İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

A MATLAB TOOLBOX for HYBRID SYSTEMS

M.Sc. Thesis by

Veysel Gürkan ANIK, Ir.

Department : **Electrical Engineering**

Programme : **Control and Automation Engineering**

JUNE 2008

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

A MATLAB TOOLBOX
for
HYBRID SYSTEMS

M.Sc. Thesis by

Veysel Gürkan ANIK, Ir.

(504051119)

Date of Submission : **5 May 2008**

Date of Examin : **11 June 2007**

Supervisor : **Prof. Dr. Leyla GÖREN (İ.T.Ü.)**

Members of the Examining Committee **Doç. Dr. Mehmet Turan SÖYLEMEZ (İ.T.Ü.)**

Yrd. Doç. Dr. Neslihan Serap ŞENGÖR (İ.T.Ü.)

JUNE 2008

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

HİBRİT SİSTEMLER
için
BİR MATLAB ARAÇ KUTUSU

YÜKSEK LİSANS TEZİ

Müh. Veysel Gürkan ANIK

(504051119)

Tezin Enstitüye Verildiği Tarih : 5 Mayıs 2008

Tezin Savunulduğu Tarih : 11 Haziran 2007

Tez Danışmanı : Prof. Dr. Leyla GÖREN (İ.T.Ü.)

Diğer Jüri Üyeleri Doç. Dr. Mehmet Turan SÖYLEMEZ (İ.T.Ü.)

Yrd. Doç. Dr. Neslihan Serap ŞENGÖR (İ.T.Ü.)

HAZİRAN 2008

ACKNOWLEDGEMENT

I would like to thank my supervisor Prof. Dr. Leyla GOREN for her constant encouragement, help for my studies and support my decision to study a year in TU Delft as an exchange student. I also would like to thank my supervisor in TU Delft; Prof. dr.ir. Bart De SCHUTTER; and my daily supervisor in TU Delft; Dr.ir. Ton van den BOOM; because their advises help me to look at my master thesis theme in different aspects and to understand it. I am grateful for my family's understanding and supporting my study. I would not be here unless their support. I would like to thank ITU EU Centre and my department for my study as an exchange student in the Netherlands. My master study is made possible by the financial support of TUBITAK.

June 2008 Veysel Gürkan ANIK

TABLE OF CONTENTS

ABBREVATIONS

DUT	:	Delft University of Technology
ELC	:	Extended Linear Complementarity system
LC	:	Linear Complementarity system
MLD	:	Mixed Logical Dynamical systems
MPC	:	Model Predictive Control
MPL	:	Max-plus-linear system
MMPS	:	Max-min-plus-scaling system
PWA	:	Piecewise-affine system
MPT	:	Multi-Parametric Toolbox

LIST OF FIGURES

A MATLAB TOOLBOX for HYBRID SYSTEMS

SUMMARY

Hybrid systems contain both analog (continuous) and logical (discrete, switching) dynamics. The class of discrete event systems essentially consists of systems that contain a finite number of resources (e.g. machines, communication channels or processors) that are shared by several users(e.g. product types, information packets or jobs) all of which contribute to the achievement of some common goal(e.g. the assembly of products, the end-to-end transmission of a set of information packets or a parallel computation).

In the first main chapter, we will introduce the Max-Plus Algebra and Max-Plus-Linear (MPL) systems. Also in this part we will explain the differences between Max-Plus-Linear systems with other discrete event system modeling tools like Automata Theory and Petri Net approach.

In the second main chapter of this master thesis, we will consider some specific subclasses of hybrid systems and their relations: Piecewise Affine systems (PWA) , Mixed Logical Dynamical (MLD) systems, Linear Complementarity (LC) systems, Extended Linear Complementarity (ELC) systems and Max-Min-Plus-Scaling (MMPS) systems.

For both MPL systems and the mentioned subclasses of hybrid systems we will consider the implementation of the model predictive control(MPC) scheme.

In the last main chapter we will explain developed functions with examples. These function can be grouped in three main groups. The first group consists of functions to convert hybrid system subclasses to each other. The second group of functions is used to implement Mr. Frau's and Mr. Benschop's functions to our toolbox. The last group of functions aims to build an general model predictive controller algorithm for Max-Min-Plus-Scalar (MMPS) systems with limitations on input.

HİBRİT SİSTEMLER için BİR MATLAB ARAÇ KUTUSU

ÖZET

Hibrit sistemler hem analog (sürekli) hem de lojik (ayrık, anahtarlamalı) dinamikler içerir. Ayrık olay sistemleri sınıfı, makine, iletişim kanalları ve işlemciler gibi sınırlı sayıda kaynak içeren sistemleri içerir. Bu kaynaklar; ürün çeşitleri, iletişim paketleri ve iş gibi çeşitli kullanıcılar arasında paylaşılır. Bu kullanıcılar çeşitli ortak hedeflerin sağlanması için; bir dizi iletişim paketinin başlangıçtan sona kadar iletimi veya paralel hesaplama gibi; çalışır.

Tezin ilk bölümünde Max-Plus cebri ve Max-Plus-Lineer (MPL) sistemler incelenmiştir. Ayrıca bu bölümde Max-Plus-Lineer sistemler ile Automata theorisi ve Petri Ağı yaklaşımları gibi diğer ayrık olay sistemi modelleme araçları arasındaki farklar açıklanmıştır.

Tezin ikinci bölümünde hibrit sistemlerin çeşitli alt sınıflarını ve bu alt sınıfların birbirleri ile ilişkilerini incelenmiştir. Bu alt sınıflar sırası ile Piecewise Affine (parçalı ilgin) (PWA) sistemler, Mixed Logical Dynamical (karışık lojik dinamik) (MLD) sistemler, Linear Complementarity (lineer tümlemeli) (LC) sistemler, Extended Linear Complementarity (genişletilmiş lineer tümlemeli) sistemler ve Max-Min-Plus-Scaling (MMPS) sistemlerdir.

Hem Max-Plus-Linear (MPL) sistemler hem de yukarıda belirtilen hibrit sistem alt sınıfları için model öngörmeli kontrolörün uygulamaları incelenmiştir.

En son ana bölümde geliştirilmiş fonksiyonlar örneklerle açıklanmaktadır. Bu fonksiyonlar üç ana grup altında toplanabilir. İlk grup hybrit sistem alt sınıflarını birbirlerine çeviren fonksiyonları içermektedir. İkinci grup ise A. Frau ve G.J. Benschop'un çalışmalarını birleştiren fonksiyonlardan oluşur. Son gruptaki fonksiyonlar ise giriş işaretleri üstünde sınırlamalar içeren Max-Min-Plus-Scaling (MMPS) sistemler için genel bir model öngörücülü kontrolör algoritması geliştirilmesini amaçlar.

1. INTRODUCTION

Hybrid systems contain both analog (continuous) and logical (discrete, switching) dynamics. Typical examples are manufacturing systems,telecommunication and computer networks,traffic control systems, digital circuits and logistics systems.

The class of discrete-event systems essentially consists of man-made systems that contain a finite number of resources(e.g. machines, communication channels or processors) that are shared by several users(e.g. product types, information packets or jobs) all of which contribute to the achievement of some common goal(e.g. the assembly of products, the end-to-end transmission of a set of information packets or a parallel computation).

This project aims to develop a MATLAB toolbox for a number of classes of hybrid and discrete event systems using previously built functions and algorithms.

The first class of system that we will consider are max-plus linear(MPL) systems. MPL systems are a subclass of discrete event systems for which the model becomes linear when formulated in the max-plus algebra, which has maximization and addition as its basic operations. Discrete-event systems in which only synchronization and no concurrency or choice occur, can be modeled using the maximization operations (corresponding to synchronization: a new operation starts as soon as all preceding operations have been finished) and addition (corresponding to durations: the finishing time of an operation equals the starting time plus duration). This leads to a description that is linear in the max-plus algebra.

In the second part of this project we consider some specific subclasses of hybrid systems: piecewise affine systems, mixed logical dynamical systems, complementarity systems, extended complementarity systems and max-min-plus-scaling systems. Note that some of these classes are equivalent,

possibly under mild additional assumptions related to well-posedness and boundedness of input,state,output or auxiliary variables.

For both MPL systems and the mentioned subclasses of hybrid systems we will consider the implementation of the model-predictive control(MPC) scheme. In the last decades MPC has shown to respond effectively to control demands imposed by tighter product quality specifications, increasing productivity demands, new environmental regulations and fast changes in the market. As a result, MPC is now widely accepted in the industry.

2. MAX-PLUS-LINEAR (MPL) SYSTEMS AS A MODELING APPROACH FOR DISCRETE EVENT SYSTEMS

2.1 Background of Discrete Event Systems

Most of the systems can be modeled with time difference equations like heating systems. These systems are called as *Time Systems* or *Time driven Systems* because they are driven by time. Unlike Time systems, a great majority of automation systems are not time-driven systems. These systems can not be modeled with time difference equations because the state of the system can change only and only if certain events which are time-independent occur. These systems which are driven by events are called as *Event driven Systems* or *Event Systems*. Checkers can be given as an example of event systems because only the play decisions can change the state of the game.

Event systems whose state's set are discrete are called as *Discrete Event Systems* (DES). Our previous example,checkers, can be given as an example of discrete event systems.

2.2 Automata Theory

Before definition of Automata we must define the term "Language". Let E be set of events of a discrete event system. A *language* is a set of finite-length strings of this E and its symbol is L. As a language can only contain finite-length strings of events, the number of elements of a language can be infinite. For example, we assume that our event set is $E_1 = \{a, b, c\}$. As the first language $L_1 = \{a, aa, aab\}$ has finite number of elements,the second language $L_2 = \{$All possible finite-length strings which begin with event a$\}$ has infinite elements. [1]

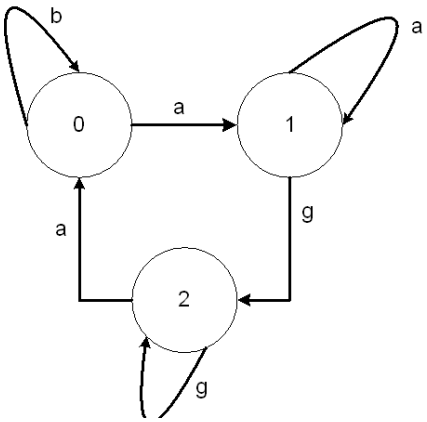

Figure 2.1: An Automaton example

Automata theory use the term ”state” which means an overall discrete status of the system like ”motor is on” or ”the phase a is on”. To find current state of the system it uses *tokens*.

Definition 2.1 *[1]A Deterministic Automaton, denoted by G, is a six-tuple*

$$G = (X, E, f, \Gamma, x_0, X_m)$$

where:

X *is the set of states*

E *is the finite set of events associated with the transitions in* G

$f : X \times E \to X$ *is the transition function*

$\Gamma : X \to 2^E$ *is the active event function (*2^E *means the set of all subsets of E.)*

x_0 *is the initial state*

$X_m \subseteq X$ *is the set of marked states.*

In Fig.2.1, we can see that $X = \{0,1,2\}$, $E = \{a,b,g\}$, $x_0 = 0$, $X_m = \{1\}$,

$$\begin{array}{lll} f(0,b) = 0 & f(1,a) = 1 & f(2,a) = 0 \\ f(0,a) = 1 & f(1,g) = 2 & f(2,g) = 2 \\ \Gamma(0) = \{a,b\} & \Gamma(1) = \{a,g\} & \Gamma(2) = \{a,g\} \end{array} \quad \textbf{(2.1)}$$

A deterministic Automata build two languages, *generated language* $L(G)$ and *marked language* $L_m(G)$. As the generated language consists of all strings s of events where $f(x_0,s)$ is defined, the marked language consists of all strings s of events where $f(x_0,s) = X_m$.

