Laboratório de Modelagem, Análise e Controle de Sistemas Não-Lineares
Departamento de Engenharia Eletrônica
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, MG Brasil
Fone: +55 31 3409-3470

Takagi-Sugeno Models in a Tensor Product Approach

Exploiting the representation

Víctor Costa da Silva Campos

Advisor: Prof. Dr. Leonardo Antônio Borges Tôrres
Co-Advisor: Prof. Dr. Reinaldo Martinez Palhares

Belo Horizonte, July 24, 2015

Laboratório de Modelagem, Análise e Controle de Sistemas Não-Lineares
Departamento de Engenharia Eletrônica
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, MG Brasil
Fone: +55 31 3409-3470

Takagi-Sugeno Models in a Tensor Product Approach

Exploiting the representation

Thesis presented to the Graduate Program in Electrical Engineering (PPGEE) of the Universidade Federal de Minas Gerais (UFMG) as a partial requirement to obtaining the degree of Doctor in Electrial Engineering

Víctor Costa da Silva Campos

Advisor: Prof. Dr. Leonardo Antônio Borges Tôrres
Co-Advisor: Prof. Dr. Reinaldo Martinez Palhares

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Belo Horizonte
July 24, 2015

"Takagi-sugeno Models In A Tensor Product Approach - Exploiting The Representation"

Víctor Costa da Silva Campos

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 10 de junho de 2015.

Por:

Prof. Dr. Leonardo Antônio Borges Tôrres
DELT (UFMG) - Orientador

Prof. Dr. Reinaldo Martinez Palhares
DELT (UFMG)

Prof. Dr. Fernando de Oliveira Souza
DELT (UFMG)

Prof. Dr. Ricardo Hiroshi Caldeira Takahashi
DMAT (UFMG)

Prof. Dr. Eduardo Stockler Tognetti
Dep. de Engenharia Elétrica (Universidade de Brasilia)

Prof. Dr. Eduardo Nunes Gonçalves
Dep. de Engenharia Elétrica (CEFET/MG)

Prof. Dr. João Pedro Hespanha
University of California Santa Barbara - UCSB (EUA)

Prof. Dr. Thierry-Marie Guerra
University of Valenciennes and Hainaut- Cambresis – UVHC (França)

Acknowledgements

These were four long years, and I thoroughly enjoyed all that I've learned and the life I've lead during them. And because of that I'm very grateful.

I'm grateful to god for this wonderful opportunity.

To Léo and Reinaldo for putting up with me through all of these 6 and a half years. I'm pretty sure I wasn't the model graduate student an advisor would wish for (given all the random stuff I do), but you sure made me feel like it! I couldn't have asked for better advisors.

To CNPq for the funding while I was here at UFMG, and CAPES for the funding while I was abroad (specially for the people responsible for my grant while I was abroad due to my *sandwich internship* not being like the usuals).

To Thierry and João for agreeing to advise me while I was in each of their labs. This experience was amazing, and I only bring good memories from both of them.

To my dear family, for all the love, support and tenderness even while I was away. All of you mean the world to me. I couldn't write this without specially thanking those that went out of their way to make me feel loved and at home, specially when I was abroad. So I'd like to take this moment and thank Jacqueline, Tio Júlio, Tia Ana Maria and Tio Fábio for meeting me in Europe and my parents Amália and José, my brothers Matheus and Marcos and my soon-to-be sister-in-law Júlia for meeting me in the US!

To my dear old time friends, who are always there, even when I'm not: Mairon, Rafael, Raul, Elvis, Daniel, Augusto, Carol, André.

To my dear (now also old time) friends from UFMG, for making me feel at home wherever we are. Specially, Tales, Cristina, Wendy, Dimas, Leandro, Luciano, Thales, Vitinho, Vinicius, Grazi, Fernanda, Tiago, Jullierme, Rogério, Johnathas, Thiago, Rodrigo, Marcus, Ana e Petrus.

Here's a shoutout to "beberrões" - I'll forever be indebted to you guys for making me company, even when I was far away.

To my dancing friends, Ana (too bad it didn't work out between us), Paulinho, Kleinny and everyone else for reminding me that there's more to life than just the university.

To all the friends I've made while I was in France, in special everyone from LAMIH - AnhTu, Victor, Raymundo, Miguel, Antonio, Rémi - it was a pleasure meeting you all.

To all the friends I've made while I was in the US, specially those from CCDC and the SBDC, Henrique, Becky, Ya, Michelle, Kunihisa, Masashi, Hari, Kyriakos, Noor. You guys made my life over there something memorable that I'll always cherish.

“A lesson without pain is meaningless. That’s because no one can gain without sacrificing something. But by enduring that pain and overcoming it...
...he shall obtain a powerful, unmatched heart.
A fullmetal heart.”

Edward Elric, *FullMetal Achemist Brotherhood*

Abstract

Takagi-Sugeno (TS) fuzzy analysis and control techniques extend several results from linear robust control theory to nonlinear systems. However, in many cases, the conditions used do not explore the membership functions' information aside from the fact that they belong to the standard unit simplex. In addition, since we are dealing with nonlinear systems, the use of quadratic Lyapunov functions carries a certain conservatism.

In this work, we propose different ways to exploit the membership functions' information in parameter dependent Linear Matrix Inequalities (LMIs) as well as new classes of candidate Lyapunov functions that cover other existing classes in the literature.

Asymptotically necessary conditions for parameter dependent LMIs are proposed based on partitioning the membership functions' image space. Such partitions are also used to propose a switched control law and we present ways to modify the conditions of this controller such that a continuous controller can be recovered.

An alternative way to reduce the number of rules of a given Takagi-Sugeno model, based on the Higher Order Singular Value Decomposition (HOSVD), is presented as well as how to model the uncertainty introduced by this rule reduction scheme.

We propose the use of local transformations of the membership functions in conjunction with piecewise fuzzy Lyapunov functions so as to find more relaxed stability conditions than others available in the literature.

A novel way to deal with the time derivative of the membership functions, in cases where they are functions of the states, is proposed. This new proposition avoids the use of direct bounds over the membership functions time-derivative, and new stability and stabilization conditions are presented.

Finally, we propose new LMI synthesis conditions based on piecewise-like Lyapunov functions. This class of Lyapunov functions behaves similarly to piecewise functions (in the sense that we can understand that a different function is valid for each region) while using a fuzzy formulation. By adapting recent conditions to deal with bounds on the Lyapunov function's membership functions time derivative, new synthesis conditions are presented to guarantee local stabilization.

Resumo

As técnicas de análise e controle de sistemas Takagi-Sugeno (TS) estendem vários resultados de controle robusto de sistemas lineares para sistemas não-lineares. Entretanto, em muitos dos casos, as condições utilizadas não exploram a informação das funções de pertinência do sistema, com exceção do fato de elas pertencerem ao simplex unitário padrão. Além disso, por se tratarem de sistemas não-lineares, o uso de funções de Lyapunov quadráticas carrega consigo um certo conservadorismo.

Neste trabalho são propostas algumas formas diferentes de se explorar a informação das funções de pertinência nas Desigualdades Matriciais Lineares (LMIs) dependentes de parâmetros bem como novas classes de funções de Lyapunov candidatas que cobrem outras classes existentes na literatura.

São propostas condições assintoticamente necessárias para LMIs dependentes de parâmetros baseadas em partições do espaço imagem das funções de pertinência. Tais partições são também utilizadas para se propor uma lei de controle chaveada de acordo com tais partições e como modificar as condições dessa lei de controle chaveada para que se possa recuperar um controlador contínuo.

Uma maneira alternativa, baseada na Decomposição de Valores Singulares de Alta Ordem (HOSVD) para a redução do número de regras de modelos TS é proposta, bem como formas de se modelar a incerteza introduzida pelo procedimento de redução de regras proposto.

É proposto o uso de transformações locais das funções de pertinência em conjunto com funções de Lyapunov fuzzy por partes de modo a se encontrar condições de estabilidade mais relaxadas do que outras apresentadas na literatura.

Uma nova forma de lidar com a derivada temporal das funções de pertinência é proposta para os casos em que as funções de pertinência são funções das variáveis de estado somente. Esta nova proposição evita o uso de limitantes sobre as derivadas temporais das funções de pertinência, e novas condições de estabilidade e estabilização são apresentadas.

Finalmente, são apresentadas novas condições LMI de síntese baseadas em funções de Lyapunov quase-por-partes. Esta classe de funções de Lyapunov se comporta de maneira similar às funções por partes (no sentido de que pode-se entender que uma função diferente é válida para cada região) e faz uso de uma formulação fuzzy. Adaptando-se condições recentes na literatura sobre limitantes superiores nas derivadas temporais das funções de pertinência da função de Lyapunov, novas condições de síntese são propostas para garantir a estabilização local.

Contents

List of Figures

List of Tables

Notation

a	Scalar
$\mathbf{a}$	Vector
A	Matrix
$\mathcal{A}$	Tensor
$\mathbf{a}_i$	i-th column vector of matrix A
a_{ij}	Element from row i, column j of matrix A
$a_{i_1 i_2 \dots i_N}$	Element from position $(i_1, i_2, \dots, i_N)$ of tensor $\mathcal{A}$
$\mathcal{A}_{(n)}$	n-mode matrix of tensor $\mathcal{A}$
$A \otimes B$	Kronecker product between matrices A and B
$\langle \cdot, \cdot \rangle$	Scalar (dot) product operator
$\times_n$	n-mode product operator between a tensor and a matrix
$\\| \cdot \\|$	Norm operator
$\mathcal{S} \overset{N}{\underset{n=1}{\times}} U^{(n)}$	Short notation for $\mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)}$
$\mathbf{1}$	Vector whose components are all equal to one
A^T	Transpose of matrix A
$*$	Transpose elements inside of a symmetric matrix
$P > 0$	Denotes that matrix P is positive definite
$P \geq 0$	Denotes that matrix P is positive semi-definite
$P < 0$	Denotes that matrix P is negative definite
$P \leq 0$	Denotes that matrix P is negative semi-definite
$P \succ 0$	Denotes that all elements of matrix P are positive
$P \succeq 0$	Denotes that all elements of matrix P are non-negative
$P \prec 0$	Denotes that all elements of matrix P are negative
$P \preceq 0$	Denotes that all elements of matrix P are non-positive

List of Acronyms

BMI Bilinear Matrix Inequality

CNO Close to Normalized

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

HOOI Higher Order Orthogonal Iteration

HOSVD Higher Order Singular Value Decomposition

INO Inverse Normalized

LMI Linear Matrix Inequality

LPV Linear Parameter Varying

NN Non Negative

NO Normalized

PDC Parallel Distributed Compensation

PDVA Grupo de Pesquisa e Desenvolvimento de Veículos Autônomos

qLPV quasi-Linear Parameter Varying

RNO Relaxed Normalized

SN Sum Normalized

TS Takagi-Sugeno

1 Introduction

Takagi-Sugeno (TS) fuzzy models (Takagi and Sugeno 1985) are capable of representing nonlinear dynamical systems as a convex combination of linear models (or in some cases affine models - Campos et al. 2013; Johansson, Rantzer, and Arzen 1999; Wang and Yang 2013). This representation allows that some of these systems' analysis and synthesis conditions be written as optimization problems subject to LMIs (Boyd et al. 1994; Teixeira and Assunção 2007).

*However, most of the conditions presented in the literature do not make much use of the membership functions information, aside from the fact that they belong to the standard unit simplex (*i.e.* they add up one and are all bigger than or equal to zero) (Sala 2009). Such conditions are, therefore, conservative since they are valid for a larger family of systems than the one we wish to analyze/control. Since we are dealing with nonlinear systems, another source of conservatism comes from the family of candidate Lyapunov functions used in the conditions.*

1.1 Motivation

Takagi-Sugeno models with a Tensor Product (TP) structure (Ariño 2007; Ariño and Sala 2007) (also known as *multisimplex* structure - Tognetti, Oliveira, and Peres 2011; Tognetti, Oliveira, and Peres 2012) are models whose membership functions can be written as the product of one-variable membership functions, and are really common in the literature (Bernal and Guerra 2010; Campos et al. 2013; Guerra et al. 2012; Mozelli, Palhares, and Avellar 2009; Rhee and Won 2006). This class of models covers those found using the well-known sector nonlinearity approach (Tanaka and Wang 2001), as well as those found using the Tensor Product model transformation (Baranyi 2004; Baranyi et al. 2003).

Examples of use of this class of TS models in the literature include:

- Less conservative sufficient conditions for double fuzzy summations (Ariño and Sala 2007);
- A fuzzy Lyapunov function whose stability analysis conditions do not depend on the time derivatives of the membership functions (Rhee and Won 2006);
- Replacing the time derivative of the membership function by partial derivatives regarding the states and considering local analysis/synthesis conditions instead of global ones (Bernal and Guerra 2010; Guerra et al. 2012).

Another advantage of this TP representation, little explored in the literature, is the fact that this representation allows the decomposition of the image space of the membership functions into the tensor product of the image spaces of the one-variable membership functions. These decomposed image spaces are simpler, and can, for example, be approximated by a finite number of samples from the functions and an approximation error (that can be calculated by means of the Mean Value Theorem). These approximations, in turn, allow the study of the membership functions' "shape" to be carried in a finite dimensional space.

When studying this "shape" information, it might be useful to split the image space into regions. In cases where the premise variables are all state variables, one can map these regions to regions in the system's state space. In these cases, an interesting approach is to employ a different Lyapunov function for each region, or a "fuzzy blending" of them.

A different problem for these systems' analysis and synthesis conditions, unrelated to their conservativeness, is that they become computationally intractable for models with a high number of rules. Due to the nature of the conditions, it is not always easy to develop conditions that scale well with the number of rules. A simple workaround to this problem is to employ an uncertain simpler model with a smaller number of rules.

1.2 Objectives

This work aims to present forms of incorporating the membership functions' "shape" information (additional information aside from the usual one that they belong to the standard unit simplex) into the LMI conditions, as well as alternative ways of reducing the number of rules of a TS model (and how to model the error introduced by this reduction as an uncertainty).

Specifically, we aim to:

- Present sufficient and asymptotically necessary conditions for LMIs with a fuzzy summation, taking into consideration the shape of the membership functions;
- Propose new control laws for TS systems whose synthesis conditions make use of the membership functions' shape;
- Propose novel families of candidate Lyapunov functions whose analysis/synthesis conditions make better use of the membership functions' shape;
- Propose alternative ways of reducing the number of rules of TS models in a way that the reduced model's behaviour is "close" to the original, and how to use the error between the models to generate an uncertain model;

1.3 Outline

This work aims to study different ways of exploiting the knowledge of TS systems' membership functions, whether for obtaining less conservative analysis/synthesis conditions, or reducing the number of rules of a given TS model. Some of the results presented here may be applied to TS models with any sort of membership functions, whereas others can only be applied in cases where the membership functions are only allowed to be functions of the state variables. To that end, the remainder of this thesis is structured as follows[1]:

Chapter 2 presents definitions and results used throughout all of the work.

