*Qdelta_hat
     z_eq_hat'*Qz_hat];
S3=[eye(Nx); M2]'*Qx_hat*[zeros(Nx,size(M1,2));M1];
S2_tot=2*(S2+xkmin1(:)'*S3);

% Final matrices
Frho2=F2+F3*xkmin1;

MILPsys.Q=sparse(S1);
MILPsys.c=S2_tot;
MILPsys.A=sparse(F1);
MILPsys.b=Frho2;
MILPsys.bin_part =(Nu*(Np)+1):((Nu+Ndelta)*(Np));
end
```

Bibliography

[1] J. Komenda, S. Lahaye, J.-L. Boimond, and T.J.J. van den Boom. "Max-plus algebra in the history of discrete event systems". In: *Annual Reviews in Control* 45 (2018), pp. 240–249.

[2] T.J.J van den Boom and B de Schutter. "MPC of implicit switching max-plus-linear discrete event systems-Timing aspects". In: *2006 8th International Workshop on Discrete Event Systems*. IEEE. 2006, pp. 457–462.

[3] B De Schutter and TJJ Van Den Boom. "Model predictive control for max-min-plus-scaling systems-efficient implementation". In: *Sixth International Workshop on Discrete Event Systems, 2002. Proceedings.* IEEE. 2002, pp. 343–348.

[4] S. Safaei Farahani. *Approximation methods in stochastic max-plus systems*. PhD thesis, Delft University of Technology, 2012.

[5] T.J.J. van den Boom, M. van den Muijsenberg, and B. de Schutter. "Model predictive scheduling of semi-cyclic discrete-event systems using switching max-plus linear models and dynamic graphs". In: *Discrete Event Dynamic Systems* 30.4 (2020), pp. 635–669.

[6] T.J.J. van den Boom, B. de Schutter, and A. Gupta. *Max-Min-Plus-Scaling Discrete-Event Systems: Modelling, Control and Scheduling*. Internal report, 2021.

[7] M. Abdelmoumni. *Max-plus linear parameter varying systems: A framework solvability*. 2021.

[8] B. de Schutter and T.J.J. van den Boom. "Max-plus algebra and max-plus linear discrete event systems: An introduction". In: *2008 9th International Workshop on Discrete Event Systems*. IEEE. 2008, pp. 36–42.

[9] W.P.M.H. Heemels, B. de Schutter, and A. Bemporad. "Equivalence of hybrid dynamical models". In: *Automatica* 37.7 (2001), pp. 1085–1091. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(01)00059-0. URL: https://www.sciencedirect.com/science/article/pii/S0005109801000590.

[10] B. de Schutter and T.J.J. van den Boom. "MPC for continuous piecewise-affine systems". In: *Systems & Control Letters* 52.3-4 (2004), pp. 179–192. DOI: 10.1016/j.sysconle.2003.11.010.

[11] J.B. Rawlings, D.Q Mayne, M.M Diehl, and S. Barbara. *Model Predictive Control: Theory, Computation, and Design 2nd Edition*. Nob Hill Publishing, 2020, pp. 112, 113, 124.

[12] F. Borrelli, A. Bemporad, and M. Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2016, pp. 190–193, 244–246.

[13] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.

[14] S. Lin, B. de Schutter, Y. Xi, and H. Hellendoorn. "Model predictive control for urban traffic networks via MILP". In: *Proceedings of the 2010 American control conference*. IEEE. 2010, pp. 2272–2277.

[15] B. de Schutter and W.P.M.H. Heemels. "Modeling and Control of Hybrid Systems". In: *Lecture notes for the course SC4160*. Delft Center for Systems and Control. 2015, pp. 111–112.

[16] S. Skogestad and I. Postlethwaite. *Multivariable feedback control: analysis and design*. John Wiley & sons, 2005, pp. 253–259.

Glossary

List of Acronyms

MMPS	Max-Min-Plus-Scaling
MPC	Model predictive control
MPL	Max-plus-linear
ELC	Extended linear complementarity
PWA	Piecewise-affine
DT	discrete-time
DE	discrete-event
LQR	Linear-quadratic regulator
MIQP	Mixed integer quadratic programming
MILP	Mixed integer linear programming
MLD	Mixed logical dynamical
CLF	Control Lyapunov function

List of Symbols

Δ	Deviation of a variable for the original equilibrium
$\ell(x, u)$	Stage cost
κ_N	Control law at stage N
$\mathbb{R}$	Set of real numbers
$\mathbb{R}_\varepsilon$	Set of real numbers including $-\infty$
$\mathbb{R}_c$	Set of real numbers including $-\infty$ and $+\infty$
$\mathbb{R}_\mathrm{T}$	Set of real numbers including ∞
$\mathbb{U}$	Input constraint set
$\mathbb{X}$	State constraint set

$\mathbb{X}_f$	Terminal constraint set
$\mathbb{Z}$	System constraint region
$\mathcal{B}$	Ball in $\mathbb{R}^n$ of unit radius
$\mathcal{X}_N$	Set of feasible states for optimal control problem at stage N
$J(x_0, \mathbf{u})$	MPC cost function
K	Optimal controller gain
N_p	Prediction horizon
u_t	Control input at time(DT)/event(DE) t
$V_f(x)$	Terminal cost
$V_N(x_0, \mathbf{u})$	MPC objective function
$V_N^0(x)$	MPC optimal value function
$\oplus$	Max-plus addition (= maximization)
$\otimes$	Max-plus multiplication (= addition)