2.3 Petri Net Theory

Although states which are an overall discrete status of the system and events are used by automata theory, Petri net use *"places"* and *"transitions"*. A *place* can be defined as a status of a part of the system like "motor 1 on the phase b" however *transition* and *event* are similar. It is obvious that a set of all places create a *state*. [1]

The arcs of Petri nets are *bipartite*. This means that the arcs can go from places to transitions or from transitions to places. Weight of an arc determines the number of token which the arc transfers.

Definition 2.2 *[1]A Petri net graph is a weighted bipartite graph*

$$(P,T,w,A)$$

where

P *is the finite set of the places*

T *is the finite set of transitions*

$A \subseteq (P \times T) \cup (T \times P)$ *is the set of arcs from the places to transitions and from transitions to places*

$w : A \to \{1,2,3,...\}$ *is the weight function of the arcs*

In Figure2.2 it can be seen that P=$\{p_1,p_2,p_3,p_4\}$,$T = \{t_1,t_2,t_3\}$,A=$\{(p_1,t_1), (p_2,t_2),(p_3,t_3),(p_4,t_2),(t_1,p_2),(t_2,p_3),(t_3,p_1)\}$

$$\begin{aligned} &w(p_1,t_1) = 1 \quad w(p_2,t_2) = 2 \quad w(p_3,t_3) = 1 \\ &w(p_4,t_2) = 1 \quad w(t_1,p_2) = 1 \quad w(t_2,p_3) = 1 \\ &w(t_2,p_3) = 1 \quad w(t_3,p_1) = 2 \end{aligned} \tag{2.2}$$

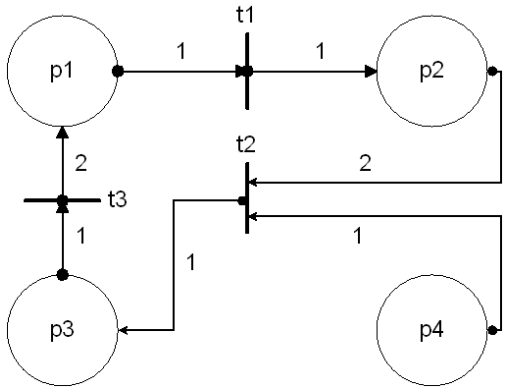

Figure 2.2: A Petri net example

As the transition t_1 occur,the transition gets 3 tokens but transfer only one token to p_3. On the other hand transition t_3 transfers 2 tokens to p_1 as it gets one token. This examples show us that the number of tokens can change as a transition occurs.

Petri net graphs can be modeled via state equation. Let the $k^{\underline{th}}$ state of the system be x(k)=[$p_1(k)$ $p_2(k)$ $p_3(k)$ ... $p_n(k)$] and $k^{\underline{th}}$ fired transition vector be u(k)=[$t_1(k)$ $t_2(k)$ $t_3(k)$ ... $t_m(k)$]. The state equation of the system can be written as

$$x(k+1) = x(k) + A * u(k) \tag{2.3}$$

$$a_{ij} = w(t_i, p_j) - w(p_j, t_i) \quad \textit{where } a_{ij} = (A)_{ij} \tag{2.4}$$

where A is incidence matrix. The incidence matrix A for the previous example is

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -2 & 1 & -1 \\ 2 & 0 & -1 & 0 \end{bmatrix} \tag{2.5}$$

2.4 Max-Plus-Linear (MPL) Systems as a Modeling Tool

Autonomous discrete event systems are discrete event systems where an event immediately occurs when its conditions are supplied. Therefore event times play an important role on the system. Event (occur) times can be used to analyze autonomous discrete event systems. Max-Plus-Linear (MPL) is developed for this purpose and it use event times. [2] [3]

2.4.1 Background of MPL System:Max-Plus Algebra

To analyze Max-Plus-Linear systems, we must study on Max-Plus algebra which is the core of these systems. We want to introduce Max-Plus algebra in this subsection.

Definition 2.3 *[3] A semi ring is a nonempty set* $\mathbf{R}$ *with the binary operations* $\otimes_R$ *and* $\oplus_R$ *where these binary operations must satisfy the following conditions:*

- $\oplus_R$ *is associative and commutative with the zero element* ε_R*;*
- $\otimes_R$ *is associative,distributive over* $\oplus_R$ *and has a unit element* e_R*;*
- e_R *is absorbing for* $\otimes_R$*;*

Definition 2.4 *[3] Max-Plus algebra is the set* $\mathbf{R}_{\max} := \mathbf{R} \cup \varepsilon$ *with following binary operations* $\oplus$ *and* $\otimes$ *where e=0 and* $\varepsilon = \infty$ *and is denoted as* $\mathbf{R}_{\max} = \{\mathbf{R}_{\max}, \oplus, \otimes, e, \varepsilon\}$.

$$a \oplus b = \max(a, b) \tag{2.6}$$

$$a \otimes b = a + b \tag{2.7}$$

The max-plus algebra have some algebraic properties:

- Commutativity:

$$\forall x, y, z \in \mathbf{R}_{max} \quad a \oplus b = b \oplus a \quad and \quad a \otimes b = b \otimes a$$

- Associativity:

$$\forall x, y, z \in \mathbf{R}_{max} \quad x \oplus (y \oplus z) = (x \oplus y) \oplus z \quad and \quad x \otimes (y \otimes z) = (x \otimes y) \otimes z$$

- Distributivity of $\otimes$ over $\oplus$:

$$\forall x, y, z \in \mathbf{R}_{max} \quad x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$$

- Existence of the zero element:

$$\forall x \in \mathbf{R}_{max} \quad x \oplus \varepsilon = x$$

- Existence of the unit element:

$$\forall x \in \mathbf{R}_{max} \quad x \otimes e = x$$

- The zero absorbing for $\otimes$:

$$\forall x \in \mathbf{R}_{max} \quad x \otimes \varepsilon = \varepsilon$$

- Idem potency of $\oplus$:

$$\forall x \in \mathbf{R}_{max} \quad x \oplus x = x$$

- $x^{\otimes n} =^{def} \underbrace{x \otimes x \otimes \ldots \otimes x}_{n times}$

The Matrix operations $\oplus$ and $\otimes$ are defined as:

- $\oplus$ operation on Matrices:

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, bij) \tag{2.8}$$

- $\otimes$ operation on Matrices where $A_{m \times l}$ and $B_{l \times n}$:

$$[A \otimes B]_{ij} = \bigoplus_{k=1}^{l} a_{ik} \otimes b_{kj} = \max_{k \in l}(a_{ik} + b_{kj}) \tag{2.9}$$

2.4.2 Max-Plus-Linear Systems

Due to Baccelli, the event systems which can be described as follows are called as *Max-Plus-Linear Systems*. [3]

$$\begin{aligned} x(k+1) &= A \otimes x(k) \oplus B \otimes u(k) \\ y(k) &= C \otimes x(k) \end{aligned} \tag{2.10}$$

In Figure 2.3, the raw material can be sent to machine A if and only if both of following conditions are satisfied.

- The previous raw material must have been sent to machine A at least 2 minutes before.
- The raw material must wait on the production line at least 1 minute.

The raw material can be sent to machine B if and only if all the following conditions are satisfied.

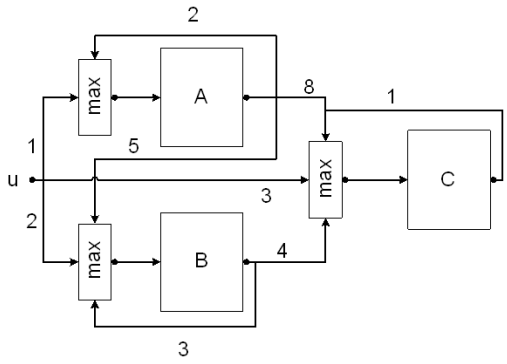

Figure 2.3: A MPL model of a production system

- The previous raw material must have been sent to machine A at least 5 minutes before.
- The previous raw material must have been sent to machine B at least 3 minutes before.
- The raw material must wait on the production line at least 2 minutes.

The raw material can be sent to machine C if and only if all the following conditions are satisfied.

- The previous raw material must have been sent to machine A at least 8 minutes before.
- The previous raw material must have been sent to machine B at least 4 minutes before.
- The previous raw material must have been sent to machine C at least 1 minute before.
- The raw material must wait on the production line at least 3 minutes.

Let assume that sending time to machine A,sending time to machine B and sending time to machine C are events x_1, x_2, x_3 respectively. So we can say that laying time of the raw material on the production line will be our input u. The production time,y, will be the sending time of raw material to machine C.

The system can be written as

$$\begin{aligned} x_1(k+1) &= max\{x_1(k)+2; u(k)+1\} \\ x_2(k+1) &= max\{x_1(k)+5; x_2(k)+3; u(k)+2\} \\ x_3(k+1) &= max\{x_1(k)+8; x_2(k)+4; x_3(k)+1; u(k)+3\} \\ y(k) &= x_3(k) \end{aligned} \tag{2.11}$$

or in Matrix form

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 2 & \varepsilon & \varepsilon \\ 5 & 3 & \varepsilon \\ 8 & 4 & 1 \end{bmatrix} \otimes x(k) \oplus \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \otimes u(k) \\ y(k) &= \begin{bmatrix} \varepsilon & \varepsilon & 0 \end{bmatrix} \otimes x(k) \end{aligned} \tag{2.12}$$

2.4.3 An Extension to MPL systems: Switching MPL Systems

An extension to MPL system model is required to model most of the systems because these systems have different operation modes. Switching MPL system model enable us to switch between these operation modes with the help of so called switching mechanism. So we can describe the switching max-plus-linear systems [4]

$$x(k) = A^{(l(k))} \otimes x(k-1) \oplus B^{(l(k))} \otimes u(k) \tag{2.13}$$

as the switching mechanism z(k) is described as

$$z(k) = \Phi(x(k-1), l(k-1), u(k), v(k)) \tag{2.14}$$

where l(k-1) is the previous mode and v(k) is additional variable.

3. HYBRID SYSTEMS

3.1 Introduction to Hybrid Systems

Hybrid systems contain both analog (continuous) and logical (discrete) dynamics. Typical examples of hybrid systems are manufacturing systems, communication and computer networks, traffic controls, digital circuits and logistic systems. Although there are many theories concerned linear differential systems and discrete event systems, a generalized theory about hybrid systems can not be found.

3.2 Subclasses of Hybrid Systems

Some methods are developed to analyze and control for some subclasses of hybrid systems. In this section we will introduce some subclasses of hybrid systems.