[1]note that, even though the text was structured in this order, it does not reflect the chronological order. If the chapters were presented in chronological order, after chapter 2, the order would be: 5, 3, 4, 6, and 7

Part I groups the results that are independent of which variables are used as premise variables in the TS model. Chapter 3 presents ways of exploring the membership functions' shape. And chapter 4 presents an alternative way of reducing the number of rules of a given TS model based on techniques from the Tensor Product Model Transformation, as well as how to model the error introduced by this rule reduction as different forms of uncertainty.

Part II groups the results that assume that the TS model's premise variables are state variables (or functions of them). Chapter 5 presents two novel families of piecewise fuzzy Lyapunov functions for affine TS models, as well as their stability analysis conditions. Chapter 6 presents a novel way to deal with the membership functions' time derivative when making use of fuzzy Lyapunov functions (sharing the same membership functions as the TS system). Chapter 7 presents a family of fuzzy Lyapunov functions whose membership functions are different from the TS system, but that give conditions similar to those one would expect to get for piecewise Lyapunov functions.

Part III is composed only of chapter 8, that summarizes, discusses and presents conclusions regarding this thesis.

2 Theoretical Foundation

2.1 Takagi-Sugeno Fuzzy Models

Throughout this work, we consider, in a general way, two distinct types of TS fuzzy models (Takagi and Sugeno 1985): those whose premise variables might be any signal, and those whose premise variables are state variables.

Free premise variables

A representative model, in this work, for cases in which the premise variables might be any signal is:

$$\begin{aligned} \textbf{Rule } i\textbf{:} \quad & \text{If } v_1 \in \mathcal{M}_1^{\alpha_{i1}} \text{ and } \dots \text{ and } v_\ell \in \mathcal{M}_\ell^{\alpha_{i\ell}}, \\ & \text{then } \dot{\mathbf{x}} = A_i\mathbf{x} + B_i\mathbf{u} \end{aligned}$$

in which $\mathbf{x} \in \mathbb{R}^n$ is the state vector; $\mathbf{u} \in \mathbb{R}^m$ is the control input vector; $\mathbf{v} \in \mathbb{R}^\ell$ is the premise variables vector (v_i is the i-th element of vector $\mathbf{v}$); $A_i \in \mathbb{R}^{n\times n}$ and $B_i \in \mathbb{R}^{n\times m}$ are matrices describing the dynamical behaviour of the system; i represents the rule number; $\mathcal{M}_j^{\alpha_{ij}}$ represents the fuzzy sets related to the premise variable v_j; α_{ij} is an index that relates each fuzzy set of the premise variable v_j with rule i (*i.e.* an index $\alpha_{21} = 3$ indicates that the third fuzzy set of the premise variable v_1 is used on the second rule); and $1 \leq \alpha_{ij} \leq r_j$ with r_j the number of fuzzy sets related to the premise v_j.

Considering that each fuzzy set $\mathcal{M}_j^{\alpha_{ij}}$ has a corresponding membership function $\omega_j^{\alpha_{ij}}(v_j)$ with

$$0 \leq \omega_j^{\alpha_{ij}}(v_j) \leq 1.$$

A normalized membership function can be obtained from $\omega_j^{\alpha_{ij}}(x_j)$, by

$$\mu_j^{\alpha_{ij}}(v_j) = \frac{\omega_j^{\alpha_{ij}}(v_j)}{\sum_{\alpha_{ij}=1}^{r_j} \omega_j^{\alpha_{ij}}(v_j)}.$$

These normalized functions have the properties

$$\mu_j^{\alpha_{ij}}(v_j) \geq 0; \qquad \sum_{\alpha_{ij}=1}^{r_j} \mu_j^{\alpha_{ij}}(v_j) = 1. \tag{2.1}$$

Such properties endow the inferred TS membership functions with the properties

Property 2.1: Membership functions decomposition

$$h_i(\mathbf{v}) = \prod_{j=1}^{\ell} \mu_j^{\alpha_{ij}}(v_j), \tag{2.2}$$

Property 2.2: *convex sum property*

$$h_i(\mathbf{v}) \geq 0, \qquad \sum_{i=1}^{r} h_i(\mathbf{v}) = 1, \qquad \sum_{i=1}^{r} \dot{h}_i(\mathbf{v}) = 0. \tag{2.3}$$

where $r = \prod_{j=1}^{\ell} r_j$ is the TS model's total rule number.

The second property may also be seen as imposing that the membership function take values in the standard $(r-1)$-dimensional simplex whose vertices are given by the standard unit vectors of $\mathbb{R}^r$.

Making use of the membership functions (2.2), the inferred TS model can be written in a compact way as:

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{v}) \left(A_i \mathbf{x} + B_i \mathbf{u} \right). \tag{2.4}$$

Premises are state variables

The only difference in the case where the premises variables are restricted to being state variables is that the premise vector is substituted by the state vector (or a subset of the state vector). However, in chapter 5, we'll also deal with affine TS models. A representative model in this case is given by (Campos et al. 2013):

Rule i**:** If $x_1 \in \mathcal{M}_1^{\alpha_{i1}}$ and … and $x_n \in \mathcal{M}_n^{\alpha_{in}}$,
then $\dot{\mathbf{x}} = A_i \mathbf{x} + \mathbf{a}_i$

in which $\mathbf{x} \in \mathbb{R}^n$ is the state vector; $A_i \in \mathbb{R}^{n \times n}$ and $\mathbf{a}_i \in \mathbb{R}^{n \times 1}$ are matrices and column vectors describing the dynamical behaviour of the system (and $\mathbf{a}_i$ is zero for every rule that is active at the origin); i represents the rule number; $\mathcal{M}_j^{\alpha_{ij}}$ represents the fuzzy sets related to the premise variable x_j; α_{ij} is an index that relates each fuzzy set of the premise variable x_j with rule i; and $1 \leq \alpha_{ij} \leq r_j$ with r_j the number of fuzzy sets related to the premise x_j.

It is worth noting that relations (2.1), (2.2) and (2.3) are also valid for this model (just substitute v by x), and, in this case, the inferred model can be compactly written as:

$$\dot{\overline{\mathbf{x}}} = \sum_{i=1}^{r} h_i(\mathbf{x}) \overline{A}_i \overline{\mathbf{x}}, \tag{2.5}$$

in which

$$\overline{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \qquad \overline{A}_i = \begin{bmatrix} A_i & \mathbf{a}_i \\ 0 & 0 \end{bmatrix}.$$

From this point forward, whenever it will not cause confusion we will the drop the dependence of the membership functions $h_i(\mathbf{x})$ or $h_i(\mathbf{v})$ and simply present them as h_i.

2.2 Fuzzy Summations

Many of the analysis and synthesis conditions for TS systems may be expressed as matrix inequalities involving fuzzy sums.

Some of these conditions, specially those related to the analysis of TS systems, may be written as

$$\sum_{i=1}^{r} h_i Q_i > 0, \tag{2.6}$$

in which Q_i are matrices containing parameters describing the systems and the decision variables of the optimization problem. In some parts of this work, expression (2.6) will be referred as a "single fuzzy summation".

Other conditions, like many of the synthesis conditions for TS systems available in the literature, may be written as

$$\sum_{i=1}^{r}\sum_{j=1}^{r} h_i h_j Q_{ij} > 0, \tag{2.7}$$

in which Q_{ij} are matrices containing parameters describing the systems and the decision variables of the optimization problem. In some parts of this work, expression (2.7) will be referred as a "double fuzzy summation".

Useful Results

In order to guarantee that condition (2.7) is valid, several sufficient conditions have been proposed in the literature. Some examples of these conditions are presented below:

Lemma 2.1: (Tanaka and Wang 2001)

A sufficient condition for expression (2.7) *to be valid, with* $i, j \in \{1, \ldots, r\}$, *is*

$$Q_{ii} > 0,$$
$$Q_{ij} + Q_{ji} > 0, \quad j > i.$$

Proof. Note that

$$\sum_{i=1}^{r}\sum_{j=1}^{r} h_i h_j Q_{ij}$$
$$= \sum_{i=1}^{r} h_i^2 Q_{ii} + \sum_{i=1}^{r-1}\sum_{j=i+1}^{r} h_i h_j \left(Q_{ij} + Q_{ji}\right)$$

Therefore the conditions on the lemma are sufficient for expression (2.7). □

Lemma 2.2: Theorem 2.2 - 1 in (Tuan et al. 2001)

A sufficient condition for expression (2.7) *to be valid, with* $i, j \in \{1, \ldots, r\}$, *is*

$$Q_{ii} > 0,$$
$$\frac{2}{r-1} Q_{ii} + Q_{ij} + Q_{ji} > 0, \quad j \neq i.$$

Proof. Note that

$$\sum_{i=1}^{r}\sum_{j=1}^{r} h_i h_j Q_{ij}$$
$$= \sum_{i=1}^{r}\sum_{j=i+1}^{r-1} \left[\frac{1}{r-1} h_i^2 Q_{ii} + \frac{1}{r-1} h_j^2 Q_{jj} + h_i h_j \left(Q_{ij} + Q_{ji}\right)\right]$$

Therefore the conditions on the lemma are sufficient for expression (2.7). □

Lemma 2.3: Adapted from Theorem 2 in (Xiaodong and Qingling 2003)

A sufficient condition for expression (2.7) *to be valid, with* $i, j \in \{1, \ldots, r\}$, *is that there exist matrices* $\Theta_{ij} = \Theta_{ji}^T$ *such that*

$$\begin{aligned} Q_{ii} &> \Theta_{ii}, \\ Q_{ij} + Q_{ji} &> \Theta_{ij} + \Theta_{ji}, \quad j > i, \\ \Theta &= \begin{bmatrix} \Theta_{11} & \ldots & \Theta_{1r} \\ \vdots & \ddots & \vdots \\ \Theta_{r1} & \ldots & \Theta_{rr} \end{bmatrix} > 0. \end{aligned}$$

Proof. Note that the first two conditions of the lemma imply in

$$\begin{aligned} &\sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j Q_{ij} \\ &\geq \sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j \Theta_{ij} \\ &= \begin{bmatrix} h_1 I \\ \vdots \\ h_r I \end{bmatrix}^T \begin{bmatrix} \Theta_{11} & \ldots & \Theta_{1r} \\ \vdots & \ddots & \vdots \\ \Theta_{r1} & \ldots & \Theta_{rr} \end{bmatrix} \begin{bmatrix} h_1 I \\ \vdots \\ h_r I \end{bmatrix} \end{aligned}$$

and the last condition implies that this expression is positive definite, therefore the conditions on the lemma are sufficient for (2.7). □

Remark 2.1

Despite having presented three different sets of sufficient conditions, for simplicity, this work makes use of the conditions presented in Lemma 2.1. Note, however, that any of the other two conditions could be used without problems.

If we are working with multiple fuzzy summations instead of a double summation, the following Lemma (which is a generalization of Lemma 2.1) will be employed in this work

Lemma 2.4: (Sala and Ariño 2007)

A sufficient condition for

$$\sum_{i_1=1}^{r} \cdots \sum_{i_q=1}^{r} \left(\prod_{\ell=1}^{q} h_{i_\ell} \right) L_{i_1 \ldots i_q} > 0,$$

is that, for every combination of $(i_1, i_2, \ldots, i_q)$ *with* $i_\ell \in \{1, 2, \ldots, r\}$, *the sum of its permutations is positive definite.*

Example 2.1. Consider that we have a condition of the form

$$\sum_{i_1=1}^{2} \sum_{i_2=1}^{2} \sum_{i_3=1}^{2} h_{i_1} h_{i_2} h_{i_3} L_{i_1 i_2 i_3} > 0,$$

the possible combinations of (i_1, i_2, i_3) are $(1,1,1)$, $(1,2,1)$, $(1,2,2)$ and $(2,2,2)$ and the use of Lemma 2.4 says that sufficient conditions for the LMIs are

$$\begin{aligned} L_{111} &> 0, \\ L_{112} + L_{121} + L_{211} &> 0, \\ L_{122} + L_{212} + L_{221} &> 0, \\ L_{222} &> 0. \end{aligned}$$

These conditions can easily be seen to be sufficient because the initial condition can be rewritten as

$$h_1^3 L_{111} + h_1^2 h_2 \left(L_{112} + L_{121} + L_{211}\right) + h_1 h_2^2 \left(L_{122} + L_{212} + L_{221}\right) + h_2^3 L_{222} > 0.$$

In addition to these fuzzy sum relaxations, the following two lemmas (modified from Bernal, Guerra, and Kruszewski 2009) will be useful throughout this work.

Lemma 2.5: Adapted from Lemma 1 in (Bernal, Guerra, and Kruszewski 2009)

Consider the normalized membership functions h_i, with $i \in \{1 \dots r\}$, and $\tilde{h}_j$, with $j \in \{1 \dots r\}$, belonging to the standard simplex of dimension $r-1$. Consider also the vectors $\mathbf{h}$, formed by grouping the h_i membership functions, and $\tilde{\mathbf{h}}$ formed by grouping the $\tilde{h}_j$ membership functions. Given a matrix $G \in \mathbb{R}^{r \times r}$ describing a transformation from $\tilde{\mathbf{h}}$ to $\mathbf{h}$, $\mathbf{h} = G\tilde{\mathbf{h}}$:

$$\sum_{i=1}^{r} h_i Q_i > 0 \Leftrightarrow \sum_{j=1}^{r} \tilde{h}_j \left(\sum_{i=1}^{r} g_{ij} Q_i \right) > 0,$$

with g_{ij} the element of row i and column j of matrix G.

Lemma 2.6: Adapted from Lemma 2 in (Bernal, Guerra, and Kruszewski 2009)

Consider the normalized membership functions h_i, with $i \in \{1 \dots r\}$, and $\tilde{h}_j$, with $j \in \{1 \dots r\}$, belonging to the standard simplex of dimension $r-1$. Consider also the vectors $\mathbf{h}$, formed by grouping the h_i membership functions, and $\tilde{\mathbf{h}}$ formed by grouping the $\tilde{h}_j$ membership functions. Given a matrix $G \in \mathbb{R}^{r \times r}$ describing a transformation from $\tilde{\mathbf{h}}$ to $\mathbf{h}$, $\mathbf{h} = G\tilde{\mathbf{h}}$:

$$\begin{aligned} &\sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j Q_{ij} > 0 \Leftrightarrow \\ &\sum_{k=1}^{r} \sum_{l=1}^{r} \tilde{h}_k \tilde{h}_l \left(\sum_{i=1}^{r} \sum_{j=1}^{r} g_{ik} g_{jl} Q_{ij} \right) > 0, \end{aligned}$$

with g_{ij} the element of row i and column j of matrix G.

2.3 Function Approximation

Under certain circumstances, the use of function approximations might simplify, or even allow, certain developments that would otherwise be too complex with the original function. Throughout this work, zero order interpolations (that generate piecewise constant functions) have a fundamental part and enable developments that would otherwise be too complex.

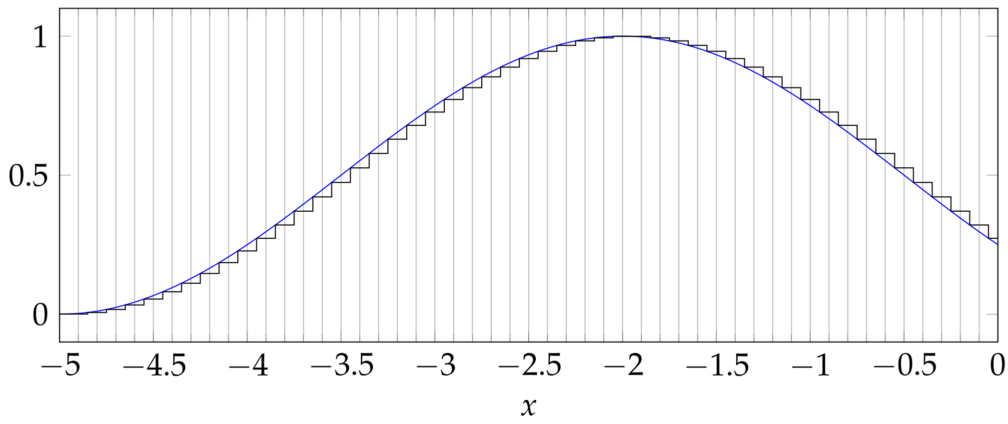

Figure 2.1: Example of a piecewise constant approximation

In that regard, this section aims to present, in a quick fashion, a way to consider these approximations and how to calculate an upper bound for the error between the original and the approximated function.

Zero order interpolation - piecewise constant functions

Consider a vector function of one variable $\mathbf{f}(x)$ in the interval $x \in [\underline{x}, \overline{x}]$ and n samples equally spaced, which is equivalent to a sampling step of

$$h = \frac{\overline{x} - \underline{x}}{n - 1}$$

A possible approximation for function $\mathbf{f}(x)$ is given by

$$\tilde{\mathbf{f}}(x) = \begin{cases} \mathbf{f}(\underline{x}), & \underline{x} \leq x < \underline{x} + \frac{h}{2} \\ \mathbf{f}(\underline{x} + ih), & \underline{x} + \frac{(2i-1)h}{2} \leq x < \underline{x} + \frac{(2i+1)h}{2} \\ \mathbf{f}(\overline{x}), & \overline{x} - \frac{h}{2} \leq x \leq \overline{x} \end{cases}$$

with
$i \in \mathbb{N}$ such that $1 \leq i < n$

Figure 2.1 presents an example of this approximation.

Error polytope

Assuming that $\mathbf{f}(x)$ is continuously differentiable, for each function $f_k(x)$ composing vector $\mathbf{f}(x)$, we may employ the Mean Value Theorem (*Mean Value Theorem* 2012)

$$\begin{aligned}
f_k(x) - f_k(x-\delta) &= \delta f_k'(y), & y \in [x-\delta, x] \\
f_k(x+\delta) - f_k(x) &= \delta f_k'(y), & y \in [x, x+\delta] \\
&\Downarrow \\
|f_k(x) - f_k(y)| &\leq \frac{h}{2} \max_z |f_k'(z)|, & y \in \left[x - \frac{h}{2}, x + \frac{h}{2}\right],\ z \in \left[x - \frac{h}{2}, x + \frac{h}{2}\right] \\
&\Downarrow \\
|f_k(x) - \tilde{f}_k(x)| &\leq \frac{h}{2} \max_y |f_k'(y)|, & y \in \left[x - \frac{h}{2}, x + \frac{h}{2}\right]
\end{aligned}$$

Using this relationship for each element of function $\mathbf{f}(x)$ separately, we can find an upper bound for the error in each component ζ_k and use it to build an error polytope around each sample.

This polytope will have 2^κ components, in which κ is the number of components of vector $\mathbf{f}(x)$, and its vertices will be given by the combinations of $\tilde{f}_k(x) + \zeta_k$ and $\tilde{f}_k(x) - \zeta_k$ for each component.

Error maximum 2-norm

In case of the error 2-norm, we may make use of the generalization of the Mean Value Theorem for vector functions (*Mean Value Theorem* 2012)

$$||\mathbf{f}(x) - \tilde{\mathbf{f}}(x)||_2 \leq \frac{h}{2} \max_y ||\frac{\partial}{\partial x}\mathbf{f}(y)||_2, \qquad y \in \left[x - \frac{h}{2}, x + \frac{h}{2}\right]$$

in which $\frac{\partial}{\partial x}\mathbf{f}(z)$ is the Jacobian vector of function $\mathbf{f}(x)$ evaluated at $\mathbf{z}$.

Several variables function - tensor product

In order to calculate the error bounds, we assume that the functions to be approximated have the following form

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}_1(x_1) \otimes \mathbf{f}_2(x_2) \otimes \cdots \otimes \mathbf{f}_n(x_n).$$

Therefore, the approximation will have the form

$$\tilde{\mathbf{f}}(\mathbf{x}) = \tilde{\mathbf{f}}_1(x_1) \otimes \tilde{\mathbf{f}}_2(x_2) \otimes \cdots \otimes \tilde{\mathbf{f}}_n(x_n).$$

Error calculation in several variables - tensor product

As shown in the following, for the polytopic error representation, it suffices to take the tensor product (Kronecker product) of the matrices representing the polytope.

$\mu_i \in \text{Polytope}_i$

$\Rightarrow \mu_i = \sum_{j=1}^{n_i} \alpha_j^i \mathbf{p}_j^i, \quad \alpha_j^i \geq 0 \quad \sum_{j=1}^{n_i} \alpha_j^i = 1$

$\mathbf{p}_j^i$ are vertices of Polytope_i

$P^i = \begin{bmatrix} \mathbf{p}_1^i & \mathbf{p}_2^i & \dots & \mathbf{p}_{n_i}^i \end{bmatrix}$, matrix representing Polytope_i

If $h = \mu_1 \otimes \mu_2 \otimes \cdots \otimes \mu_m$

$$\Rightarrow h = \left(\sum_{j_1=1}^{n_1} \alpha_{j_1}^1 \mathbf{p}_{j_1}^1\right) \left(\sum_{j_2=1}^{n_2} \alpha_{j_2}^2 \mathbf{p}_{j_2}^2\right) \cdots \left(\sum_{j_m=1}^{n_m} \alpha_{j_m}^m \mathbf{p}_{j_m}^m\right)$$

$$h = \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \cdots \sum_{j_m=1}^{n_m} \left(\prod_{i=1}^{m} \alpha_{j_i}^i\right) \left(\prod_{i=1}^{m} \mathbf{p}_{j_i}^i\right)$$

$h = \sum_{j=1}^{n} \overline{\alpha}_j \overline{\mathbf{p}}_j$, by reordering the indices with $n = \prod_{i=1}^{m} n_i$

$\overline{\alpha}_j$is given by the product of $\alpha_{j_i}^i$ with the indices according to the reordering

$\overline{\mathbf{p}}_j$is given by the product of $\mathbf{p}_{j_i}^i$ with the indices according to the reordering

$\Rightarrow h \in \text{Polytope}$

$P = \begin{bmatrix} \overline{\mathbf{p}}_1^i & \overline{\mathbf{p}}_2^i & \dots & \overline{\mathbf{p}}_{n_i}^i \end{bmatrix}$, matrix that represents the Polytope

$P = P^1 \otimes P^2 \otimes \cdots \otimes P^m$

Unfortunately, a simple way to propagate the bound on the 2-norm of the error was not found. Two different alternatives in this case are: using the Mean Value Theorem for several variables and recalculate the bound (it can be rather troublesome in some cases), or finding the smallest ball that contains the error polytope (since the center is known, it suffices to find the farthest vertex from the center and use that distance as radius. This approach is simple but at the same time might be conservative).

2.4 Tensor Product Model Transformation

The tensor product model transformation (Baranyi 2004; Baranyi et al. 2003) is a numerical technique that, given a quasi-Linear Parameter Varying (qLPV) system representation, finds an equivalent convex representation. It has interesting applications in numerically obtaining a TS model of a system, as well as reducing the number of rules of a given model.

In order to ease the presentation, initially we present some multilinear algebra concepts and the singular value decomposition generalization Higher Order Singular Value Decomposition (HOSVD). Afterwards, the tensor product model transformation is presented. Finally, we present an example applying the technique to a dynamical system.

The following presentation was based on the master's dissertation (Campos 2011), which in turn was based on (Baranyi 2004; De Lathauwer, De Moor, and Vandewalle 2000a).

Preliminary Concepts

The key point of the tensor product model transformation is using the HOSVD to find, among all the sampled values, those that really carry information (think of something similar to finding a base of linearly independent vectors for a subspace). However, in order to enunciate the theorem that defines this decomposition, some definitions are necessary.

The first of them, called *n-mode matrix* or *unfolding matrix* of a tensor, is a matrix representation of high order tensors that allow some tensor operations to be represented as matrix operations.

Definition 2.1: n-mode matrix of a tensor

Consider an N-th order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Its *n-mode matrix*, $\mathcal{A}_{(n)} \in \mathbb{R}^{I_n \times J}$, with $J = \prod_{\substack{k=1 \\ k \neq n}}^{N} I_k$, is a possible matrix representation of the tensor. This matrix contain the tensor's element $a_{i_1 i_2 \ldots i_N}$ in row i_n and column

$$\begin{aligned} &(i_{n+1} - 1) I_{n+2} I_{n+3} \ldots I_N I_1 I_2 \ldots I_{n-1} + (i_{n+2} - 1) I_{n+3} I_{n+4} \ldots I_N I_1 I_2 \ldots I_{n-1} + \ldots \\ &+ (i_N - 1) I_1 I_2 \ldots I_{n-1} + (i_1 - 1) I_2 I_3 \ldots I_{n-1} + (i_2 - 1) I_3 I_4 \ldots I_{n-1} + \cdots + i_{n-1}. \end{aligned}$$

In other words, this definition shows that the *n-mode* unfolding will be such that:

- The n-th index of each element indicates in which row it will be on the *n-mode matrix*.
- The columns are "unfolded" on the following order: first the $(n-1)$-th element, followed by the $(n-2)$-th element and so on in a circular fashion.

Example 2.2. Given a tensor $\mathcal{A} \in \mathbb{R}^{2 \times 2 \times 2}$, its unfolding is done as described above.

In the case of the *1-mode matrix*, the first index indicates the elements row and the columns are "unfolded" first by the third index and next by the second.

In the case of the *2-mode matrix*, the second index indicates the elements row and the columns are "unfolded" first by the first index and next by the third.

In the case of the *3-mode matrix*, the third index indicates the elements row and the columns are "unfolded" first by the second index and next by the first.

Its unfolding matrices are given by:

$$\begin{aligned} \mathcal{A}_{(1)} &= \left[\begin{array}{cc|cc} a_{111} & a_{112} & a_{121} & a_{122} \\ a_{211} & a_{212} & a_{221} & a_{222} \end{array}\right], \\ \mathcal{A}_{(2)} &= \left[\begin{array}{cc|cc} a_{111} & a_{211} & a_{112} & a_{212} \\ a_{121} & a_{221} & a_{122} & a_{222} \end{array}\right], \\ \mathcal{A}_{(3)} &= \left[\begin{array}{cc|cc} a_{111} & a_{121} & a_{211} & a_{221} \\ a_{112} & a_{122} & a_{212} & a_{222} \end{array}\right]. \end{aligned}$$

By defining the scalar product between two tensors, we can use it to define a norm and orthogonality between two tensors.

Definition 2.2: Scalar product of two tensors

The scalar product $\langle \mathcal{A}, \mathcal{B} \rangle$ of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} b_{i_1 i_2 \ldots i_N} a_{i_1 i_2 \ldots i_N}.$$

Definition 2.3: Tensors orthogonality

Two tensors are said orthogonal if their scalar product is 0.

Definition 2.4: Frobenius norm of a tensor

The Frobenius norm of a tensor $\mathcal{A}$ is given by

$$\|\mathcal{A}\| = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}.$$

Finally, we need to define the n-mode product between a tensor and a matrix. This product can be seen as a generalization of left and right matrix multiplications (equivalent to 1-mode and 2-mode products in this notation).

Definition 2.5: n-mode product

The n-mode product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by a matrix $U \in \mathbb{R}^{J_n \times I_n}$, represented as $\mathcal{A} \times_n U$, is a tensor belonging to $\mathbb{R}^{I_1 \times I_2 \times \cdots I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ whose elements are given by

$$(\mathcal{A} \times_n U)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} u_{j_n i_n}.$$

This operation may be described in terms of the n-mode matrices of the tensor $\mathcal{A}$ and of the resulting tensor as

$$(\mathcal{A} \times_n U)_{(n)} = U \mathcal{A}_{(n)}.$$

Property 2.3

Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and matrices $F \in \mathbb{R}^{J_n \times I_n}$ and $G \in \mathbb{R}^{J_m \times I_m}$, with $n \neq m$, then

$$(\mathcal{A} \times_n F) \times_m G = (\mathcal{A} \times_m G) \times_n F = \mathcal{A} \times_n F \times_m G.$$

Property 2.4

Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and matrices $F \in \mathbb{R}^{J_n \times I_n}$ and $G \in \mathbb{R}^{K_n \times J_n}$, then

$$(\mathcal{A} \times_n F) \times_n G = \mathcal{A} \times_n (GF).$$

From these definitions, we can enunciate the HOSVD, which will be essential for the tensor product model transformation.

Theorem 2.1: HOSVD - (De Lathauwer, De Moor, and Vandewalle 2000a)

Every tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *may be written as the product*

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)},$$

represented in a short manner as

$$\mathcal{A} = \mathcal{S} \mathop{\times}_{n=1}^{N} U^{(n)},$$

in which

1. $U^{(n)} = \begin{bmatrix} \mathbf{u}_1^{(n)} & \mathbf{u}_2^{(n)} & \dots & \mathbf{u}_{I_n}^{(n)} \end{bmatrix}$ *is an unitary matrix of dimensions* $I_n \times I_n$ *called n-mode singular matrix;*
2. $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$*, called core tensor, is a tensor whose subtensors* $\mathcal{S}_{i_n=\alpha}$*, obtained by fixing the n-th index in* α*, has the following properties:*
   a) *all-orthogonality: two subtensors* $\mathcal{S}_{i_n=\alpha}$ *and* $\mathcal{S}_{i_n=\beta}$ *are orthogonal if for all possible values of* n*,* α *and* β *subject to* $\alpha \neq \beta$*:*

   $$\langle \mathcal{S}_{i_n=\alpha}, \mathcal{S}_{i_n=\beta} \rangle = 0 \quad \forall \quad \alpha \neq \beta;$$

   b) *ordering:*

   $$\|\mathcal{S}_{i_n=1}\| \geq \|\mathcal{S}_{i_n=2}\| \geq \cdots \geq \|\mathcal{S}_{i_n=I_n}\| \geq 0$$

   for all possible values of n*.*

The Frobenius norm $\|\mathcal{S}_{i_n=i}\|$*, symbolized as* $\sigma_i^{(n)}$*, are the n-mode singular values of* $\mathcal{A}$ *and vector* $\mathbf{u}_i^{(n)}$ *is the corresponding n-mode singular vector.*

The calculus of this decomposition is done in two steps [1]. Initially, the n-mode singular matrices are calculated followed by the core tensor.