3.2.1 Piecewise-Affine (PWA) Systems

Piecewise affine (PWA) systems are described by Sontag as

$$\begin{array}{c} x(k+1) = A_i x(k) + B_i u(k) + f_i \\ y(k) = C_i x(k) + D_i u(k) + g_i \end{array} \quad for \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i \tag{3.1}$$

where Ω_i is a convex polyhedral. This convex polyhedral Ω_i is formed by inequalities in the input/state space. [5] [6]

Piecewise affine systems are simplest extension of linear systems which perform hybrid system behavior. PWA systems are one of the most analyzed subclasses of hybrid systems. Saturation (3.1) can be modeled as Piecewise affine system like

$$y(t) = \begin{cases} -3 & if \quad x(t) \leq -3 \\ x(t) & if \quad -3 < x(t) < 3 \\ 3 & if \quad x(t) \geq 3 \end{cases} \tag{3.2}$$

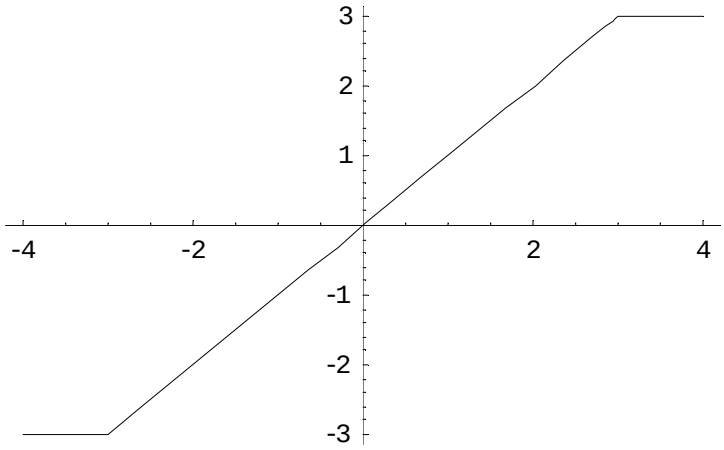

Figure 3.1: Saturation

3.2.2 Mixed Logical Dynamical (MLD) Systems

Mixed logical dynamical systems are analyzed by Bemporad and Morari as

$$x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \tag{3.3}$$

$$y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \tag{3.4}$$

$$E_1x(k) + E_2u(k) + E_3\delta(k) + E_4z(k) \leq g_5 \tag{3.5}$$

where $x(k) = \begin{bmatrix} x_t^T(k) & x_b^T(k) \end{bmatrix}^T$ with $x_t(k) \in R^{n_r}$ and $x_b(k) \in \{0,1\}^{n_b}$, where $z(k) \in R^{r_r}$ and $\delta(k) \in \{0,1\}^{r_b}$ are auxiliary variables. [5] [7]

3.2.3 Linear Complementarity (LC) Systems

Linear Complementarity systems are analyzed at first by Heemels and described as

$$x(k+1) = Ax(k) + B_1u(k) + B_2w(k) \tag{3.6}$$

$$y(k) = Cx(k) + D_1u(k) + D_2w(k) \tag{3.7}$$

$$v(k) = E_1x(k) + E_2u(k) + E_3w(k) + g_4 \tag{3.8}$$

$$0 \leq v(k) \perp w(k) \geq 0$$

with $v(k), w(k) \in R^s$ where v(k) and w(k) are orthogonal to each other. The variables v(k) and w(k) are called as complementarity variables. [5]

Heemels's example [8] (3.2) will give us an detailed explanation on "Linear Complementarity System". The left cart is attached to a wall by a spring. The

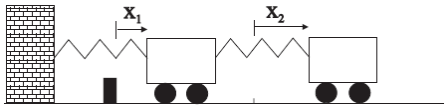

Figure 3.2: Heemels's example

motion of the left cart is constrained by a completely inelastic stop.

$$\begin{aligned} \dot{x_1}(t) &= x_3(t) \\ \dot{x_2}(t) &= x_4(t) \\ \dot{x_3}(t) &= -2x_1(t) + x_2(t) + u(t) \\ \dot{x_4}(t) &= x_1(t) - x_2(t) \\ y(t) &:= x_1(t) \\ 0 \leq u(k) &\perp y(k) \geq 0 \end{aligned} \tag{3.9}$$

where $u(k)$ is reaction force of the stop. The last condition is satisfied because the reaction force of the stop $u(t)$ exists only when $y(t) = 0$.

3.2.4 Extended Linear Complementarity (ELC) Systems

Extended linear complementarity(ELC) systems are analyzed by De Schutter and De Moor and described as

$$x(k+1) = Ax(k) + B_1u(k) + B_2d(k) \tag{3.10}$$

$$y(k) = Cx(k) + D_1u(k) + D_2d(k) \tag{3.11}$$

$$E_1x(k) + E_2u(k) + E_3w(k) \leq g_4 \tag{3.12}$$

$$\sum_{i=1}^{p} \prod_{j \in \Phi_i} (g_4 - E_1x(k) - E_2u(k) - E_3d(k))_j = 0 \tag{3.13}$$

where $d(k) \in R^r$ is a auxiliary variable. The last condition can be written as

$$\prod_{j \in \Phi_i} (g_4 - E_1x(k) - E_2u(k) - E_3d(k))_j = 0 \tag{3.14}$$

due of the inequality condition. [5]

The PWA system [5]

$$x(k+1) = \begin{cases} x(k) + u(k) & if \quad x(k) \geq 0 \\ -x(k) + u(k) & if \quad x(k) < 0 \end{cases} \tag{3.15}$$

can be remodeled as ELC system like

$$\begin{aligned} x(k+1) &= -x(k)+u(k)+2d(k) \\ -d(k) &\leq 0 \\ x(k)-d(k) &\leq 0 \\ 0 &= \big(x(k)-d(k)\big)\big(-d(k)\big) \end{aligned} \tag{3.16}$$

3.2.5 Max-Min-Plus-Scaling (MMPS) Systems

Max-Min-Plus-Scaling (MMPS) systems are analyzed by De Schutter and Van den Boom.

Definition 3.1 *[5] [2] [9] A Max-min-plus-scaling expression f of the variables $x_1,\ldots,x_n$ is defined as*

$$f := x_i \,|\, \alpha \,|\, max(f_k, f_l) \,|\, min(f_k, f_l) \,|\, f_k + f_l \,|\, \beta f_k \tag{3.17}$$

where f_k, f_l are MMPS expressions. Max-Min-Plus-Scaling (MMPS) systems are described as

$$\begin{aligned} x(k+1) &= M_x(x(k),u(k),d(k)) \\ y(k) &= M_y(x(k),u(k),d(k)) \\ M_c(x(k),u(k),d(k)) &\leq c \end{aligned} \tag{3.18}$$

where M_x, M_y, M_c are MMPS expressions.

The MMPS functions have some properties:

- Distribution of the addition over both minimum and maximum:

$$\begin{aligned} \min(f_1,f_2)+f_3 &= \min(f_1+f_3, f_2+f_3) \\ \max(f_1,f_2)+f_3 &= \max(f_1+f_3, f_2+f_3) \end{aligned} \tag{3.19}$$

- Multiplication:

$$\begin{aligned} \beta \min(f_1,f_2) &= \min(\beta f_1, \beta f_2) \\ \beta \max(f_1,f_2) &= \max(\beta f_1, \beta f_2) \end{aligned} \tag{3.20}$$

$$\begin{aligned} -\alpha \min(f_1,f_2) &= \max(-\alpha f_1, -\alpha f_2) \\ -\alpha \max(f_1,f_2) &= \min(-\alpha f_1, -\alpha f_2) \end{aligned} \tag{3.21}$$

- Nestings of min(max) operations can be simplified in the following way:

$$\min\Big(\min\big(\ldots,\big(\min(f_1,f_2),f_3\big)\ldots\big),f_k\Big) = \min(f_1,f_2,\ldots,f_k) \tag{3.22}$$

- Expressions in the form min()+...+min() can be reduced in the following way(it holds for maximization):

$$\sum_{i=1}^{l} \min(f_{ij}) = \min_{(j_1,...,j_l)\in\{1,..,k_1\}\times...\times\{1,..,k_l\}} (\sum_{i=1}^{l} f_{ij_i}) \tag{3.23}$$

- Minimization is distributive with respect to maximization and vice versa:

$$\begin{gathered} \min\big(\max(f_1,f_2),\max(f_3,f_4)\big) = \\ \max\big(\min(f_1,f_3),\min(f_1,f_4),\min(f_2,f_3),\min(f_2,f_4)\big) \\ \max\big(\min(f_1,f_2),\min(f_3,f_4)\big) = \\ \min\big(\max(f_1,f_3),\max(f_1,f_4),\max(f_2,f_3),\max(f_2,f_4)\big) \end{gathered} \tag{3.24}$$

3.3 Canonical forms of MMPS functions

All MMPS expressions can be written in canonical form. To write MMPS function into canonical forms will reduce the computational time.

Definition 3.2 *[10] An MMPS function is in conjunctive form if it is written as*

$$\min_{j\in 1,...,l} \big(\max_{i\in I_j}(\alpha_i^T x + b_i)\big) \tag{3.25}$$

or in disjunctive form if it is written as

$$\max_{j\in 1,...,l} \big(\min_{i\in I_j}(\alpha_i^T x + b_i)\big) \tag{3.26}$$

where $I_1,...,I_l \subseteq \{1,...,N\}$ are index sets and there are N components in the form $\alpha_i^T x + b_i$.

Definition 3.3 *[10] A level-n expression is an expression with n-1 nesting. The number n equals the maximum number of min and max operations encountered in each MMPS expression before arriving at an argument of the form $\alpha_i^T x + b_i$.*

Some properties about conjunctive and disjunctive forms are given:

- The expression $\max(f_1,...,f_k) + \min(g_1,...,g_l)$ can be written as:

$$\begin{gathered} \max(f_1,...,f_k) + \min(g_1,...,g_l) \\ = \max\big(\min(f_1+g_1,...,f_1+g_l),..., \\ \min(f_k+g_1,...,f_k+g_l)\big) \\ = \min\big(\max(f_1+g_1,...,f_k+g_1),..., \\ \min(f_1+g_l,...,f_k+g_l)\big) \end{gathered} \tag{3.27}$$

- A conjunctive form can be converted to disjunctive form and vice versa:

$$\begin{aligned}&\min\left(\max(f_{11},...,f_{1k_1}),...,\max(f_{l1},...,f_{lk_l})\right)\\&=\max\left(\min(f_{11},f_{21},...,f_{l1}),...,\min(f_{1k_1},f_{2k_2},...,f_{lk_l})\right)\end{aligned} \tag{3.28}$$

- The expression $\min(\max(),\max(),...,\min(),\min())$ can be easily written in conjunctive form:

$$\begin{aligned}&\min\left(\max(f_1,f_2),\min(f_3,f_4)\right)\\&=\min\left(\max(f_1,f_2),f_3,f_4\right)\\&=\min\left(\max(f_1,f_2),\max(f_3,f_3),\max(f_4,f_4)\right)\end{aligned} \tag{3.29}$$

- The expression $\max\Big(\min\left(\max(f_1,f_2),f_3\right),f_4\Big)$ can be written in conjunctive form:

$$\begin{aligned}&\max\Big(\min\left(\max(f_1,f_2),f_3\right),f_4\Big)\\&=\max\Big(\max\left(\min(f_1,f_3),\min(f_2,f_3),f_4\right)\Big)\\&=\max\left(\min(f_1,f_3),\min(f_2,f_3),f_4\right)\\&=\min\left(\max(f_1,f_2,f_4),\max(f_1,f_3,f_4),....,\max(f_3,f_4)\right)\\&=\min\left(\max(f_1,f_2,f_4),\max(f_3,f_4)\right)\end{aligned} \tag{3.30}$$

3.4 Equivalence of Hybrid System's subclasses

Previously we have said that there are methods to analyze and control of some hybrid system's subclasses. In order to analyze and control hybrid systems some generalized methods must be developed. Therefore we must show the relationships between hybrid system's subclasses. In this section we will analyze these relationships.