The n-mode singular matrices are the matrices of left singular vectors of the unfolding matrices. Therefore, each matrix (and the respective n-mode singular values) is found by a singular value decomposition of each of the tensor's unfolding matrices.

Having calculated the singular matrices, the core tensor is found by

$$\mathcal{S} = \mathcal{A} \times_1 U^{(1)^T} \times_2 U^{(2)^T} \cdots \times_N U^{(N)^T}.$$

Tensor Product Model Transformation - Steps

The tensor product model transformation is a numerical technique that finds convex approximations for functions, valid inside of a compact region of the function's domain. These convex approximations can be represented by a polytope of values (not necessarily belonging to the function's image space) whose convex combination is capable of representing the function's behaviour. This representation is equivalent to a TS fuzzy model of the function.

When applied to a qLPV model of a dynamical system, this technique is thus capable of finding a TS fuzzy representation of the system. In this case, we have a polytope of linear systems, because by fixing the parameter (state) values of the qLPV model, it becomes a linear model. This representation, in turn, allows the use of LMIs for controller synthesis, or analysis of the system.

[1] For more details, see (De Lathauwer, De Moor, and Vandewalle 2000a)

In the following, the tensor product model transformation steps and its use in a general setting are presented. Afterwards, an example of its specific use in obtaining a TS model of a system is presented.

Sampling and Tensor Representation

The first step when applying the tensor product model transformation is, given a function to be approximated, sample it inside of an hyper-rectangular region (that corresponds to the validity domain of the approximation) and store these samples in a tensor.

Given a function $f : [\underline{x_1}, \overline{x_1}] \times \cdots \times [\underline{x_N}, \overline{x_N}] \to \mathbb{R}^{M_1 \times \cdots \times M_m}$, in which $[\underline{x_1}, \overline{x_1}] \times \cdots \times [\underline{x_N}, \overline{x_N}]$ is an hyper-rectangular subset of $\mathbb{R}^N$, define a sample grid of size $I_1 \times \cdots \times I_N$ over the function's domain.

Define a tensor, $\mathcal{S}_d \in \mathbb{R}^{I_1 \times \cdots \times I_N \times M_1 \times \cdots \times M_m}$, that stores the sampled values of f for each grid point.

Assumption 2.1

The usual basic assumption when employing the tensor product model transformation is that the function we are approximating can be exactly represented as an interpolation (be it piecewise constant, piecewise linear or some different interpolation scheme [a]) of the samples. This amounts to saying that the approximation error between the original function and the interpolated function is not taken into account in this procedure.

[a] A requirement to guarantee that we get a convex representation at the end of the transformation is that the interpolation scheme only generates values given by the convex combination of the sample points. Some schemes with interesting properties are presented in (Campos, Tôrres, and Palhares 2015).

Therefore, the interpolation can be written as

$$f(\mathbf{x}) \approx \mathcal{S}_d \underset{n=1}{\overset{N}{\times}} \boldsymbol{\lambda}^{(n)}(x_n). \tag{2.8}$$

in which the functions $\boldsymbol{\lambda}^{(n)}(x_n)$ are weight functions that, when multiplied by the samples, generate the desired interpolation (*e.g.* piecewise Lagrange polynomials (de Boor 2001, Theorem 3)).

Example 2.3. Consider the function $f(x, y) = xy(1 + \sin(x)\cos(y))$, with $x \in [-10, 10]$ and $y \in [-10, 10]$, presentend in Figure 2.2. Consider that $f : [-10, 10] \times [-10, 10] \to \mathbb{R}$.

By defining a sampling grid step of 0.1 in x and in y, we get a sampling grid of size 201×201. Evaluating the function in each grid point and storing it in tensor $\mathcal{S}_d$, we actually get a matrix of size 201×201, because the last direction is a singleton (a direction of dimension 1).

Higher Order Singular Value Decomposition

Having defined tensor $\mathcal{S}_d \in \mathbb{R}^{I_1 \times \cdots \times I_N \times M_1 \times \cdots \times M_m}$ storing the samples of the function to be approximated, we make use of the HOSVD presented in the previous section to decompose the tensor.

The idea here is presenting the sample tensor as weighted sum, where the weights change according to the function's variables. In that regard, it is not interesting to decompose the tensor in all $N + m$ different dimensions. In this case, it is decomposed only in the first N dimensions, that represent the variables of the function being approximated.

We can then write it as:

$$\mathcal{S}_d = \mathcal{S} \underset{n=1}{\overset{N}{\times}} U^{(n)}.$$

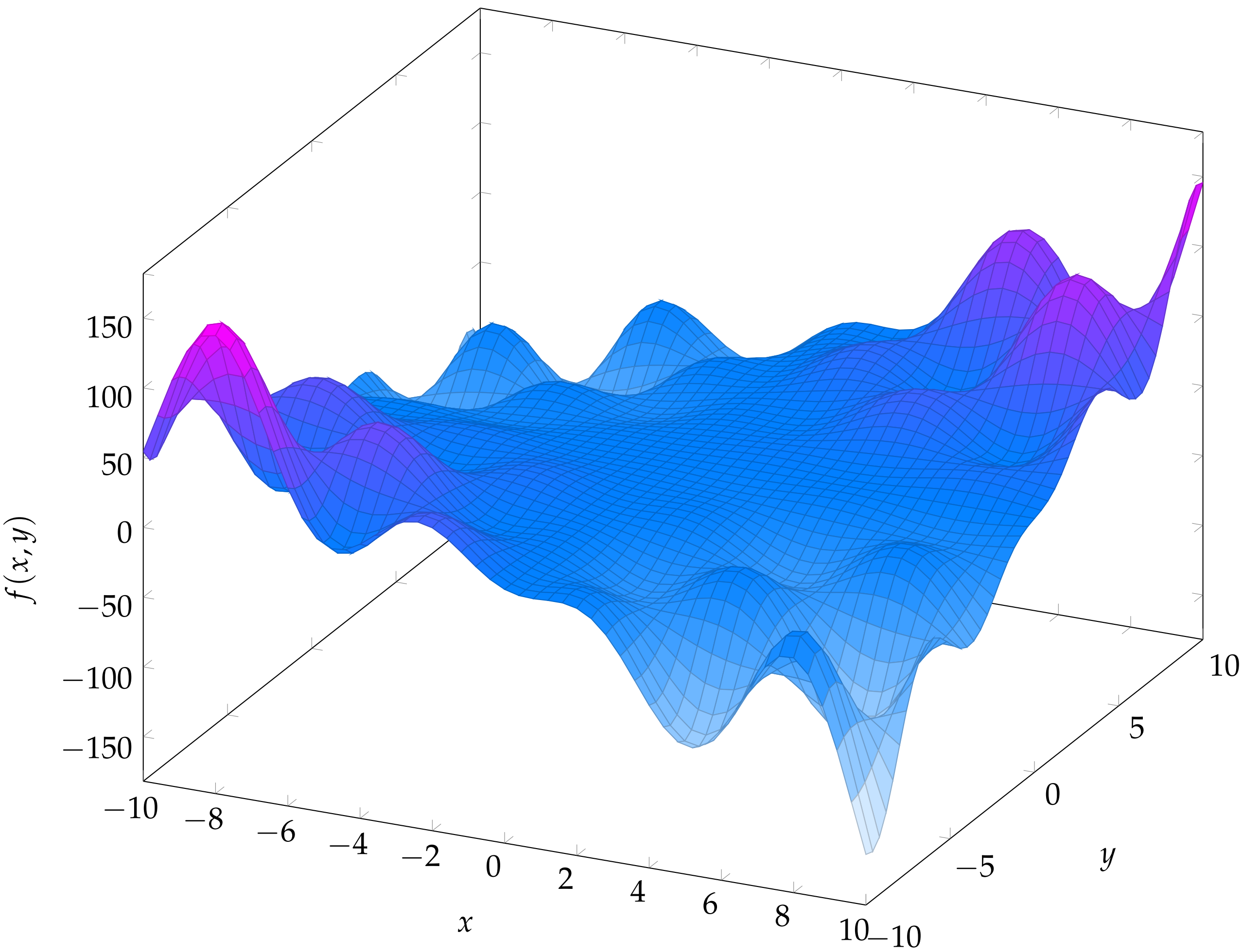

Figure 2.2: $f(x,y) = xy(1+\sin(x)\cos(y))$, used in example 2.3.

In this notation, we may understand the sample tensor as a weighted sum, in which each matrix $U^{(n)}$ represents the weights relative to variable n in the sample grid. Each column vector $\mathbf{u}_i^{(n)}$ represents a different weighting function for that variable.

As previously presented, the HOSVD calculation happens in two steps. In the first step, a singular value decomposition of each of the unfolding matrices is done to find the $U^{(n)}$ matrices and the n-mode singular values. In this stage, each column $\mathbf{u}_i^{(n)}$ found corresponds to an n-mode singular value.

Keeping only the columns whose singular values are different than zero, we find an exact representation for the tensor. Discarding columns whose singular values are not different than zero, we get a representation with a smaller complexity, but with a certain error. This reduced representation does not have the property of being the best reduced representation (in terms of mean squared error), as opposed to the case of reducing a matrix by means of the singular value decomposition. However, it is possible to show that the approximation error for discarding a column whose singular value is nonzero is bounded (and an upper bound for this approximation error can be calculated using the discarded singular values) (De Lathauwer, De Moor, and Vandewalle 2000a).

Thus, this is the tensor product model transformation step that allows a choice between the final complexity of the representation (number of weight functions for each variable, that is equivalent to the number of columns of the $U^{(n)}$ matrices) and the approximation precision (Baranyi 2004).

Example 2.4. Considering still the function used in example 2.3 and the tensor (matrix) $\mathcal{S}_d$. Keeping only the $\mathbf{u}_i^{(n)}$ columns whose singular values are greater than 1×10^{-5} times the greatest singular value of that mode, we get a representation with $U^{(1)} \in \mathbb{R}^{201\times 2}$, $U^{(2)} \in \mathbb{R}^{201\times 2}$ and $\mathcal{S} \in \mathbb{R}^{2\times 2}$.

Thinking of the $U^{(n)}$ matrices, presented in Figure 2.3, as weighting functions of the grid's n-th variable, we have that, inside of the approximation domain, the function $f(x,y) = xy(1+\sin(x)\cos(y))$ can be represented by the weighted sum of the elements of the core tensor $\mathcal{S}$ with the $\mathbf{u}_i^{(n)}$ columns as weighting functions.

In this example's case, we have:

$$f(x,y) = \sum_{i=1}^{2}\sum_{j=1}^{2} \mathbf{u}_i^{(1)} \mathbf{u}_j^{(2)} \mathcal{S}_{ij}.$$

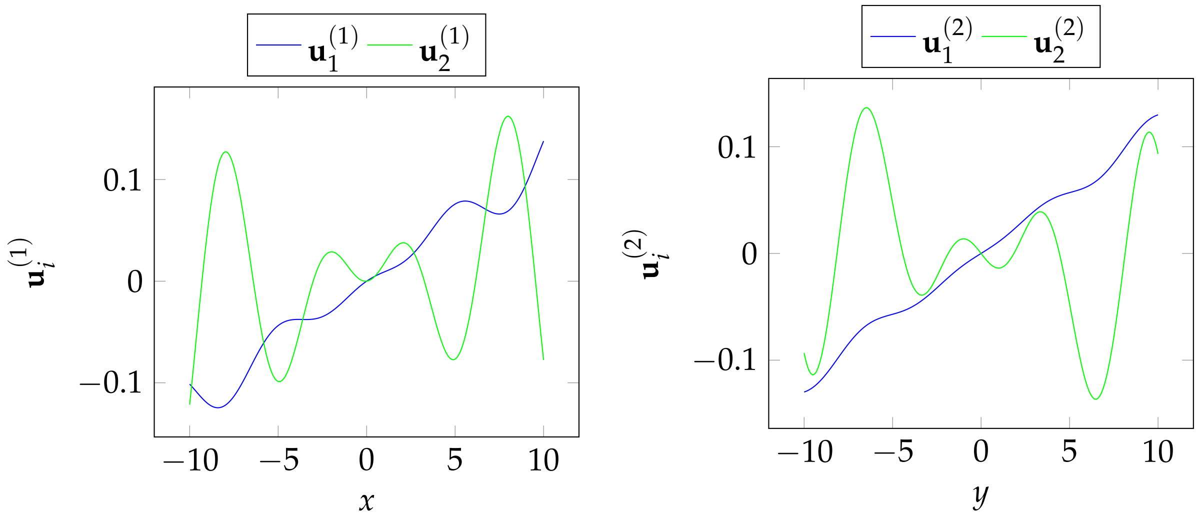

Figure 2.3: weighting functions from example 2.4.

Convex representations (convex hull manipulation)

The use of the HOSVD allowed us to decompose the function as a weighted sum of one-variable terms. However, in some cases, we wish for a representation with additional characteristics.

In order to obtain a representation similar to a TS model, it is necessary for the weighting functions to have properties similar to those of a fuzzy set's membership functions. In the case of the tensor product model transformation, these properties are imposed as certain characteristics of the weight matrices $U^{(n)}$, which, in turn, can be seen as characteristics of the core tensor $\mathcal{S}$.

For the purpose of imposing characteristics over a given weight matrix, it is multiplied on the right by a square transformation matrix. For this transformation to be valid, it is necessary that the transformation matrix be invertible.

Considering $\widetilde{U}^{(n)} = U^{(n)} T_n$, we have that $U^{(n)} = \widetilde{U}^{(n)} T_n^{-1}$. By making use of properties 2.3 and 2.4, we can show that

$$\mathcal{S} \underset{n=1}{\overset{N}{\times}} U^{(n)} = \widetilde{\mathcal{S}} \underset{n=1}{\overset{N}{\times}} \widetilde{U}^{(n)},$$

with $\widetilde{\mathcal{S}}$ given by

$$\widetilde{\mathcal{S}} = \mathcal{S} \underset{n=1}{\overset{N}{\times}} T_n^{-1}.$$

In the literature, the characteristics are usually defined on the weighting functions (obtained after interpolating the column weight matrices). However, since the transformations used to impose these characteristics are applied to the weight matrices, in this work, we define these characteristics on the weight matrices directly.

In the following, we present the most common characteristics used in the literature (Petres et al. 2005) and why they are desirable.

Sum Normalized and Non Negative

Definition 2.6: Sum Normalized

Sum Normalized (SN) - A matrix $U^{(n)}$ is said sum normalized if the sum of its columns, $\mathbf{u}_i^{(n)}$, results in a vector whose components are all equal to one (represented by $\mathbf{1}$). Written differently, a matrix $U^{(n)}$ with r columns is said sum normalized if

$$\sum_{i=1}^{r} \mathbf{u}_i^{(n)} = \mathbf{1}.$$

Definition 2.7: Non Negative

Non Negative (NN) - A matrix $U^{(n)}$ is said non negative if none of its elements is negative. Written differently, a matrix $U^{(n)}$ is said non negative if

$$u_{ij}^{(n)} \geq 0, \quad \forall i, j.$$

When all weight matrices are SN and NN, the resulting model is equivalent to a TS model and the weighting functions (corresponding to the columns of the weight matrices) are equivalent to the membership functions of a TS model. In that regard, whenever we would like to find a TS fuzzy approximation, all weight matrices must be SN and NN.

An algorithm that calculates, separately, the SN and NN transformation of the weight matrices is presented in (Baranyi 1999).

Example 2.5. Considering the weight matrices obtained in example 2.4 and applying a transformation to make them SN and NN, we get the weighting function in Figure 2.4. It is interesting to note that in both cases it was not possible to obtain a transformation without increasing the number of columns of the weight matrices (and, accordingly, a new function has been added to each plot in Figure 2.4 comparing to the ones in Figure 2.3).

Remark 2.2

In order for the SN transformation to be possible without increasing the number of columns, it is necessary that the vector $\mathbf{1}$ belongs to the image space of the weight matrix we would like to transform. When it does not, it is necessary to include a new column so that the image space will include the $\mathbf{1}$ vector. In this case, the SN transformation has a trivial solution given by the inclusion of the column $\mathbf{1} - \sum_{i=1}^{r} \mathbf{u}_i^{(n)}$.

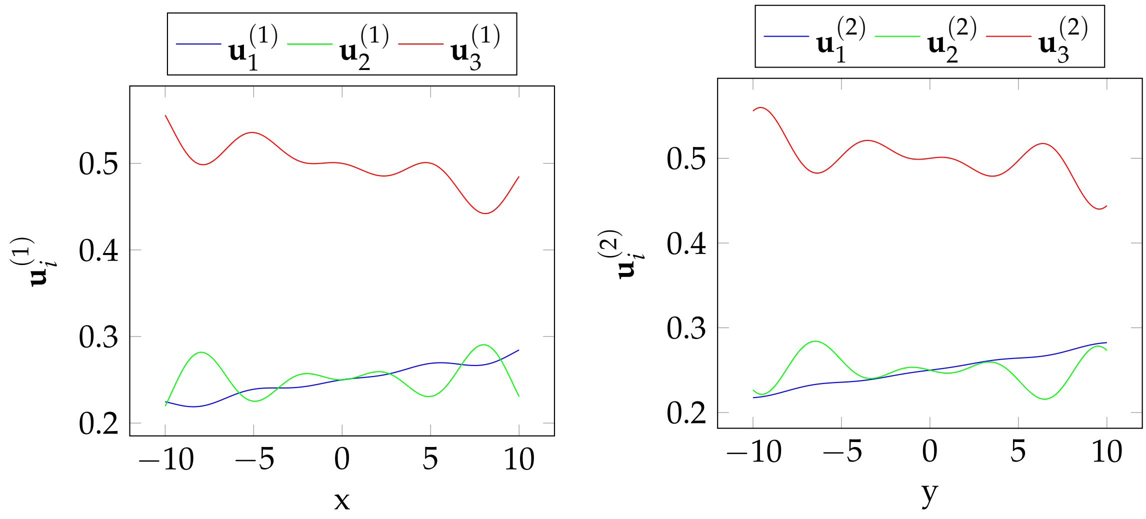

Figure 2.4: SN-NN weighting functions obtained in example 2.5.

Normalized and Close to Normalized

Definition 2.8: Normalized

Normalized (NO) - A matrix $U^{(n)}$ is said normalized if it is SN and NN and, in addition, the maximum values of each column, $\mathbf{u}_i^{(n)}$, are the same and equal to one. Written differently, a matrix $U^{(n)}$ is said normalized if it is SN, NN and

$$\max(\mathbf{u}_i^{(n)}) = 1, \quad \forall i.$$

When the weight matrices are NO, the set represented by the polytope formed by the vertices stored in the core tensor is equivalent to the convex hull of the sampled points used for the tensor product model transformation. However, in most cases, this characteristic leads to a very large number of columns, thus increasing the approximation's complexity.

An algorithm for the NO transformation of a given SN and NN weight matrix is presented in (Baranyi 1999). As presented in the paper, the problem of finding this transformation boils down to finding the convex hull of the points formed by the rows of the weight matrix.

There are many packages available online (*e.g.: Qhull* 2011) for finding the convex hull vertices of a given point cloud.

Definition 2.9: Close to Normalized - usual definition

Close to Normalized (CNO) - A matrix $U^{(n)}$ is said close to normalized if it is SN, NN and the maximum values of each column are close to one.

In the case of the CNO representation, the set formed by the core tensor polytope is not equivalent to the convex hull anymore. In this case, however, we search for a representation with a fixed number of rules (weight matrices columns), that is close to the NO representation, such that the core tensor polytope is close to the convex hull. For that reason, in most cases, we search for a CNO representation instead of NO one, because, usually we seek a representation with reduced complexity.

Since we are searching for a representation whose polytope is close to the convex hull in the CNO representation, in this work we make use of an alternative definition for this characteristic.

Definition 2.10: Close to Normalized - alternative definition

CNO - A matrix $U^{(n)}$ is said close to normalized if it is SN, NN, and the convex hull of its row vectors has the largest possible volume inside of the standard unit simplex (*i.e.* it is not possible to enlarge the convex hull without violating the SN and NN conditions).

Based on this definition, we have that the CNO transformation will be given by the smallest volume simplex that covers the row vectors of the weight matrix. Transforming this simplex into the standard unit simplex, we get a CNO matrix.

CNO Transformation

The following presentation is a transposed version of the presentation in (Campos et al. 2013).

In order to reduce the computational cost of the optimization problem, one may find the convex hull of the row vectors, and keep only its vertices in order to find the transformation (since if the smallest volume simplex covers the convex hull, it will also cover all points). Note that this is an optional step (and in high dimensional cases may be quite costly by itself) and the transformation can be calculated using all the points.

The aim of this transformation is finding a matrix K such that

$$U = \tilde{U}K, \tag{2.9}$$

and matrix $\tilde{U}$ is SN and NN.

That is the same as imposing that the rows of $\tilde{U}$ have the *convex sum property* and imposing that

$$\begin{aligned} \text{sum}(UT) &= \mathbf{1}, \\ UT &\succeq 0, \end{aligned}$$

in which $T = K^{-1}$, and the first constraint means that the sum of the rows of UT must be equal to a vector of ones.

On the other hand, the volume v of a $r-1$ dimensional simplex described by r vertices is given by (Stein 1966)

$$v = \frac{1}{(r-1)!}\det\left(\begin{bmatrix} k_{11} & k_{12} & \dots & k_{1(r-1)} & 1 \\ k_{21} & k_{22} & \dots & k_{2(r-1)} & 1 \\ \vdots & \vdots & \vdots & \vdots & 1 \\ k_{r1} & k_{r2} & \dots & k_{r(r-1)} & 1 \end{bmatrix}\right).$$

But, since the rows of K must satisfy the *convex sum property*, we get

$$v = \frac{1}{(r-1)!}\det\left(K\begin{bmatrix} I_{r-1} & \mathbf{1_{r-1}} \\ \mathbf{0}_{r-1}^T & 1 \end{bmatrix}\right) = \frac{1}{(r-1)!}\det(K).$$

Therefore, in order to find the smallest volume simplex that covers the points in the rows of U, we want to minimize

$$J = \det(K) = \det(T)^{-1}.$$

Finally, we can enunciate the result below.

Theorem 2.2: CNO Transformation

The transformation matrix T that endows the weight matrix U with the CNO characteristic can be found by solving the following optimization problem

$$\begin{aligned} \min_{T} \quad & -\ln(\det(T)) \\ s.t. \quad & sum(UT) = \mathbf{1}, \\ & UT \succeq 0. \end{aligned}$$

Example 2.6. Still considering the weight matrices obtained in example 2.4 and applying the CNO transformation described above, we get the weighting functions presented in Figure 2.5.

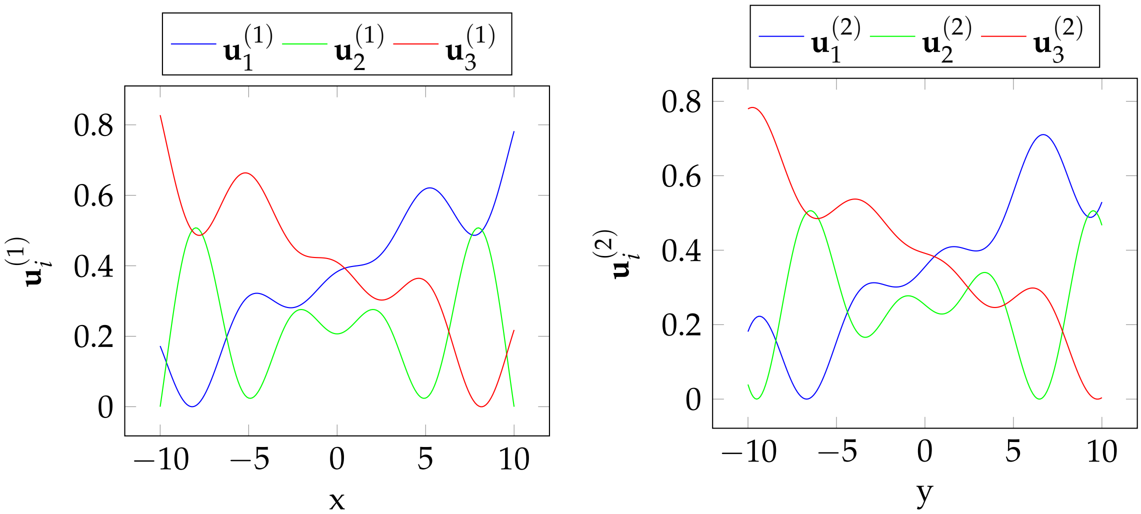

Figure 2.5: CNO weighting functions obtained in example 2.6.

Relaxed Normalized and Inverse Normalized

Definition 2.11: Relaxed Normalized

Relaxed Normalized (RNO) - A matrix $U^{(n)}$ is said relaxed normalized if it is SN, NN, and in addition the maximum value of all of its columns, $\mathbf{u}_i^{(n)}$, are the same. Written differently, a matrix $U^{(n)}$ is said relaxed normalized if it is SN, NN, and

$$\max(\mathbf{u}_i^{(n)}) = \max(\mathbf{u}_j^{(n)}), \quad \forall i, j.$$

Definition 2.12: Inverse Normalized

Inverse Normalized (INO) - A matrix $U^{(n)}$ is said inverse normalized if it is SN, NN and in additions the minimum of each of its columns $\mathbf{u}_i^{(n)}$ are the same and equal to zero. Written differently, a matrix $U^{(n)}$ is said inverse normalized if it is SN, NN and

$$\min(\mathbf{u}_i^{(n)}) = 0, \quad \forall i.$$

Just like the CNO characteristic, the RNO and INO characteristics are relaxed forms of imposing characteristics over the polytope in a way that it represents a smaller set that the one we would get when imposing only the SN and NN characteristics over the weight matrices.

It is noteworthy that, for matrices with two columns, the INO condition is equivalent to the NO condition. In addition, imposing that all columns of the weight matrices have the same maximum value (RNO) together with imposing that all columns have a minimum value equal to zero (INO) implies in well distributed weighting functions, so that no weighting function has more importance over another (so the same can be said of the corresponding vertices).

An algorithm for finding, simultaneously, a RNO and INO weight matrix, given a SN and NN matrix, is presented in (Varkonyi et al. 2005).

Example 2.7. Considering the weight matrices obtained in example 2.5 (since the transformation used in this stage requires a SN and NN matrix) and applying the RNO-INO transformation, we get the weighting functions presented in Figure 2.6.

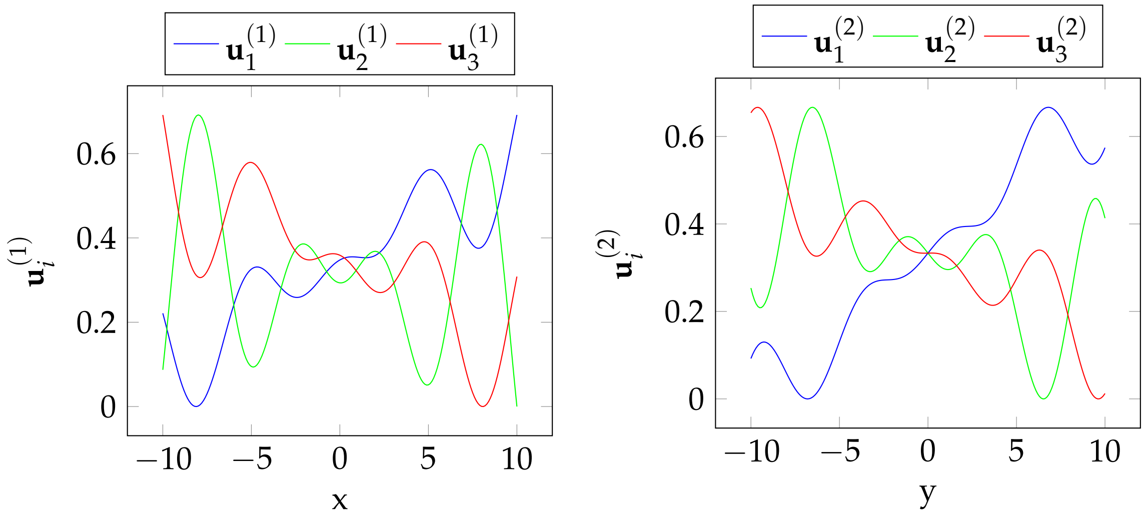

Figure 2.6: RNO-INO weighting functions obtained in example 2.7.

It is interesting to note that, as commented above, we found functions with similar representativeness for all vertices. Which is a little different from what happened in the CNO transformation, and justifies the use of the RNO-INO in certain cases.

It is interesting to note that all these representations are equivalent, and therefore generate the same approximation for function $f(x,y)$. As a matter of completeness, Figure 2.7 presents the squared approximation error obtained by using the RNO-INO weighting functions (which is the same for all the other representations).

Approximating the weight matrices

Having found the weight matrices with the desired characteristics, it is necessary to find functions that replace each of the columns of each weight matrix, such that the approximation found can be employed for points that do not belong to the sampling grid.