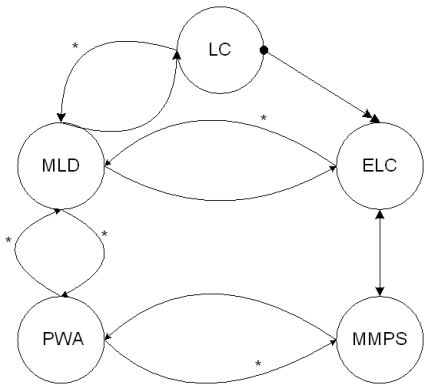

Figure 3.3: Graphical representation of the equivalences of hybrid systems. (*) means condition.

3.4.1 MLD and LC systems

Proposition 3.1 *[5]Every MLD system can be written as an LC system.*

Proof At first, we must remember the condition $\delta(k) \in \{0,1\}^{r_b}$ which implies $0 \geq \delta_i(k) \perp 1-\delta_i(k) \leq 0$. If the variable $v_1(k)$ is determined as $v_1(k) = e - \delta(k)$ where e denotes the vector for which all entries all equal to one,it can be easily shown that $\delta(k) \perp v1(k)$. It is indicated that binary constraints on x_b(k+1),u_b(k),y_b(k) are included in complementarity condition.

We can define $v_2(k)$ by using (3.4). Let us define $v_2(k)$ as

$$v_2(k) = g_5 - E_1 x(k) - E_2 u(k) - E_3 \delta(k) - E_4 z(k) \tag{3.31}$$

It can be seen $v_2(k) \geq 0$ since the inequality 3.5 . The inequality implies that a w_2(k) exists such that

$$0 \leq v_2(k) \perp w_2(k) \geq 0 \tag{3.32}$$

Auxiliary variable z(k) is not allowed in LC systems where only nonnegative complementarity variables are possible. Therefore, we must split the variable z(k) in its negative and positive parts.

$$\begin{aligned} z(k) &:= z^+(k) - z^-(k) \\ z^+(k) &= max(0; z(k)) \\ z^-(k) &= max(0; -z(k)) \end{aligned} \tag{3.33}$$

It is obvious that $0 \leq z^+(k) \perp z^-(k) \geq 0$. In addition to that, we add two extra auxiliary vectors $v_3(k) = z^+(k)$ and $v_4(k) = z^-(k)$.

At least we have written our MLD system as an LC system:

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 u(k) + \begin{bmatrix} B_2 & 0 & B_3 & -B_3 \end{bmatrix} w(k) \\ y(k) &= Cx(k) + D_1 u(k) + \begin{bmatrix} D_2 & 0 & D_3 & -D_3 \end{bmatrix} w(k) \\ \underbrace{\begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \\ v_4(k) \end{bmatrix}}_{=:v(k)} &= \begin{bmatrix} e \\ g_5 - E_1 x(k) - E_2 u(k) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -I & 0 & 0 & 0 \\ -E_3 & 0 & -E_4 & E_4 \\ 0 & 0 & 0 & I \\ 0 & 0 & I & 0 \end{bmatrix} \underbrace{\begin{bmatrix} \delta(k) \\ w_2(k) \\ z^+(k) \\ z^-(k) \end{bmatrix}}_{=:w(k)} \\ 0 \leq v(k) &\perp w(k) \geq 0 \end{aligned} \tag{3.34}$$

Proposition 3.2 *[5] Every LC system can be written as an MLD system, provided that the variables w(k) and v(k) are (component wise) bounded.*

Proof The complementarity condition say that $0 \leq v(k) \perp w(k) \geq 0$. To satisfy this condition one of $v_i(k)$ and $w_i(k)$ must be equal to zero for each $i \in \{1,...,s\}$ as the other variable is nonnegative. To produce $\delta(k) \in \{0,1\}^s$ we represent v(k) and w(k) as

$$\begin{array}{cc} w(k) \leq M_w \delta(k) & v(k) \leq M_v(e-\delta(k)) \\ w(k) \geq 0 & v(k) \geq 0 \end{array} \tag{3.35}$$

where M_w and M_v are diagonal matrices containing upper-bounds on w(k) and v(k) respectively. By setting z(k)=w(k) and replacing v(k) in the inequality 3.8. we can easily rewrite LC system as a MLD model

$$\begin{aligned} x(k+1) &= Ax(k)+B_1u(k)+B_2z(k) \\ y(k) &= Cx(k)+D_1u(k)+D_2z(k) \end{aligned} \tag{3.36}$$

$$\begin{bmatrix} 0 \\ E_1 \\ 0 \\ -E_1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ E_2 \\ 0 \\ -E_2 \end{bmatrix} u(k) + \begin{bmatrix} -M_w \\ M_w \\ 0 \\ 0 \end{bmatrix} \delta(k) + \begin{bmatrix} I \\ E_3 \\ -I \\ -E_3 \end{bmatrix} z(k) \leq \begin{bmatrix} 0 \\ M_v e - g_4 \\ 0 \\ g_4 \end{bmatrix}$$

3.4.2 LC and ELC systems

Proposition 3.3 *[5] Every LC system can be written as an ELC system*

Proof It can be written as

$$\begin{aligned} x(k+1) &= Ax(k)+B_1u(k)+B_2\underbrace{w(k)}_{=d(k)} \\ y(k) &= Cx(k)+D_1u(k)+D_2w(k) \\ -E_1x(k) &- E_2u(k)-E_3w(k) \leq g_4 \\ -w(k) &\leq 0 \\ \sum_{i=1}^{p}\prod_{j\in\Phi_i}(g_4 &+ E_1x(k)+E_2u(k)+E_3w(k))_j(w(k))_j = 0 \end{aligned} \tag{3.37}$$

3.4.3 PWA and MLD systems

Proposition 3.4 *[5] Every well-posed PWA system can be rewritten as an MLD system assuming that the set of feasible states and inputs is bounded.*

As MLD model only allows non strict inequalities in 3.5 , by rewriting discontinuous PWA systems as an MLD model strict inequalities like $x(k) < 0$. This variable x(k) must be approximated by $x(k) \leq -\varepsilon$ for a positive number ε that implies -$\varepsilon < x(k) < 0$ cannot occur. By continuous PWA systems this inequality can be written non strictly or $\varepsilon = 0$.

Proposition 3.5 *[5] A completely well-posed MLD system can be rewritten as a PWA system.*

3.4.4 MMPS and ELC systems

Proposition 3.6 *[5] The classes of MMPS and ELC systems coincide.*

Proof

- Expressions of the form $f = x_i, f = \alpha, f = f_k + f_l$ and $f = \beta f_k$ results in linear equations of the form 3.10 and 3.11 .
- An expression of the form $f = \max(f_k; f_l) = -\min(-f_k; -f_l)$ can be written as

$$f - f_l \geq 0 \quad f - f_k \geq 0 \quad (f - f_k)(f - f_l) = 0 \tag{3.38}$$

  which is an expression of the form 3.12 and 3.13 .

It can be shown that two or more ELC systems can be combined into a large ELC system. So every MMPS can be rewritten as an ELC system.

The conditions 3.10 and 3.11 can be easily written as MMPS expression without max and min operations of the form 3.18 . The condition 3.12 show that

$$(g_4 - E_1 x(k) - E_2 u(k) - E_3 d(k))_j \geq 0 \quad \text{for each j} \tag{3.39}$$

The condition 3.13 show that $(g_4 - E_1x(k) - E_2u(k) - E_3d(k))_j \geq 0$ for $\forall \in \{1,2,...,p\} : \exists j \in \Phi_i$. This condition can be rewritten as

$$\min_{j\in\Phi}(g_4 - E_1x(k) - E_2u(k) - E_3d(k))_j \geq 0 \quad for \quad i = 1,2,...,p \tag{3.40}$$

The condition 3.12 can be rewritten for as

$$\min_{j\in\Psi}(g_4 - E_1x(k) - E_2u(k) - E_3d(k))_j \geq 0 \tag{3.41}$$

where $\Psi = \{j \in \{1,2,...,q\} \,|\forall i \in \{1,2,...,p\} : j \notin \Phi_i\}$. So conditions 3.13 and 3.12 can be rewritten as the last two previous conditions respectively.

3.4.5 MLD and ELC systems

Proposition 3.7 *[5] Every MLD system can be rewritten as an ELC system.*

Proof If we make an abstraction of the range of the variables then 3.3-3.5 coincide with 3.10-3.12 with d(k)=$[\delta^T(k)\ z^T(k)]^T$. The condition $\delta_i(k) \in$0,1 is equivalent to the ELC conditions

$$-\delta_i(k) \leq 0 \quad \delta_i(k) \geq 1 \quad \delta_i(k)(1 - \delta_i(k)) = 0 \tag{3.42}$$

So every MLD system can be rewritten as ELC system.

The condition $\delta_i(k) \in$0,1 is equivalent to the MMPS condition

$$\min(\delta_i(k); 1 - \delta_i(k)) = 0 \tag{3.43}$$

Proposition 3.8 *[5] 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 (component wise) bounded.*

Proof

$$(g_4)_j - (E_1x(k) - E_2u(k) - E_3d(k))_j \leq M_j\delta_j(k) \quad j \in \Phi_i$$
$$\sum_{j\in\Phi_i} \delta_j(k) \leq m_i - 1 \tag{3.44}$$

where $\delta i(k) \in$0,1 are auxiliary variables and M_j is the upper-bound for $(g_4)_j - (E_1x(k) - E_2u(k) - E_3d(k))_j$.For some j=h; $(g_4)_{j=h} - (E_1x(k) - E_2u(k) - E_3d(k))_{j=h} = 0$.

At least by defining z(k)=d(k) we can rewrite ELC system as an MLD system.

3.4.6 PWA and MMPS systems

Theorem 3.9 *If f is a continuous PWA function, then there exist sets* $I_1,...,I_l \subseteq \{1,...,N\}$ *such that*

$$f = \max_{j\in\{1,...,l\}} \min_{i\in I_i} (\alpha_i x + \beta_i) \tag{3.45}$$

It is used two strategies to rewrite the continuous PWA systems as MMPS system: Gorokhovik-Zorko strategy and Ovchinnikov strategy.