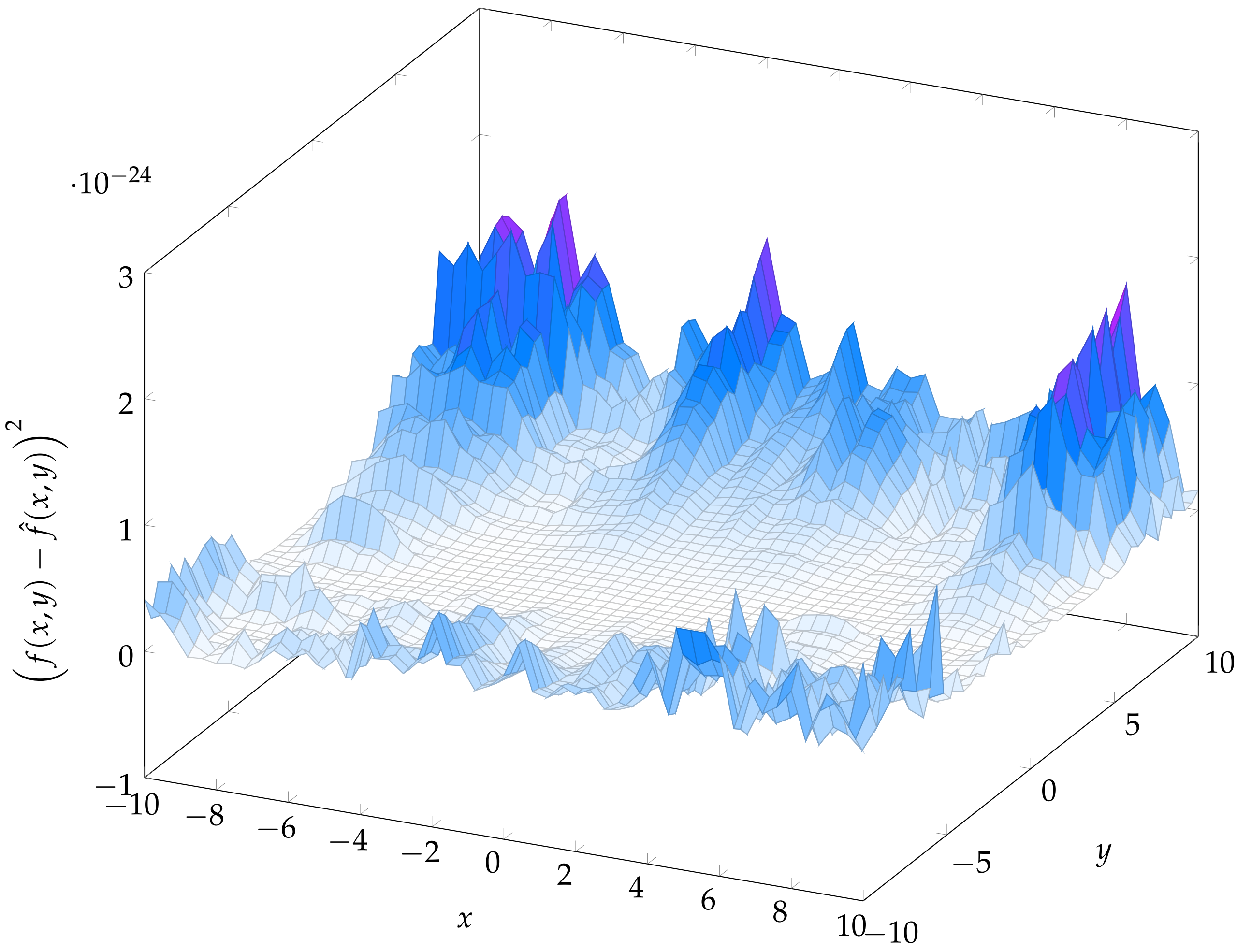

Figure 2.7: Squared approximation error when using the weight functions from example 2.7.

From equation (2.8):

$$
\begin{aligned}
f(\mathbf{x}) &\approx \mathcal{S}_d \bigtimes_{n=1}^{N} \boldsymbol{\lambda}^{(n)}(x_n),\\
f(\mathbf{x}) &\approx \left(\mathcal{S} \bigtimes_{n=1}^{N} U^{(n)}\right) \bigtimes_{n=1}^{N} \boldsymbol{\lambda}^{(n)}(x_n),\\
f(\mathbf{x}) &\approx \mathcal{S} \bigtimes_{n=1}^{N} (\boldsymbol{\lambda}^{(n)}(x_n)U^{(n)}),\\
f(\mathbf{x}) &\approx \mathcal{S} \bigtimes_{n=1}^{N} \mathbf{u}^{(n)}(x_n),
\end{aligned}
\tag{2.10}
$$

in which $\mathbf{u}^{(n)}(x_n)$ is a row vector whose components are given by the interpolation of the columns of $U^{(n)}$.

This result shows that the same weighting functions that were considered to generate an interpolation of the original sampling values are used to generate the interpolation of the weighting matrices and, in turn, generate the membership functions. In that regard, we seek interpolation schemes whose weighting functions have properties similar to normalized membership functions (*i.e.* they are non negative and they add up to one).

In the following we present some weighting functions that could be used, initially presented in (Campos, Tôrres, and Palhares 2015). They are illustrated in Figure 2.8.

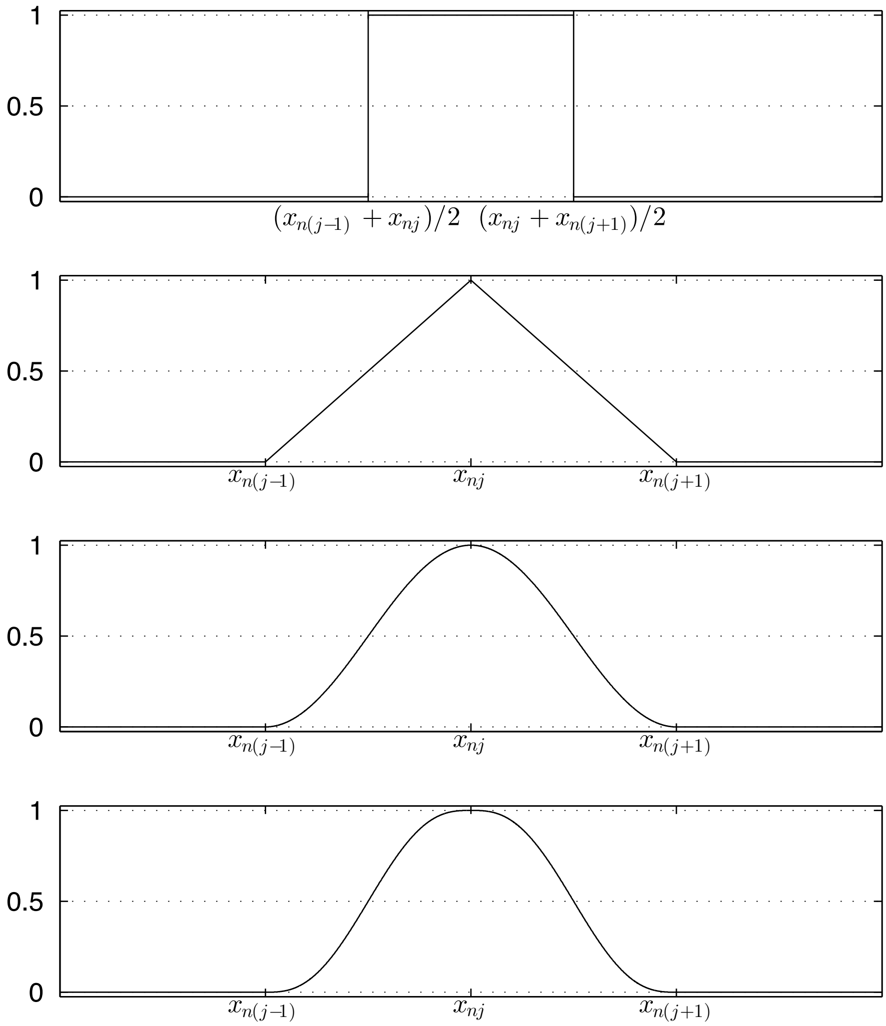

Figure 2.8: Illustration of possible weighting functions, $\lambda_j^{(n)}(x_n)$, that can be used in the sampling step. The first one illustrates a rectangular (piecewise constant) membership function. The second, a triangular (piecewise linear) membership function. The third, a piecewise sinusoidal function. And the last, a piecewise fifth degree polynomial function.

In the definition of the interpolation weighting functions $\lambda_j^{(n)}(x_n)$ below, we consider the sampling grid over $[\underline{x}_n, \overline{x}_n]$ to be defined by samples x_{nj} with $j \in 1, 2, ..., I_n$ and $x_{n(j-1)} < x_{nj}$.

Rectangular (Piecewise Constant) Functions The simplest membership function shape. Guarantees that the approximation error between the approximation and the original function in equation (2.8) gets smaller as the number of samples increases. This weighting function can be written as

$$\lambda_j^{(n)}(x_n) = \begin{cases} 1, & x_n \in \left[\frac{x_{n(j-1)}+x_{nj}}{2}, \frac{x_{nj}+x_{n(j+1)}}{2}\right], \\ 0, & \text{otherwise.} \end{cases}$$

Triangular (Piecewise Linear) Functions A commonly used weighting function shape (Baranyi et al. 2003), that in the sampling step is equivalent to making a linear interpolation of the samples. Guarantees the continuity of the interpolated functions. This weighting function can be written as

$$\lambda_j^{(n)}(x_n) = \begin{cases} \frac{x_n - x_{n(j-1)}}{x_{nj} - x_{n(j-1)}}, & x_n \in \left[x_{n(j-1)}, x_{nj}\right], \\ \frac{-x_n + x_{n(j+1)}}{x_{n(j+1)} - x_{nj}}, & x_n \in \left[x_{nj}, x_{n(j+1)}\right], \\ 0, & \text{otherwise.} \end{cases}$$

Piecewise Sinusoidal Functions A weighting function shape that guarantees that the interpolated membership functions will be continuously differentiable. This weighting function can be described by

$$\lambda_j^{(n)}(x_n) = \begin{cases} \frac{1}{2}\left(1 + \sin\left(\frac{\pi\left(2x_n - x_{nj} - x_{n(j-1)}\right)}{2\left(x_{nj} - x_{n(j-1)}\right)}\right)\right), & x_n \in \left[x_{n(j-1)}, x_{nj}\right], \\ \frac{1}{2}\left(1 - \sin\left(\frac{\pi\left(2x_n - x_{n(j+1)} - x_{nj}\right)}{2\left(x_{n(j+1)} - x_{nj}\right)}\right)\right), & x_n \in \left[x_{nj}, x_{n(j+1)}\right], \\ 0, & \text{otherwise.} \end{cases}$$

Piecewise Fifth Degree Polynomial A weighting function shape that guarantees the continuity of the second derivative of the interpolated membership functions. This weighting function can be described by

$$\lambda_j^{(n)}(x_n) = \begin{cases} 1 - \frac{6x_n^5 + a_1 x_n^4 + a_2 x_n^3 + a_3 x_n^2 + a_4 x_n + a_5}{\left(x_{n(j-1)} - x_{nj}\right)^5}, & x_n \in \left[x_{n(j-1)}, x_{nj}\right], \\ \frac{6x_n^5 + b_1 x_n^4 + b_2 x_n^3 + b_3 x_n^2 + b_4 x_n + b_5}{\left(x_{nj} - x_{n(j+1)}\right)^5}, & x_n \in \left[x_{nj}, x_{n(j+1)}\right], \\ 0, & \text{otherwise.} \end{cases}$$

with

$$
\begin{aligned}
a_1 &= -15\left(x_{n(j-1)} + x_{nj}\right),\\
a_2 &= 10\left(x_{n(j-1)}^2 + 4x_{n(j-1)}x_{nj} + x_{nj}^2\right),\\
a_3 &= -30x_{n(j-1)}x_{nj}\left(x_{n(j-1)} + x_{nj}\right),\\
a_4 &= 30x_{n(j-1)}^2 x_{nj}^2,\\
a_5 &= -x_{nj}^3\left(10x_{n(j-1)}^2 - 5x_{n(j-1)}x_{nj} + x_{nj}^2\right),\\
b_1 &= -15\left(x_{nj} + x_{n(j+1)}\right),\\
b_2 &= 10\left(x_{nj}^2 + 4x_{nj}x_{n(j+1)} + x_{n(j+1)}^2\right),\\
b_3 &= -30x_{nj}x_{n(j+1)}\left(x_{nj} + x_{n(j+1)}\right),\\
b_4 &= 30x_{nj}^2 x_{n(j+1)}^2,\\
b_5 &= -x_{n(j+1)}^3\left(10x_{nj}^2 - 5x_{nj}x_{n(j+1)} + x_{n(j+1)}^2\right).
\end{aligned}
$$

For simplification purposes, whenever we do not say otherwise, in this work we employ piecewise linear weighting functions for the columns of the weight matrices. These functions can be implemented as a linear interpolation table, with the domain points given by the grid points in the direction corresponding to the direction of the weight matrix, and the image points given by the values in the columns of the weight matrices.

It is possible to rewrite (2.10) as:

$$f(\mathbf{x}) \approx \sum_{i_1=1}^{I_1} \ldots \sum_{i_N=1}^{I_N} \mathbf{u}_{i_1}^{(1)}(x_1)\ldots\mathbf{u}_{i_N}^{(N)}(x_N)\mathcal{S}_{i_1\ldots i_N}, \tag{2.11}$$

in which $\mathcal{S}_{i_1\ldots i_N}$, just like $f(\mathbf{x})$, belongs to $\mathbb{R}^{M_1\times\cdots\times M_m}$. If the weight matrices used in the interpolation step have the SN and NN characteristics, this expression shows that $f(\mathbf{x})$ can be approximated by a convex combination of the $\mathcal{S}_{i_1\ldots i_N}$ points. In this case, these points are called the vertices of the convex representation.

Example applied to Dynamical Systems

When applying the tensor product model transformation to dynamical system, we usually have the following situation: given a qLPV model of a nonlinear dynamical system find a TS fuzzy representation of the system.

Consider the qLPV model given by:

$$
\begin{aligned}
\dot{\mathbf{x}} &= A(\mathbf{x})\mathbf{x} + B(\mathbf{x})\mathbf{u},\\
\mathbf{y} &= C(\mathbf{x})\mathbf{x} + D(\mathbf{x})\mathbf{u},
\end{aligned}
$$

in which $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^k$, $A(\mathbf{x}) \in \mathbb{R}^{n\times n}$, $B(\mathbf{x}) \in \mathbb{R}^{n\times m}$, $C(\mathbf{x}) \in \mathbb{R}^{k\times n}$ and $D(\mathbf{x}) \in \mathbb{R}^{k\times m}$.

A possible approach would be to define the function

$$f(\mathbf{x}) = \begin{bmatrix} A(\mathbf{x}) & B(\mathbf{x}) \\ C(\mathbf{x}) & D(\mathbf{x}) \end{bmatrix}$$

and inside of an hyper-rectangular region , find an approximation as was done previously. In this way, we end up with a TS representation for the system.

In this case, $\mathcal{S}_{i_1 \ldots i_N}$ represents the matrices of the vertex linear systems that compose the TS model of the system.

However, in most of the cases, not all the states in $\mathbf{x}$ affect the values of the A, B, C and D matrices and in some cases some of these matrices may be constants.

Thus, in most cases, it is useful to find a subset of the state vector, $\check{\mathbf{x}}$, that influences the system matrices and find an approximation for the function $f(\check{\mathbf{x}})$, reducing the final size of the sampling grid. In addition to this, in order to alleviate the computational burden, form a function containing only the non constant terms from the matrices.

When we do not include all matrices to form the $f(\check{\mathbf{x}})$ function, $\mathcal{S}_{i_1 \ldots i_N}$ will contain only the ones that were included. However, since the other terms are constant, they are the same for all the vertex systems.

Some recent works (Baranyi et al. 2007; Nagy et al. 2009) present other ways to reduce the computational effort of the tensor product model transformation. However, this reduction is outside of the scope of this work and, therefore, is not dealt with here.

Example 2.8. Consider the kinematic equations of a mobile robot whose movement has been constrained to the xy plane:

$$\begin{aligned} \dot{x} &= v\cos(\theta), \\ \dot{y} &= v\sin(\theta), \\ \dot{\theta} &= w. \end{aligned}$$

In which x is the robot's position in the x direction, y is the robot's position in the y direction, θ is the robot's yaw angle, v is the robot's translational velocity, and w its angular velocity. Considering that the state space is composed by $\mathbf{z} = [\ x \quad y \quad \theta\ ]^T$, the input vector is given by $\mathbf{u} = [\ v \quad w\ ]^T$, and the output vector is given by $\mathbf{p} = [\ x \quad y\ ]^T$, we get:

$$\begin{aligned} \dot{\mathbf{z}} &= B(\mathbf{z})\mathbf{u} \\ \mathbf{p} &= C\mathbf{z} \\ B(\mathbf{z}) &= \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned}$$

In this example's case we get to see what was commented above. Out of the three state variables, only one of them, θ, influences the system matrices. In addition to this, only the B matrix changes with the state. The A and D matrices are zero matrices, whereas the C matrix is a constant matrix.

Therefore, the function we must approximate in this example is $B(\theta)$ and we know that the other matrices will be the same for all the vertex systems found in the final model.

Since θ is the yaw angle, we know that $\theta \in [-\pi, \pi]$. In that regard, we have decided to use 101 samples over $\theta \in [-\pi, \pi]$. We have that $B(\theta) \in \mathbb{R}^{3\times 2}$, and, thus, with 101 samples we get a tensor $\mathcal{S}_d \in \mathbb{R}^{101\times 3\times 2}$.

By following the same steps presented previously, it is possible to find several representations for the model. The different weighting functions in each representation are presented in Figure 2.9.

Since all the representations are equivalent, they have the same representation mean square error for the points belonging to the grid. In this example, this error is 1.2321×10^{-25}.

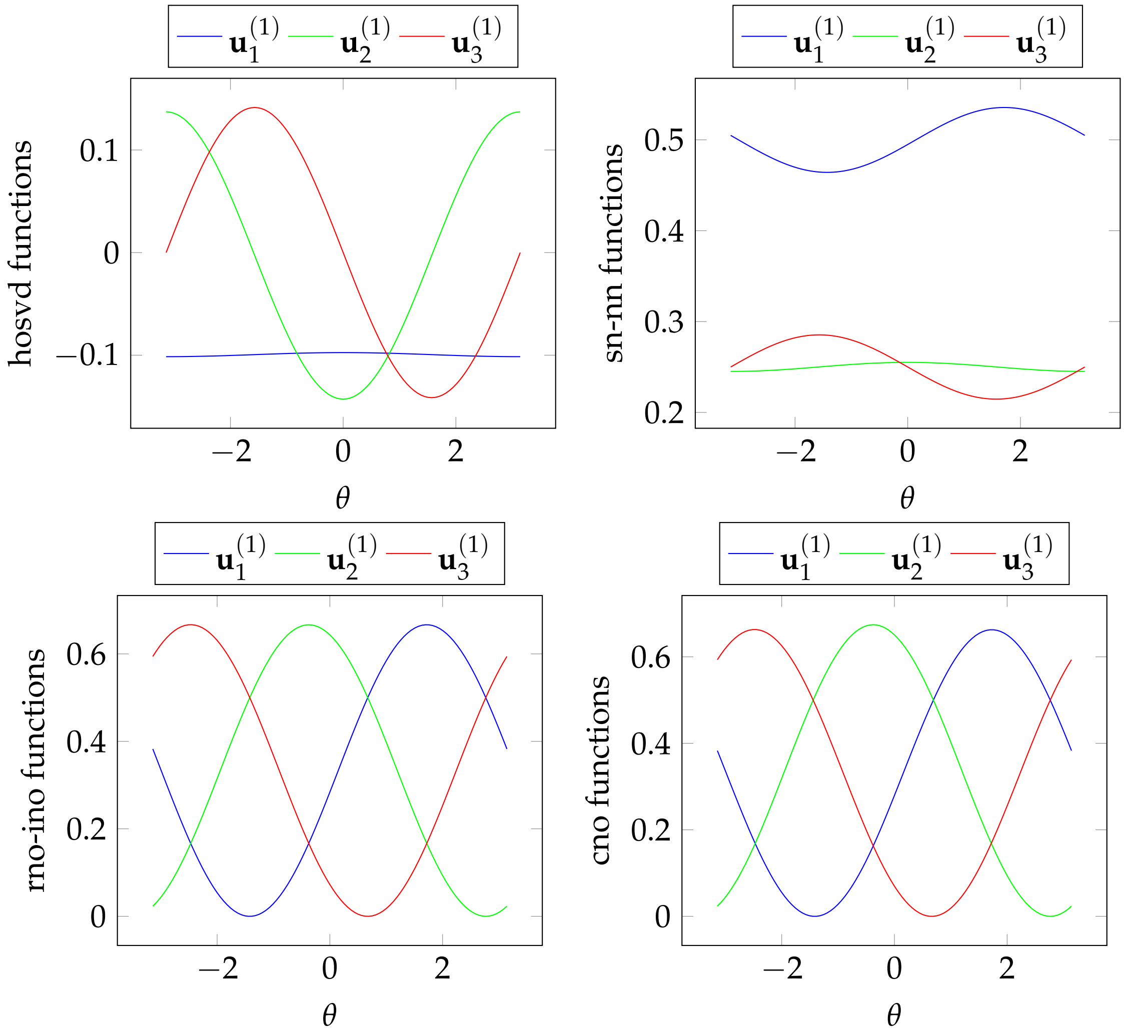

Figure 2.9: Different weighting functions obtained in example 2.8.

2.5 Membership Functions Transformations

Membership functions transformations can be useful in different situations: be it for finding a system representation with certain special properties, be it to apply (even if only locally) lemmas 2.5 and 2.6 to reduce the conservativeness of sufficient conditions.

Transformations combination

When dealing with TS models whose membership functions have a tensor product structure (*i.e.* models in which property 2.1 is valid), one can usually simplify the transformation calculations by finding a transformation for the normalized one-variable membership functions, $\mu_j^{\alpha_{ij}}(v_j)$, for each of the premise variables separately and then combining these to form a transformation of the multivariable normalized membership functions, $h_i(\mathbf{v})$.

In order to do so, consider the column vector $\mathbf{h}$, whose ith element is $h_i(\mathbf{v})$ given in (2.2), and the column vector $\boldsymbol{\mu}_j$, whose elements are $\mu_j^{\alpha_{ij}}(v_j)$, with $1 \leq \alpha_{ij} \leq r_j$. In this case, using the Kronecker product operation, one has that from (2.2)

$$\mathbf{h} = \boldsymbol{\mu}_1(v_1) \otimes \boldsymbol{\mu}_2(v_2) \otimes \cdots \otimes \boldsymbol{\mu}_\ell(v_\ell). \tag{2.12}$$

Consider also the transformed column vectors $\tilde{\mathbf{h}}$ and $\tilde{\boldsymbol{\mu}}_j$ satisfying

$$\tilde{\mathbf{h}} = \tilde{\boldsymbol{\mu}}_1(v_1) \otimes \tilde{\boldsymbol{\mu}}_2(v_2) \otimes \cdots \otimes \tilde{\boldsymbol{\mu}}_\ell(v_\ell),$$

with each transformed vector $\tilde{\boldsymbol{\mu}}_j$ given by a linear transformation such that

$$\boldsymbol{\mu}_j(v_j) = K_j \tilde{\boldsymbol{\mu}}_j(v_j),$$

and $K_j \in \mathbb{R}^{r_j \times \tilde{r}_j}$.

We can therefore rewrite (2.12) as

$$\mathbf{h}(\mathbf{v}) = K_1 \tilde{\boldsymbol{\mu}}_1(v_1) \otimes K_2 \tilde{\boldsymbol{\mu}}_2(v_2) \otimes \cdots \otimes K_\ell \tilde{\boldsymbol{\mu}}_\ell(v_\ell).$$

In addition, by repeatedly employing the Kronecker product property that $(A \otimes B)(C \otimes D) = AC \otimes BD$ (Bernstein 2009)

$$\mathbf{h}(\mathbf{v}) = (K_1 \otimes K_2 \otimes \cdots \otimes K_\ell)\left(\tilde{\boldsymbol{\mu}}_1(v_1) \otimes \tilde{\boldsymbol{\mu}}_2(v_2) \otimes \cdots \otimes \tilde{\boldsymbol{\mu}}_\ell(v_\ell)\right),$$
$$\mathbf{h}(\mathbf{v}) = G\tilde{\mathbf{h}}(\mathbf{v}),$$

with

$$G = (K_1 \otimes K_2 \otimes \cdots \otimes K_\ell), \tag{2.13}$$

and $G \in \mathbb{R}^{r \times \tilde{r}}$.

The aforementioned equations show that if one finds the K_j transformation matrices for each premise variable separately, they can be combined to yield a transformation of the membership functions h_i by taking their Kronecker product.

If, in addition, we impose that the K_j matrices have the *convex sum property*, then the transformed membership functions will also have this property.

Transformations combination - alternative presentation

For completeness, we present here an alternative development for combining these transformations, using the tensor product representation (from the tensor product model transformation presented in this chapter).

Considering a TS model for which the property 2.1 is valid, as well as a convex combination of arbitrary matrices $L_i \in \mathbb{R}^{\kappa_1 \times \kappa_2}$, such that

$$Y(\mathbf{v}) = \sum_{i=1}^{r} h_i(\mathbf{v}) L_i.$$

This relationship can be rewritten as

$$\mathcal{Y}(\mathbf{v}) = \mathcal{L} \mathop{\times}_{j=1}^{\ell} \boldsymbol{\mu}_j^T(v_j), \tag{2.14}$$

in which $\boldsymbol{\mu}_j(v_j)$ is a vector containing the normalized unidimensional membership functions, as defined previously; $\mathcal{L} \in \mathbb{R}^{r_1 \times \cdots \times r_\ell \times \kappa_1 \times \kappa_2}$ is a tensor of order $(\ell + 2)$ containing the L_i matrices such that the first ℓ indices $(k_1, \ldots, k_\ell)$ indicate which fuzzy sets $\mathcal{M}_1^{k_1}, \ldots, \mathcal{M}_\ell^{k_\ell}$ are related to the corresponding rule (*i.e.* $\mathcal{L}$ is a tensor whose element $l_{k_1 \ldots k_\ell k_{\ell+1} k_{\ell+2}}$ correspond to the element in row $k_{\ell+1}$ and column $k_{\ell+2}$ of matrix L_i obtained from rule i, whose premise variables fuzzy sets are $\mathcal{M}_1^{k_1}, \ldots, \mathcal{M}_n^{k_\ell}$); $\mathcal{Y}(\mathbf{v}) \in \mathbb{R}^{1 \times \cdots \times 1 \times \kappa_1 \times \kappa_2}$ is a tensor constructed from matrix $Y(\mathbf{v})$ by putting all indices equal to 1 before the indices of matrix $Y(\mathbf{x})$ (*i.e.* they are different representations of the same object).

In the tensor product model transformation literature, it is usual to transform the vectors $\boldsymbol{\mu}_j(v_j)$ in order to find better suited representations (Baranyi 1999, 2004; Varkonyi et al. 2005). These transformations are done considering each premise variable separately, but can be combined as a single transformation of the $h_i(\mathbf{v})$ membership functions. In that regard, we can generate transformations that can be used with lemmas 2.5 and 2.6.

These transformations usually have the form

$$\tilde{\boldsymbol{\mu}}_j^T(v_j) = \boldsymbol{\mu}_j^T(v_j)T_j,$$

in which the vectors $\tilde{\boldsymbol{\mu}}_j(v_j)$ satisfy the *convex sum property*.

In addition, by using property 2.4 of the *n-mode* product, equation (2.14) can be rewritten as

$$\mathcal{Y}(\mathbf{v}) = (\mathcal{L} \mathop{\times}_{j=1}^{\ell} K_j) \mathop{\times}_{j=1}^{\ell} \tilde{\boldsymbol{\mu}}_j^T(v_j), \tag{2.15}$$

in which

$$K_j = T_j^{-1}.$$

By using the fact that the *n-mode* product between a tensor and the identity matrix in any direction is equal to the tensor, equations (2.14) and (2.15) can be rewritten as

$$\mathcal{Y}(\mathbf{v}) = \mathcal{L} \mathop{\times}_{j=1}^{\ell} \boldsymbol{\mu}_j^T(v_j) \times_{\ell+1} I_{\kappa_1} \times_{\ell+2} I_{\kappa_2}, \tag{2.16}$$

$$\mathcal{Y}(\mathbf{v}) = \mathcal{H} \mathop{\times}_{j=1}^{\ell} \tilde{\boldsymbol{\mu}}_j^T(v_j) \times_{\ell+1} I_{\kappa_1} \times_{\ell+2} I_{\kappa_2}, \tag{2.17}$$

in which

$$\mathcal{H} = (\mathcal{L} \mathop{\times}_{j=1}^{\ell} K_j \times_{\ell+1} I_{\kappa_1} \times_{\ell+2} I_{\kappa_2}). \tag{2.18}$$

Note that the $\ell + 1$ unfolding matrix of $\mathcal{Y}$, denoted by $\mathcal{Y}_{(\ell+1)}$ is equivalent to the $Y(\mathbf{v})$ matrix, and using the unfolding matrix of the right side of equation (2.16), we get

$$\begin{aligned} Y(\mathbf{v}) &= I_{\kappa_1} \mathcal{L}_{(\ell+1)} (I_{\kappa_2} \otimes \boldsymbol{\mu}_1^T(v_1) \otimes \cdots \otimes \boldsymbol{\mu}_\ell^T(v_\ell))^T, \\ Y(\mathbf{v}) &= \mathcal{L}_{(\ell+1)} \left[I_{\kappa_2} \otimes (\boldsymbol{\mu}_1^T(v_1) \otimes \cdots \otimes \boldsymbol{\mu}_\ell^T(v_\ell))^T \right], \\ Y(\mathbf{v}) &= \mathcal{L}_{(\ell+1)} \left[I_{\kappa_2} \otimes \mathbf{h} \right], \end{aligned} \tag{2.19}$$

with $\mathbf{h}$ given by equation (2.12).