3.4.6.1 Gorokhovik-Zorko strategy

Definition 3.4 *Hypograph is a region above or below the graph and it's symbol is hyp(.).*

Proposition 3.10 $I_j \subseteq \{1,...,M\}$ *is an index set for the MMPS function y in 3.45 , and so* $I_j \in \{I_1,...,I_l\}$ *if and only if*

$$\min_{i\in I_j} f_i \leq f \tag{3.46}$$

or equivalently

$$hyp(\min_{i\in I_j} f_i) \subseteq hyp\ f \tag{3.47}$$

Let us consider a PWA function $f : X \to R$ with $X' \subset R_n$, where X' is a closed polyhedron. So there exists a polyhedral partition $\{X_{ij}\}_{i\in\{1,...,M\}, j\in\{1,...,m_i\}}$ of X' such that $f(x) = \alpha_i^T x + \beta_i$ on each X'_{ij} for every $i \in$1, . . .,M and j=1, . . .,m_i, where m_i is the number of polyhedral in which the affine term $\alpha_i^T x + \beta_i$ is defined and M is the number of affine terms. [11]

The hypograph of each min term can be computed as the intersection of hypographs of all its argument since every min term is a concave function.

Therefore the hypograph of the PWA function can be rewritten as the union of the polyhedra H_i, for i=1,...,M where we define H_i as follows [11]:

$$H_i = hyp(f_i) \cap ((\cup_{j=1,...,m_i} X'_{ij}) \times R) \tag{3.48}$$

and so

$$hyp\ f = \cup_{i=1,...,M} H_i \tag{3.49}$$

Definition 3.5 *The power set of a set R is the set of all its subsets, and it is denoted as $P(R)$.*

So we can rewrite a continuous PWA as MMPS system due to Gorokhovik-Zorko strategy [11] with the help of following algorithm [10]:

Algorithm 1

1. Let $I = \{1, ..., M\}$ and S a set defined as $S = P(I) - \emptyset$;
2. for each set $I_j \in S$ do;
3. if $hyp(\min_{i \in I_j} f_i) \nsubseteq hyp\ f$, then remove I_j from S;
4. endfor;

3.4.6.2 Ovchinnikov Strategy

Let $f : X' \to R$ be the continuous PWA function, so we can define f with $X' \subset R^n$, where X' is a closed polyhedron. The affine components of f can be denoted as f_i for $i = 1, ..., M$.

The hyperplanes that are nonempty solution sets of the equation in the form $f_i = f_j$ for $i < j$ and have nonempty intersections with the interior of X' form an hyperplane arrangement H. As the arrangement is n dimensional, these hyperplanes are (n-1) dimensional. Corollary, these hyperplanes generate a polyhedral partition in X' and the set of these polyhedral partition is denoted as T.

Definition 3.6 *[12]A facet is a (n-1) dimensional face of a polyhedron in R^n.*

Definition 3.7 *[12]Two polyhedral regions are adjacent if they have a facet in common.*

At first we will choose all the pairs of affine component f_p, f_q of f on P such that the following conditions are satisfied:

1. There is a pair of adjacent regions $P, Q \in T$ such that $f_p = f$ on P and $f_q = f$ on Q.

2. $f = \max(f_p, f_q)$ on $P \cup Q$

Then the hyperplanes that are nonempty solution sets of the equations in the form $f_p = f_q$ and for each pair f_p, f_q that satisfies the previous condition build an arrangement. This hyperplane arrangement is denoted as H' where $H' \subseteq H$. Corollary, The set of regions obtained through the subdivision of X' by the hyperplanes in H' is denoted as T'.

If the regions of T' are denoted as $T'_1, ..., T'_t$ and for each $j = 1, ..., t$ the index set S_j is denoted as $S - j = \left\{ i \in \{1, ..., M\} : f_i(x) \geq f(x), \forall x \in T'_j \right\}$, we can represent the function f by the following equivalent MMPS function y:

$$y = \max_{j=1,...,t} \min_{i \in S_j} f_i \tag{3.50}$$

.

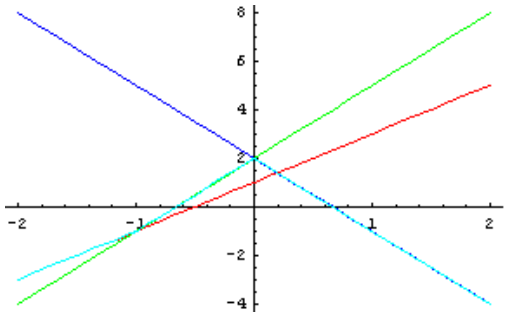

Figure 3.4: An Ovchinnikov strategy example

The PWA system in Fig.3.4 is given as:

$$f(x) = \begin{cases} 2x+1 & for\ x < -1 \\ 3x+2 & for\ -1 \leq x < 0 \\ -3x+2 & for\ 0 \leq x \end{cases} \tag{3.51}$$

We can divide the PWA system in 3 parts and we can rewrite the PWA system as:

$$f(x) = \begin{cases} \min(2x+1; -3x+2) & for\ x < -1 \\ \min(3x+2; -3x+2) & for\ -1 \leq x < 0 \\ \min(3x+2; -3x+2) & for\ 0 \leq x \end{cases} \quad \textbf{(3.52)}$$

At the end of Ovchinnikov strategy PWA system can be rewritten as an MMPS system:

$$f(x) = \max(\min(2x+1; -3x+2); \min(3x+2; -3x+2)) \quad \textbf{(3.53)}$$

We can rewrite PWA systems as MMPS systems due to Ochinnikov strategy [12] with the help of following algorithms [10]:

Algorithm 2

1. Let X' be the domain of the function f;
2. for each pair of adjacent polyhedral regions $X_{ik}, X_{jl} \in X'$, with $i < j$, do;
3. if $f_i \geq f_j$ on X_{ik} and $f_i \leq f_j$ on X_{jl}, insert the hyperplane that splits the two regions in H';
4. endfor;
5. Let T' the set of regions given by the intersection of X' with the regions of the hyperplane arrangement H';return T';stop;

Algorithm 3

1. Let T' be the region set returned by Algorithm 2;
2. for each region $T'_j \in T'$, with $j = 1, ..., M$, do;
3. if $f_i \geq f$ on T'_j, then insert the index i in the index set S_j;
4. endfor;
5. return the index sets S_j for $j = 1, ..., t$;

Now we can rewrite the continuous PWA function as in 3.50 .

4. MODEL PREDICTIVE CONTROL (MPC) AND ITS APPLICATION TO HYBRID SYSTEMS

4.1 Model Predictive Control

Mostly used PID controllers use error,derivative of error and integral of error as control parameters. Although PID controllers can maintain suitable robust optimized control solutions for linear dynamical systems,it can not promise suitable robust solutions for hybrid systems. One of the most important reasons of this situation is that PID controllers only use present and previously errors. Therefore PID controllers can not be easily adapted to systems whose reference signal are high-frequency signals. [13]

To developed a more suitable control solution the reference signal must be predicted for a period. Therefore *Model-Predictive Controller* are developed. The main idea of MPC is to predict oncoming reference signal for a finite period with the help of previous reference signals. MPC can be easily adapted to limitations on the control signal. On the other hand,one of the most important handicaps of MPC is that we must wait to build database of previous reference signals in order to predict oncoming reference signals. Another important handicap of MPC is that the prediction horizon must be updated for each step. Therefore we can only use the first control signal for each step although the control sequence of prediction horizon has been computed. [13]

It is used two time intervals by MPC so called *prediction horizon* and *control horizon*. Prediction horizon is the predicted reference signal interval and symbolized as N_p. As shorter prediction horizons can cause inaccurate predictions,longer prediction horizons cause long computation times and inaccurate predictions. The decision of the prediction horizon is an important matter. To enable smooth response and control signals it is used constant control

signals or constant control signal derivatives for interval $[N_c, N_p]$. The time interval [0,N_c] is called as control horizon.

Bordons & Camacho(1995) show that our MPC problem can be written as

$$\tilde{y}(k) = H\tilde{u}(k) + g(k) \tag{4.1}$$

where

$$\tilde{y}(k) = \begin{bmatrix} \hat{y}(k+1|k) \\ \vdots \\ \hat{y}(k+N_p|k) \end{bmatrix}, \tilde{r}(k) = \begin{bmatrix} r(k+1) \\ \vdots \\ r(k+N_p) \end{bmatrix}, \tilde{u}(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} \tag{4.2}$$

$$H = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & \cdots & CB \end{bmatrix}, g(k) = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix} \tag{4.3}$$

A performance index or cost function J,which penalize reference tracking errors and control inputs size, is used in MPC to optimize the controlled system output.

$$J = J_{out} + \lambda J_m = \sum_{j=1}^{N_p} \|\hat{y}(k+j|k) - r(k+j)\|^2 + \lambda \sum_{j=1}^{N_p} \|u(k+j-1)\|^2 \tag{4.4}$$

Additional linear constraints are described as

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) \tag{4.5}$$

We must remember control horizon rule too

$$u(k+j) = u(k+N_c-1) \quad for\ j = N_c, N_{c+1}, ... \tag{4.6}$$

The parameters N_p,N_c and λ are the three basic MPC tuning parameters. The prediction horizon N_p is related to the length of the step response of the process where the time interval (1,N_p) should contain the dynamics of the system. The control horizon N_c is taken equal to the system order where $N_p \geq N_c$. The parameter $\lambda \geq 0$ makes a connection between tracking errors and control effort. The parameter λ is chosen as small as possible because the controller will be more stable as this parameter is decreasing.

4.1.1 MPC for MPL systems

It can be shown the similarity of plus-times systems and max-plus-linear systems. [4] [6] So we can write

$$\hat{y}(k+j|k) = C \otimes A^{\otimes j} \otimes x(k) \oplus \oplus_{i=0}^{j-1} C \otimes A^{\otimes j-i} \otimes B \otimes u(k+i) \tag{4.7}$$

or

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

$$H = \begin{bmatrix} C \otimes B & \varepsilon & \cdots & \varepsilon \\ C \otimes A \otimes B & C \otimes B & \cdots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ C \otimes A^{\otimes N_p - 1} \otimes B & C \otimes A^{\otimes N_p - 2} \otimes B & \cdots & C \otimes B \end{bmatrix} \tag{4.8}$$

$$g(k) = \begin{bmatrix} C \otimes A \\ C \otimes A^{\otimes 2} \\ \vdots \\ C \otimes A^{\otimes N_p} \end{bmatrix}$$

The control horizon rule of MPC for MPL can be written as

$$\delta u(k+j) = \delta u(k+N_c) \quad for\ j = N_c, N_{c+1}, ..., N_{p-1} \tag{4.9}$$

or

$$\delta^2 u(k+j) = 0 \quad for \quad j = N_c, N_{c+1}, ..., N_{p-1}$$

So our standard MPC-MPL problems obtained as

$$\min_{\tilde{u}(k)} J = \min_{\tilde{u}(k)} J_{out,p_1} + \lambda J_{in,p_2} \tag{4.10}$$

subject to

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k) \tag{4.11}$$

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) \tag{4.12}$$

$$\delta u(k+j) = \delta u(k+N_c) \quad for \quad j = N_c, N_{c+1}, ..., N_{p-1} \tag{4.13}$$

$$\delta^2 u(k+j) = 0 \quad for \quad j = N_c, N_{c+1}, ..., N_{p-1} \tag{4.14}$$

De Schutter& Van den Boom(2001) suggest relaxed MPC method to solve this problem. It can be shown that objective function J and $\tilde{y}$ are monotonically nondecreasing functions. So we can rewrite the condition as

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) = -h(k) \tag{4.15}$$

Theorem 4.1 *Let the objective function J and mapping $\tilde{y} \rightarrow F(k)\hat{y}$ be monotonically nondecreasing functions of $\tilde{y}$. Let $(u^*,\tilde{y}^*)$ be an optimal solution of the relaxed MPC problem. If we define $\tilde{y}^{\#} = H \otimes \tilde{u}^* \oplus g(k)$ then $(u^*,\tilde{y}^{\#})$ is an optimal solution of the original MPC problem.*

4.1.2 MPC for switching MPL Systems

The procedure to find best MPC for MPL systems can be adapted to MPC for switching MPL systems. The additional constraint v(k) and timing problem of events reasoned some modifications on the procedure.

Due to correspondence of the input u(k) on the event times x(k) it can be written

$$\Delta u(k+j) = u(k+j) - u(k+j-1) \geq 0 \quad for\ j = \{0,1,...,N_p\} \tag{4.16}$$

It is wanted that the change rate of the input in the interval $[N_c - 1, N_p]$ will be constant.

$$\Delta u(k+m) = \Delta u(k+N_c-1) \quad for\ m = \{N_c,...,N_p\} \tag{4.17}$$

Additional constraint v(k) will not change in the interval $[N_c - 1, N_p]$.

$$\Delta v(k+j) = 0 \quad for\ j = \{N_c,...,N_p-1\} \tag{4.18}$$

Additional criteria on u(k) and output y(k) can be written as

$$A_c(k)\tilde{u}(k) + B_c\tilde{y}(k) \leq c_c(k) \tag{4.19}$$

So our problem can be described as

$$\min_{\{\tilde{u}(k)\in U, \tilde{v}(k)\in U(k)\}} J(k) \tag{4.20}$$

where

$$\begin{array}{rl} x(k) = A^{(l(k))} \otimes x(k-1) \oplus B^{(l(k))} \otimes u(k) & \\ \Phi(k+j-1), l(k+j-1), u(k+j), v(k+j) \in Z^{(l(k+j))} & for\ j = 0,...,N_p-1 \\ \Delta u(k+j) = u(k+j) - u(k+j-1) \geq 0 & for\ j = \{0,1,...,N_p\} \\ \Delta u(k+m) = \Delta u(k+N_c-1) & for\ m = \{N_c,...,N_p\} \\ \Delta v(k+j) = 0 & for\ j = \{N_c,...,N_p-1\} \\ A_c(k)\tilde{u}(k) + B_c\tilde{y}(k) \leq c_c(k) & \end{array} \tag{4.21}$$

If the constraint v(k) is a binary value,the problem will be an integer optimization problem where global minimum search algorithms like genetic algorithms or

tabu search can be used. Some cases can be solved via mixed integer linear programming (MILP). If the constraint v(k) has a real value, the problem can be solved via ELCP. [4]

Another problem in this optimization is that event times are not strict because of constraint v(k). An event time estimator mechanism is proposed to annihilate this problem. [4]

4.2 Application of MPC for Hybrid system's subclasses

In this section we will discuss two applications of MPC for hybrid system's subclasses: Max-Min-Plus-Scaling systems and Mixed Logical Dynamical systems.

4.2.1 MPC for MMPS systems

Theorem 4.2 *A scalar-valued MMPS function f can be written into the min-max canonical form*

$$f = min_{i=1,\ldots,K} max_{j=1,\ldots,n_i} (\alpha_{(i,j)}^T x + \beta_{(i,j)}) \tag{4.22}$$

or into the max-min canonical form

$$f = \max_{i=1,\ldots,L} \max_{j=1,\ldots,m_i} (\gamma_{(i,j)}^T x + \delta_{(i,j)}) \tag{4.23}$$

for some integers K,L,n_i,m_i; vectors α_i,γ_i; real numbers β_i,δ_i. For vector-valued MMPS functions the above statements hold component wise. [10] [14]

In MMPS-MPC we compute each step k an optimal control input that minimize the cost function over the period [k,k+N_p-1] where N_p is the prediction horizon. We assume that for each step k current state can be measured or estimated. We can estimate y(k +j|k) of the output after step k+j based on the state x(k-1) and future inputs u(k+i) . We obtain y(k+j|k)=F_j(x(k-1),u(k),u(k+1),...,u(k+j)) for as a MMPS function.

The cost function $J(k) = J_{out}(k) + \lambda J_{in}(k)$ used in MMPS-MPC where λ is a nonnegative weight parameter.

$$\begin{aligned} \tilde{u}(k) &= \begin{bmatrix} u^T(k) & \ldots & u^T(k+N_p-1) \end{bmatrix} \\ \tilde{r}(k) &= \begin{bmatrix} r^T(k) & \ldots & r^T(k+N_p-1) \end{bmatrix} \\ \tilde{y}(k|k) &= \begin{bmatrix} y^T(k|k) & \ldots & y^T(k+N_p-1|k) \end{bmatrix} \end{aligned} \tag{4.24}$$

In the practice, there are constraints on input and output signals. These constraints are modeled in MMPS as $C_c(k, x(k-1), \tilde{u}(k), \tilde{y}(k)) \geq 0$.

In MMPS-MPC is used the control horizon N_c which means that the input signal is constant after sample step k+N_c.

$$u(k+j) = u(k+N_c-1) \quad \forall j \in [N_c, N_p-1] \tag{4.25}$$

A more smaller control horizon N_c implies more smoother signal. On the other hand, the control horizon must be so wide that the controller has enough degrees of freedom to reach the constraints.

Some of the optimization algorithms to solve MMPS-MPC problem can be multi-start nonlinear optimization based on sequential programming (SQP) and method based on the extended linear complementarity problem (ELCP). SQP uses large number of initial start points and perform several optimization runs to find optimal solution. In addition, the objective functions in the MMPS-MPC problem are non-differentiable and PWA makes SQP less suitable for this problem. On the other hand, ELCP needs exponentially growing compute time for large numbers of input and state signals.

De Schutter & Van den Boom's algorithm for MMPS-MPC problem: [9] [14] [2]

Due to theorem in this subsection the objective function can be written in min-max canonical form as

$$J(k) = \min_{i=1,\ldots,L} \max_{j=1,\ldots,n_i} (\alpha_{(i,j)}^T u(k) + \beta_{(i,j)}(k)) \tag{4.26}$$

for appropriately defined integers L,n_i,vectors $\alpha_{(i,j)}$(k) and integers $\beta_{(i,j)}$.Note that the transformation into canonical form has performed and redundant terms are removed.

The derivation below is similar to the cutting-plane algorithm for convex optimization. The control horizon constraint is linear in $\tilde{u}(k)$. The original MPC constraint $C_c(k, x(k-1), \tilde{u}(k), \tilde{y}(k)) \geq 0$ will be not linear in $\tilde{u}(k)$ after substitution of $\tilde{y}$. Therefore we assume that there are only linear constraints on the input:

$$P(k)\tilde{u}(k) + q(k) \geq 0 \tag{4.27}$$

In practice such constraints occur if we have to guarantee that the control signal or control signal rate will stay within certain bounds. The optimization algorithm used below can also deal with convex constraints.

Optimization problem to obtain the optimal MPC input signal at sample step k is

$$\min_{u(\tilde{k})} \min_{i=1,\ldots,L} \max_{j=1,\ldots,n_i} \quad (\alpha_{(i,j)}^T u(k) + \beta_{(i,j)}(k)) \quad \text{subject to} P(k)\tilde{u}(k) + q(k) \geq 0 \tag{4.28}$$

or

$$\min_{i=1,\ldots,L} \min_{u(\tilde{k})} \max_{j=1,\ldots,n_i} \quad (\alpha_{(i,j)}^T u(k) + \beta_{(i,j)}(k)) \quad \text{subject to} P(k)\tilde{u}(k) + q(k) \geq 0 \tag{4.29}$$

The subproblem $\min_{\tilde{u}(k)} \max_{j=1,\ldots,n_i} (\alpha_{(i,j)}^T u(k) + \beta_{(i,j)}(k))$ subject to $P(k)\tilde{u}(k) + q(k) \geq 0$ is equivalent to the following LP problem:

$$\min_{t(k),\tilde{u}(k)} t(k) \text{ subject to} \begin{cases} t(k) \geq \alpha_{(i,j)}^T u(k) + \beta_{(i,j)}(k) & \forall j \in [1, n_i] \\ P(k)\tilde{u}(k) + q(k) \geq 0 \end{cases} \tag{4.30}$$

This LP can be solved efficiently using a simplex algorithm or an interior-point algorithm.After the LP problem for all $i \in [1, L]$ is solved,we will select $\tilde{u}_i^{opt}$ where $\alpha_{(i,j)}^T \tilde{u}_i^{opt}(k) + \beta_{(i,j)}(k)$ is minimum.

4.2.2 MPC for MLD systems

Definition 4.1 *[7] A vector $x_e \in R^n \times \{0,1\}^{n_l}$ is said to be an equilibrium state for MLD system and the input $u_e \in R^{m_l} \times \{0,1\}^{m_l}$ if $[x_e', u_e']' \in C$ and $x(t, t_0, x_e, u_e) = x_e, \forall t \geq t_0, \forall t_0 \in Z$. The pair (x_e, u_e) is called as an equilibrium pair.*

Definition 4.2 *[7] Let (x_e, u_e) be an equilibrium pair for a MLD system and let the system be well posed. Assume that $g = \lim_{t\to\infty} g_t$ exists. For $i \in g$ and $j \in g$, let $\delta_{e,i}, z_{e,j}$ the corresponding equilibrium auxiliary variables. An auxiliary vector*

δ *(or z) is said to be definitely admissible if* $\delta_i = \delta_{e,i}, \forall i \in g, (z_j = z_{e,j}, \forall j \in g)$ *and* $\exists t_e$ *such that*

$$E_{2t}\delta + E_{3t}z \leq E_{1t}u_e + E_{4t}x_e + E_{5t}, \forall t \geq t_e \tag{4.31}$$

Model predictive control of MLD systems depends on equilibrium pair (x_e, u_e) and their corresponding equilibrium auxiliary variables (δ_e, z_e). If the components $\delta_{e,i}, z_{e,j}, i \notin g, j \notin g$ correspond to desired steady-state values for the indefinite auxiliary variables, then our problem [7] can be written as

$$\begin{aligned} \min_{v_0^{T-1}} J(v_0^{T-1}, x(t)) \quad \equiv \quad & \sum_{k=0}^{T_1} \|v(k) - u_e\|_{Q_1}^2 + \|\delta(k|t) - \delta_e\|_{Q_2}^2 + \\ & + \|z(k|t) - z_e\|_{Q_3}^2 + \|x(k|t) - x_e\|_{Q_4}^2 + \\ & + \|y(k|t) - y_e\|_{Q_5}^2 \end{aligned} \tag{4.32}$$

subject to

$$\begin{gathered} x(T|t) = x_e \\ x(k+1|t) = Ax(k|t) + B_1 v(k) + B_2 \delta(k|t) + B_3 z(k|t) \\ y(k|t) = Cx(k|t) + D_1 v(k) + D_2 \delta(k|t) + D_3 z(k|t) \\ E_2 \delta(k|t) + E_3 z(k|t) \leq E_1 v(k) + E_4 x(k|t) + E_5 \end{gathered} \tag{4.33}$$

where $Q_1 = Q_1' > 0, Q_2 = Q_2' \geq 0, Q_3 = Q_3' \geq 0, Q_4 = Q_4' \geq 0, Q_5 = Q_5' \geq 0$, $x(k|t) \equiv x(t+k, x(t), v_0^{T-1})$ and $\delta(k|t), z(k|t), y(k|t)$ are similarly defined. Only the first element of input vector v(k) is applied to the system.

$$u(t) = v_t^*(0) \tag{4.34}$$

The defined control law in 4.33 is called as *mixed integer predictive control(MIPC) law*. MIQP solvers are used to find reliable solution.

5. PREVIOUS MATLAB IMPLEMENTATIONS

In this chapter we will analyze the previous works on implementation of model predictive control of hybrid systems on Matlab. One of the most comprehensive implementation works on Matlab is "Multi-Parametric Toolbox(MPT)" [15] which specializes on analysis and control of PWA and MLD systems. On the other hand, Andre Frau's master thesis analyze the equivalence of PWA and MMPS systems and minimization of MMPS functions [10]. An another work on hybrid systems, G.J. Benschop's master thesis, analyze minimization of MMPS function and MPC for hybrid system's subclasses [14].

5.1 Multi-Parametric Toolbox

MPT toolbox enables not only to model PWA systems with the help of Hybrid Identification Toolbox, MLD systems with the help of Hybrid System Description Language and to model nonlinear systems but also to analyze and control these systems. Toolbox designs model predictive controller for these systems. Toolbox contain various solvers like mpMILP and mpLP. The most important reason of the popularity of MPT toolbox is that the controller can be implemented by real systems with the help of "Real Time Workshop" of Matlab. Another important reason of this popularity is that these toolbox is developing still.

To model PWA and MLD systems MPT toolbox is using "HYSDEL" which has two main parts. [16] The first one, called INTERFACE, contains the declaration of all variables and parameters, so that it is possible to make the proper type checks. The second part, IMPLEMENTATION, is composed of specialized sections where the relations among the variables are described.

AUX SECTION: The HYSDEL section AUX contains the declaration of the auxiliary variables used in the model. These variables will become the ä and z variables in the MLD model.

AD SECTION: The HYSDEL section AD allows one to define Boolean variables from continuous ones, and is based exactly on the same semantics of the event generator (EG) described earlier. HYSDEL does not provide explicit access to the time instance, however this limitation can be easily overcome by adding a continuous state variable t such that $t' = t + T_s$, where T_s is the sampling time.

LOGIC SECTION: The section LOGIC allows one to specify arbitrary functions of Boolean variables: In particular the mode selector is a Boolean function and therefore it can be modeled in this section.

DA SECTION: The HYSDEL section DA defines continuous variables according to "if then else" conditions on Boolean variables. This section models part of the switched affine

CONTINUOUS SECTION: The CONTINUOUS section describes the linear dynamics, expressed as difference equations.

LINEAR SECTION: HYSDEL allows also one to define a continuous variable as an affine function of continuous variables in the LINEAR section. This section, together with the CONTINUOUS and AD sections allows more flexibility when modeling the SAS. This extra flexibility allows algebraic loops that may render undefined the trajectories of the model. The HYSDEL compiler integrates a semantic checker that is able to detect and report such abnormal situations.

AUTOMATA SECTION: The AUTOMATA section specifies the state transition equations of the finite state machine (FSM) as a collection of Boolean functions.

OUTPUT SECTION: The OUTPUT section allows one to specify static linear and logic relations for the output vector y = $[y_r\ y_b\ ]$. Finally HYSDEL allows one more section:

MUST SECTION: This section specifies arbitrary linear and logic constraints on continuous and Boolean variables, and therefore it allows for defining the sets Xr, Xb, Ur, Ub, Yr, Yb (more generally, the MUST section allows also mixed constraints on states, inputs, and outputs).

For example the system

$$x(k+1) = \begin{cases} 2*x(k) & \text{if} \quad x(k) \geq 0 \\ x(k)+u(k)-1 & \text{if} \quad x(k) < 0 \& x(k)+u(k)-1 < 0 \\ 2 & \text{if} \quad x(k) < 0 \& x(k)+u(k)-1 \geq 0 \end{cases} \tag{5.1}$$

can be written in HYSDEL as:

```
SYSTEM sample {
INTERFACE {
STATE {
REAL xr [-10, 10]; }
INPUT {
REAL ur [-2, 2]; }
}
IMPLEMENTATION {
AUX {
REAL z1, z2, z3;
BOOL de, df, d1, d2, d3; }
AD {
de = xr ≥ 0;
df = xr + ur - 1 ≥ 0; }
LOGIC {
d1 = ~de & ~df;
d2 = de;
d3 = ~de & df; }
DA {
z1 = {IF d1 THEN xr + ur - 1 };
z2 = {IF d2 THEN 2 * xr };
z3 = {IF d3 THEN 2 }; }
CONTINUOUS {
xr = z1 + z2 + z3; }
} }
```

5.2 Continuous PWA and MMPS systems

Andre Frau's master thesis contains two main theme, to convert continuous PWA systems into MMPS systems and vice versa and minimization of MMPS functions. Two strategies is used to convert continuous PWA systems into MMPS systems. First strategy to convert continuous PWA systems is the Gorohkovik-Zorko strategy. This strategy has given good results but the code depends on the number of affine components of PWA systems. On the other hand, the Ovchinnikov strategy is faster than Gorohkovik-Zorko strategy while it's efficiency is not so good [10]. According to Frau, these strategies can be developed to increase their efficiency. Minimization of MMPS functions is an important theme because it will decrease the computation time of analyze and control programs.

5.3 Minimization of MMPS expression

Definition 5.1 *[10] The function f is in its minimal realization if there does not exist a function*

$$f_1(x) = \min_{j\in\{1,\ldots,l'\}} \max_{i\in I_j}(\alpha_i^T x + b_i) \tag{5.2}$$

with l'<l, such that $f(x) \equiv f_1(x)$*, and next, if we cannot remove any entry in some of the index sets* I_j *without modifying the meaning of the function.*

We can reduce the number of max functions. The expression $\max_{i\in I_{l_1}}$ can be removed if

$$\max_{i\in I_{l_1}}(\alpha_i^T x + b_i) \geq \max_{i\in I_{l_2}}(\alpha_i^T x + b_i) \tag{5.3}$$

5.4 Comparison of MPC for hybrid systems

According to Benschop's master thesis and De Schutter(2004) MPC for MMPS problem can be solved most efficiently by nonlinear constraint optimization(SQP). [14] [9] It is stated in Benschop's master thesis that other MPC for hybrid system's subclasses are not so efficient. Although there are very effective methods to compute the global optima of MPC for MLD problem, the computational time depends on the size of the problem(N) and prediction horizon(T). The problem has a complexity of $N2^T$. [7] On the other hand, the

solver can be interrupted at any intermediate step (t+1) to obtain a suboptimal solution$u*_t$ which satisfies

$$J(U^*_{t+1}, x(t+1)) \leq J(U_1, x(t+1)) \tag{5.4}$$

6. MATLAB TOOLBOX

Our motivation in this master thesis is to develop a MATLAB toolbox for hybrid system's subclasses especially for MMPS systems. We must solve the following problems to develop a toolbox:

- Conversion of hybrid system's subclasses
- Integration of previously built functions and algorithms to the new toolbox
- Some improvements on the computational time of controllers

6.1 Conversion of hybrid system's subclasses

We have developed at first functions *getELCstruct* and *getLCstruct* to test Extended Linear Complementarity and Linear Complementarity system classes,respectively.
After we developed test functions for ELC and LC systems, we developed functions to convert hybrid system's subclasses.

6.1.1 lc2elc

This function is developed to convert Linear Complementarity systems to Extended Linear Complementarity systems. We can convert the following Linear Complementarity system to an Extended Linear Complementarity system:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} w(k) \\ y(k) &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x(k) \\ v(k) &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x(k) \\ w(k) &= u(k) \\ 0 \leq v(k) &\perp w(k) \geq 0 \end{aligned} \tag{6.1}$$

In MATLAB, we have followng result:

$$b.A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \quad b.B1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad b.B2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
$$b.C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad b.D1 = 0 \quad b.D2 = 0$$
$$b.E1 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad b.E2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad b.E3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$
$$b.G4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad b.Phi = \{\begin{bmatrix} 1 & 2 \end{bmatrix}\} \tag{6.2}$$

6.1.2 lc2mld

This function is developed to convert Linear Complementarity systems to Mixed Logical Dynamical systems. We can convert the Linear Complementarity system in the previous example to Mixed Logical Dynamical system. As result we have

$$c.A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \quad c.B1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad c.B2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
$$c.C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad c.D1 = 0 \quad c.D2 = 0$$
$$c.E1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \quad c.E2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad c.E3 = \begin{bmatrix} -10 \\ 10 \\ 0 \\ 0 \end{bmatrix}$$
$$c.E4 = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} \quad c.G5 = \begin{bmatrix} 0 \\ 3 \\ 0 \\ 0 \end{bmatrix} \tag{6.3}$$

The bug of the function is that the function does not check if the argument is a Linear Complementarity system which has been converted from a Mixed Logical Dynamical system. If the argument is a Linear Complementarity system which has been converted from a Mixed Logical Dynamical system, the new Mixed Logical Dynamical system will be much complicated system than the original.

6.1.3 mld2elc

This function is developed to convert Mixed Logical Dynamical systems to Extended Linear Complementarity systems. The following system can be written

as Extended Linear Complementarity system:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \delta(k) + \begin{bmatrix} 0 \\ 0.2 \end{bmatrix} z(k) \\ y(k) &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x(k) \\ \begin{bmatrix} 1 & 0 \end{bmatrix} x(k) &+ u(k) - 2\delta(k) \leq 1 \end{aligned} \tag{6.4}$$

In MATLAB, we had the following results:

$$\begin{gathered} b.A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad b.B1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad b.B2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0.2 \end{bmatrix} \\ b.C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad b.D1 = 0 \qquad b.D2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ b.E1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad b.E2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad b.E3 = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & -2 \end{bmatrix} \\ b.G4 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \qquad b.Phi : [23] \end{gathered} \tag{6.5}$$

6.1.4 mld2lc

This function is developed to convert Mixed Logical Dynamical systems to Linear Complementarity systems. We can convert the Mixed Logical Dynamical system in the previous example to Linear Complementarity system.

$$\begin{gathered} b.A1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad b.B1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad b.B2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0.2 & -0.2 \end{bmatrix} \\ b.C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad b.D1 = 0 \qquad b.D2 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\ b.E1 = \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad b.E2 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} \quad b.E3 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ b.G4 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \end{gathered} \tag{6.6}$$

The bug of the function is that the function does not check if the argument is a Mixed Logical Dynamical system which has been converted from a Linear Complementarity system. If the argument is a Mixed Logical Dynamical system

which has been converted from a Linear Complementarity system, the new system will be much complicated system than the original.

6.2 Integration of Mr. Frau's and Mr. Benschop's functions

We have developed the function *frau2ben* to convert struct formated MMPS functions; the form which is used by Mr. Frau; to cell formated MMPS functions, the form which is used by Mr. Benschop. The function *ben2frau* is developed to convert cell formated MMPS functions to struct formated MMPS functions.

6.3 Functions for MMPS systems

We have developed the function *mmpscalc* to calculate the value of the MMPS function. As *mmpscalc* can be used only for a MMPS function for only entry of states and inputs, the function *sisresp* is developed to calculate the system response of a MMPS system to a series of inputs and initial values of the states. Example:

$$\begin{aligned} x_1(k+1) &= \max\left[\min\left(4x_1x(k)+x_2(k)+\max[x_2(k),x_1(k)]+3, 2x_2(k)\right), 2x_1(k)\right] \\ x_2(k+1) &= \max(x_1(k), x_2(k)+0.5u(k)+1) \\ y(k) &= \max(x_1(k), x_2(k)) \end{aligned} \tag{6.7}$$

In MATLAB:

$$\begin{aligned} b &= mmpscalc(a\{1\}, \{'x1','x2','u'\}, [2;3;7]) \\ b &= 6 \end{aligned} \tag{6.8}$$

$$\begin{aligned} [out, sta] &= sisresp(\{'x1','x2'\}, [2;3], \{'u'\}, [7,10,7,9], \{'y'\}, a) \\ out &= \begin{bmatrix} 3 & 7.5 & 15 & 30 \end{bmatrix} \\ sta &= \begin{bmatrix} 2 & 6 & 15 & 30 & 60 \\ 3 & 7.5 & 13.5 & 18 & 30 \end{bmatrix} \end{aligned} \tag{6.9}$$

Our aim is to develop a function to calculate model predictive controller for MMPS systems without limitations on outputs and states. We will use van den Boom & de Schutter's linear programing based model predictive controller algorithm. As we mention previous chapter we must calculate the cost function

$$
\begin{aligned}
cost\,func &= PMat(i,1) * |J(y-r)|_{i,1} + RMat(i,1) * |J(y-r)|_{i,inf} + \\
&\quad QMat(i,1) * |J(u)|_{i,1} + SMat(i,1) * |J(u)|_{i,inf} \qquad \textbf{(6.10)}
\end{aligned}
$$

To calculate the cost function we develop the function "find_cost". Example:

$$
\begin{aligned}
b\{1\} &= \{'min', \{'+','x', \{'*', 2, 'u'\}, 1\}, \{'+', \{'*', 1.5, 'x'\}, 'u'\}\} \\
b\{2\} &= 'x1' \\
c &= find\_cost([0],[1],[0],[0],\{'x'\},\{'y'\},\{'u'\},2,b) \\
c &= \max[\min(x+2u(0)+1, 1.5x+u(0)) - r_1(2), r_1(2) \\
&\quad - \min(x+2u(0)+1, 1.5x+u(0)), x-r_1(1), r_1(1)+x] \qquad \textbf{(6.11)}
\end{aligned}
$$

In order to apply linear programing based model predictive controller algorithm we must rewrite our cost function in the *conjunctive* form. For this purpose we are using by mr. Benschop developed function *can_form_mmps*. Sometimes the function *mplusm* which is called in *can_form_mmps* because the function can be to complicated that "length" function which is a original Matlab function can not handle the number of variables. After the MMPS function converted into conjunctive form, the MMPS (cost)function in conjunctive form can be converted easily to struct formated MMPS functions with the help of the function *ben2frau*. If we continue previous example:

$$
\begin{aligned}
d &= can\_form\_mmps(c,1,'conj') \\
disp\_mmps(d) & \\
\min \quad & \Big[\quad max\big(1+-r_1(2)+2u(0)+x,1+-r_1(2)+2u(0)+x\big), \\
& \max \quad \big(-r_1(1)+x,-r_1(1)+x\big), \\
& \max \quad \big(r_1(1)+-x,r_1(1)+-x\big), \\
& \max \quad \big(-1+r_1(2)+-2u(0)+-x,-1+r_1(2)+-2u(0)+-x\big), \\
& \max \quad \big(r_1(2)+-u(0)+-1.5x,r_1(2)+-u(0)+-1.5x\big), \\
& \max \quad \big(-r_1(2)+u(0)+1.5x,-r_1(2)+u(0)+1.5x\big), \\
& \max \quad \big(-r_1(1)+x,-r_1(1)+x\big),\max\big(r_1(1)+-x,r_1(1)+-x\big), \\
& \max \quad \big(-1+r_1(2)+-2u(0)+-x,-1+r_1(2)+-2u(0)+-x\big), \\
& \max \quad \big(r_1(2)+-u(0)+-1.5x,r_1(2)+-u(0)+-1.5x\big)\Big] \qquad \textbf{(6.12)} \\
e &= ben2frau(d)
\end{aligned}
$$

$$
e.alpha = \begin{bmatrix}
-1 & 2 & 1 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & -1.5 & 0 \\
0 & 0 & -1.5 & 0 \\
0 & 0 & 1.5 & 0 \\
0 & 0 & 1.5 & 0
\end{bmatrix}
\quad
e.beta = \begin{bmatrix}
1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}
$$

$$
\begin{aligned}
e.degisken &= \{'r\_1(2)','u(0)','x','r\_1(1)'\} \\
e.terms &= \{[1,2][3,4][5,6][7,8][9,10][11,12]\} \qquad \textbf{(6.13)}
\end{aligned}
$$

The function *duz_costfunc* enables us to separate inputs from other variables (references and initial values of states). So our system can be written as

$$\begin{aligned} f &= duz\_costfunc(e,\{'x'\},\{'r\_1(1)','r\_1(0)'\},\{'u'\},2) \\ f.alpha &: [12x1double] \\ f.beta &: [12x1double] \\ f.betanew &: [12x3double] \\ f.degisken &: \{'u(0)','u(1)'\} \\ f.terms &: \{[1,2],[3,4],[5,6],[7,8],[9,10],[11,12]\} \end{aligned} \quad \textbf{(6.14)}$$

So our mpc-mmps problem can be written as

$$\min_{t(k),\tilde{u}(k)} t(k) \text{ subject to } \begin{cases} t(k) \geq f.alpha_{(i,j)}^{T} u(k) + f.beta_{(i,j)}(k) \\ \qquad\qquad + f.betanew * \begin{bmatrix} stateinitials \\ references \end{bmatrix} \\ \qquad\qquad \forall j \in [1, n_i] \\ A\_ex * u(k) \leq b\_ex \\ A\_eq * u(k) = b\_eq \end{cases} \quad \textbf{(6.15)}$$

where A_ex,b_ex,A_eq,b_eq are additional limitation matrices on inputs. This problem can be solved with the help of the function *mpc_mmps*.

7. CONCLUSION

In the second chapter we have analyzed discrete event system models. As widely used discrete event system modeling tools automata theory and petri nets approach use the state of entire system as main point, Max-Plus-Linear (MPL) systems are concentrating on event times. This structure of MPL systems enables to develop a modeling tool for hybrid systems that have both discrete event systems and time systems characteristics at the same time.

In the third chapter we have introduced hybrid system's subclasses and their relationships. It can be shown that hybrid system's subclasses are equivalent under some assumptions [5]. In the fourth chapter we have given a short introduction to model predictive control. After the main idea and mechanism of model predictive control have been explained, implementations of MPC on MPL and some hybrid system's subclasses are given.

In the fifth chapter we have shown some comprehensive MATLAB applications related hybrid systems. the most important and comprehensive work is absolutely "Multi-Parametric Toolbox (MPT)". MPT has features as to design MPC for PWA and MLD systems in addition to some more complex features. The other important MATLAB implementations that we are analyzed are A.Frau's master thesis and G.J. Benschop's master thesis. As A.Frau's thesis is concentrating on conversion between continuous PWA and MMPS systems and minimization of MMPS systems, G.J. Benschop's thesis is concentrating on MPC for hybrid system's subclasses and their comparison.

In the last main chapter we have explained the functions which we have developed to achieve our goals. These three goals were

- Conversion of hybrid system's subclasses
- Integration of previously built functions and algorithms to the new toolbox

- Some improvements on the computational time of controllers

We have developed functions to convert hybrid system's subclasses to each other. We can say that functions *ben2frau* and *frau2ben* help us to integrate A.Frau's and G.J. Benschop's works to each other and the whole toolbox. We have developed a model predictive controller algorithm for MMPS systems with limitations on inputs.

The most important future work is to develop a model predictive controller algorithm for MMPS systems with limitations on states and outputs. Another important future work can be a model predictive controller algorithm for switched MPL systems and for other hybrid system's subclasses. In the future it can be developed functions to convert ELC systems to MLD and MMPS systems. Functions to convert MMPS systems to ELC systems are required too. The bug which we explained in the subsection *mld2lc* and *lc2mld* can be corrected.

REFERENCES

[1] **Cassandras, C.G. and Lafortune, S.** Introduction to Discrete Event Systems.

[2] **Neocoara, I.**, 2006. Model Predictive Control for Piecewise Affine and Max-Plus-Linear Systems, Ph.D. thesis, Delft University of Technology.

[3] **Heidergott, B.**, **Olsder, G. and van der Woude, J.**, 2006. Max Plus at Work, princeton Series in APPLIED MATHEMATICS, Princeton.

[4] **van den Boom, T. and Schutter, B.D.**, 2006. Modeling and control of discrete event systems using switching max-plus-linear systems, *Control Engineering Practice*, 1199.

[5] **Heemels, W.**, **Schutter, B.D. and Bemporad, A.**, 2001. Equivalence of hybrid dynamical models, *Automatica*, 1085.

[6] **Schutter, B.D. and van den Boom, T.**, 2001. Model predictive control for max-plus-linear discrete event systems, *Automatica*, 1049.

[7] **Bemporad, A. and Morari, M.**, 1999. Control of systems integrating logic,dynamics, and constraints, *Automatica*, 407.

[8] **Heemels, W.P.M.H.**, 30 november 1999. Linear Complementarity Systems:A Study in Hybrid Dynamics, Ph.D. thesis, Technische Universiteit Eindhoven.

[9] **Schutter, B.D. and van den Boom, T.**, 2004. MPC for continuous piecewise-affine systems, *Systems & Control Letters*, 179.

[10] **Frau, A.**, 2007. PWA and MMPS functions: equivalences and transformations, Master's thesis, Delft University of Technology.

[11] **Gorokhovik, V. and Zorko, O.**, 1994. Piecewise affine functions and polyhedral sets, *Optimization*, 31:209–221.

[12] **Ovchinnikov, S.** Max-min representation of piecewise linear functions, *Beitr̀age zur Algebra und Geometrie/Contributions to Algebra and Geometry*, 43(1):297–302.

[13] **Mayne, D.**, **Rawlings, J.**, **Rao, C. and Scokaert, P.**, 2000. Constrained model predictive control: Stability and optimality, *Automatica*, 789.

[14] **Benschop, G.**, 2001. Model predictive control for a class of hybrid systems, Master's thesis, Delft University of Technology.

[15] **Kvasnica, M.**, **Grieder, P. and Baotić, M.**, 2004, Multi-Parametric Toolbox (MPT), *http://control.ee.ethz.ch/ mpt/*.

[16] **Bemporad, A.**, 2003, Modeling, Control, and Reachability Analysis of Discrete-Time Hybrid Systems.

CIRCULUM VITAE

Veysel Gürkan ANIK was born in 1982 in Istanbul. Right after he graduated Istabul (erkek) Lisesi in 2001, he begun his study at Control Engineering program. He graduated in 2005 and he applied to Control and Automation Engineering master(Msc) program right after graduation. His career began in December 2005 as research assistant at Istanbul Technical University.