By using the unfolding matrix in (2.18), we get

$$\begin{aligned} \mathcal{H}_{(\ell+1)} &= I_{\kappa_1} \mathcal{L}_{(\ell+1)} (I_{\kappa_2} \otimes K_1 \otimes \cdots \otimes K_\ell)^T, \\ \mathcal{H}_{(\ell+1)} &= \mathcal{L}_{(\ell+1)} \left[I_{\kappa_2} \otimes (K_1 \otimes \cdots \otimes K_\ell)^T \right], \\ \mathcal{H}_{(\ell+1)} &= \mathcal{L}_{(\ell+1)} \left[I_{\kappa_2} \otimes G \right], \end{aligned} \tag{2.20}$$

in which

$$G = (K_1 \otimes \cdots \otimes K_n)^T. \tag{2.21}$$

By making use of (2.19) and (2.17), we get

$$Y(\mathbf{v}) = \mathcal{H}_{(\ell+1)} \left[I_{\kappa_2} \otimes \tilde{\mathbf{h}} \right], \tag{2.22}$$

with

$$\tilde{\mathbf{h}} = (\tilde{\boldsymbol{\mu}}_1^T(v_1) \otimes \cdots \otimes \tilde{\boldsymbol{\mu}}_\ell^T(v_\ell))^T.$$

From (2.19), (2.20) and (2.22)

$$\mathcal{L}_{(\ell+1)}\left[I_{\kappa_2}\otimes \mathbf{h}\right] = \mathcal{L}_{(\ell+1)}\left[I_{\kappa_2}\otimes G\right]\left[I_{\kappa_2}\otimes \tilde{\mathbf{h}}\right],$$
$$\left[I_{\kappa_2}\otimes \mathbf{h}\right] = \left[I_{\kappa_2}\otimes G\right]\left[I_{\kappa_2}\otimes \tilde{\mathbf{h}}\right],$$

and applying the Kronecker product property that $(A \otimes B)(C \otimes D) = AC \otimes BD$ (Bernstein 2009)

$$I_{\kappa_2}\otimes \mathbf{h} = I_{\kappa_2}\otimes G\tilde{\mathbf{h}}, \quad \Rightarrow \quad \mathbf{h} = G\tilde{\mathbf{h}}.$$

This last equation shows that the G matrix defines a linear relationship between $\mathbf{h}$ and $\tilde{\mathbf{h}}$.

Remark 2.3

Note that the G matrix given by equation (2.21) seems to be given by the transpose of the G matrix given in (2.13). They are however the same G matrix. The transpose appears in equation (2.21) due to the form the K_j matrices are usually defined in the tensor product model transformation literature (matrices that transform row vectors) instead of the usual form used in the previous section (matrices that transform column vectors).

2.6 Usual Uncertainty Representations

The uncertainty representations for TS models can usually be divided in two categories:

- *Models with uncertainty in the consequent*: are the usual form found in the literature, and possess a structure similar to that found in uncertainty representations usually found in the robust control of linear systems;
- *Models with uncertainty in the premise*: are a less usual form of representing uncertainty.

In the following some examples of uncertainty representations found in the literature are presented.

Uncertainty in the consequent

Norm Bounded Uncertainty A simple extension of the norm bounded uncertainty usually found in the robust control of linear systems (Tanaka and Wang 2001, chapter 5), (Mansouri et al. 2009). Format:

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{z})\left[\left(A_i + D_{a_i}\Delta_{a_i}E_{a_i}\right)\mathbf{x} + \left(B_i + D_{b_i}\Delta_{b_i}E_{b_i}\right)\mathbf{u}\right],$$
$$||\Delta_{a_i}||_2 \leq 1, \qquad ||\Delta_{b_i}||_2 \leq 1$$

Lumped (matched) Uncertainty An uncertainty representation usually found in works dealing with variable structure control (Lin et al. 2007, chapter 5), (Choi 2008), (Choi 2010). Example of format:

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{z})\left[A_i\mathbf{x} + B_i\left(I + \Delta_i\right)\mathbf{u} + \mathbf{f}_i(\mathbf{x},\mathbf{u},t)\right],$$
$$\Delta_i \text{ bounded}, \quad \mathbf{f}_i = B_i\tilde{\mathbf{f}}_i, \quad ||\tilde{\mathbf{f}}|| \text{ bounded}$$

Structured Linear Fractional Form An extension of the norm bounded uncertainty that manages to be less conservative by being more structured. (Feng 2011, chapter 8), (Zhang, Zhou, and Li 2007), (Zhou et al. 2005). Format:

$$\begin{aligned}
&\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{z}) \left[\tilde{A}_i \mathbf{x} + \tilde{B}_i \mathbf{u}\right], \\
&\left[\begin{array}{cc} \tilde{A}_i & \tilde{B}_i \end{array}\right] = \left[\begin{array}{cc} A_i & B_i \end{array}\right] + W_i \Delta \left[\begin{array}{cc} N_{1_i} & N_{2_i} \end{array}\right] \\
&\Delta = F\left[I - JF\right]^{-1} \quad \text{or} \quad \Delta = \left[I - JF\right]^{-1} F \\
&I - JJ^T > 0 \quad \text{and } J \text{ known} \\
&||F||_2 \leq 1
\end{aligned}$$

Polytopic Uncertainty Despite being heavily used in the robust control of linear systems, this representation (Sofianos and Kosmidou 2009) is seldom used for TS systems due to the increase in the number of LMIs. Format:

$$\begin{aligned}
&\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{z}) \left[(A_i + \Delta A_i)\,\mathbf{x} + (B_i + \Delta B_i)\,\mathbf{u}\right], \\
&\left[\Delta A_i \quad \Delta B_i\right] = \sum_{k=1}^{m} \alpha_{ik} \left[\tilde{A}_{ik} \quad \tilde{B}_{ik}\right], \\
&\alpha_{ik} \geq 0, \quad \sum_{k=1}^{m} \alpha_{ik} = 1
\end{aligned}$$

Elemental Parametric Uncertainty An uncommon representation (Chen, Ho, and Chou 2009; Ho, Tsai, and Chou 2007) and quite similar to the polytopic uncertainty (in the sense that one can be easily converted into the other). Format:

$$\begin{aligned}
&\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{z}) \left[(A_i + \Delta A_i)\,\mathbf{x} + (B_i + \Delta B_i)\,\mathbf{u}\right], \\
&\left[\Delta A_i \quad \Delta B_i\right] = \sum_{k=1}^{m} \alpha_{ik} \left[\tilde{A}_{ik} \quad \tilde{B}_{ik}\right], \\
&\underline{\alpha_{ik}} \leq \alpha_{ik} \leq \overline{\alpha_{ik}}
\end{aligned}$$

Uncertainty in the premise

The use of representations of this form is less common (though lately it has been gaining more attention), due to the fact that it has no equivalent in the robust control of linear systems. Unlike the other representations, that lump all uncertainty into the local linear models, the uncertainty is lumped in the membership functions of the model.

A specific format is presented in the following.

Interval Type 2 TS Fuzzy Model

This representation makes use of TS interval type 2 fuzzy models (that can be seen as a collection of TS models) to represent the model's uncertainty (Lam and Leung 2011, chapter 10).

In this representation case, the membership of each rule returns a closed interval, instead of returning a scalar.

$$h_i = \left[\underline{h}_i, \overline{h}_i\right].$$

In order to represent the uncertainty in the memberships, at each instant, another "membership" (that many times is unknown) selects values from these intervals to do the model's inference. The inferred model can, therefore, be written as:

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} \left(\underline{\nu}_i(\mathbf{z})\underline{h}_i(\mathbf{z}) + \overline{\nu}_i(\mathbf{z})\overline{h}_i(\mathbf{z}) \right) [A_i\mathbf{x} + B_i\mathbf{u}] = \sum_{i=1}^{r} \tilde{h}_i(\mathbf{z}) [A_i\mathbf{x} + B_i\mathbf{u}] .$$

Since the unknown "membership", $\underline{\nu}_i$ and $\overline{\nu}_i$, chooses values inside of the interval for each rule, we have that $\underline{\nu}_i + \overline{\nu}_i = 1$. Note that the limit values of the intervals generated by the memberships do not have to necessarily obey the characteristics of a normalized membership function (*i.e.* they do not necessarily belong to the standard unit simplex), but the "real membership functions", $\tilde{h}_i$, have to.

Part I

Premise variables are free

3 Exploiting the membership functions

In the last years, several asymptotically exact conditions for fuzzy summations have been proposed in the literature (Kruszewski et al. 2009; Lo and Liu 2010; Montagner, Oliveira, and Peres 2009; Sala 2007). However, for not taking into account the shape of the membership functions directly, these conditions are asymptotically exact only in cases where the membership functions may assume any combination of values in the standard unit simplex (as said in Kruszewski et al. 2009, Remark 1*).*

In (Campos, Tôrres, and Palhares 2012), asymptotically exact conditions that take into account the membership functions' shape, by means of simplicial partitions of the membership functions' image space, were proposed. In this chapter, these conditions are presented, as well as ways to extend them and different uses and interpretations for the simplicial partitions.

3.1 Asymptotically necessary conditions

As discussed in section 2.2, several TS analysis and synthesis conditions can be written as fuzzy summations. Even though the usual is to apply sufficient conditions for the summations (like the lemmas presented in section 2.2), many sufficient and asymptotically necessary conditions have been proposed in the literature. However, as explained above, these conditions are not always asymptotically necessary.

Despite this, these conditions may serve as motivation for the development of conditions that are asymptotically necessary in all situations. In this work's case, the conditions presented in (Kruszewski et al. 2009) served as inspiration.

The main idea of the conditions presented in (Kruszewski et al. 2009) is to partition the standard unit simplex, Δ, into n simplices in a way that

$$\Delta^{(m)}, m \in \{1, \dots, n\},$$
$$\Delta^{(1)} \cup \cdots \cup \Delta^{(n)} = \Delta.$$

An example of this partition is presented in Figure 3.1 for a system with three rules (note that, since the third rule is given by one minus the sum of the other two rules, its information is omitted in the plot).

Afterwards, the condition with the fuzzy summation is replaced by multiple conditions with fuzzy summations over each of the $\Delta^{(m)}$ simplices. Letting the $\Delta^{(m)}$ simplices volume tend to zero is equivalent to demanding that the fuzzy summation condition be valid for all points in the standard unit simplex, and, therefore, in the case where the membership functions assume values over all of Δ, leads to necessary conditions.

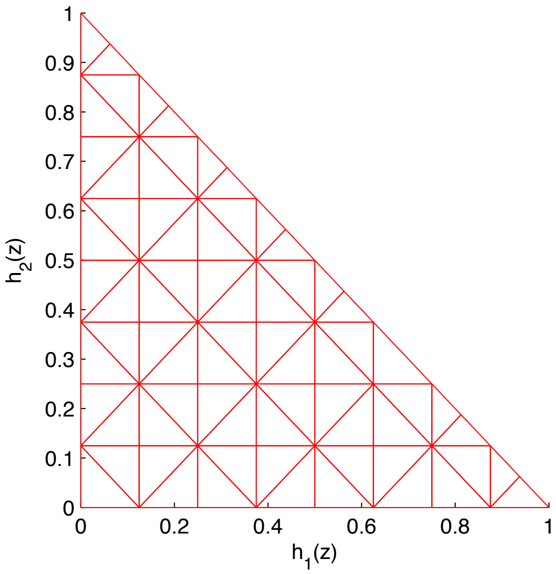

Figure 3.1: Example of a standard unit simplex partition of a 3 rule system using the methodology proposed in (Kruszewski et al. 2009).

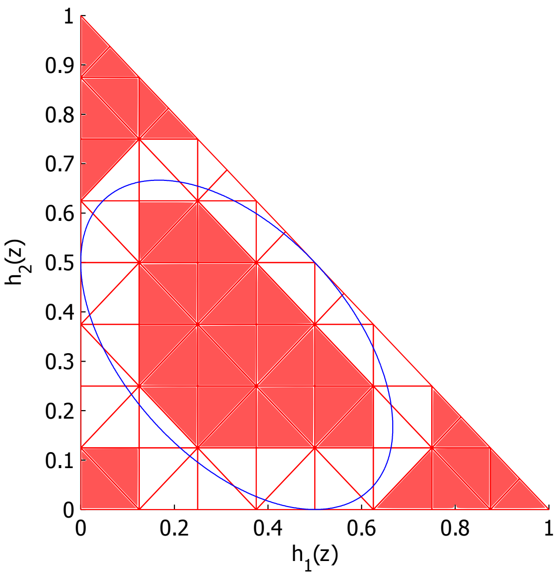

Figure 3.2: Example of the conservativeness generated by the methodology proposed in (Kruszewski et al. 2009).

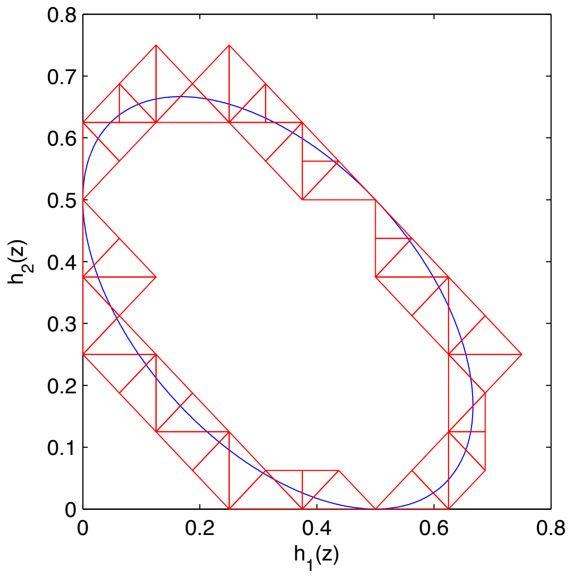

Figure 3.3: Example of a partition that satisfy the proposed condition.

Consider, however, the case presented in Figure 3.2 in which the membership functions only assume values over the curve presented. In this case, it is easy to realize that the methodology proposed in (Kruszewski et al. 2009) imposes unnecessary conditions (as can be seen by the filled simplices in Figure 3.2).

In that regard, the main idea presented in (Campos, Tôrres, and Palhares 2012) is to generate a partition of the standard simplex Δ composed of n simplices such that

$$\begin{aligned} &\Delta^{(m)}, m \in \{1, \dots, n\}, \\ &\mathbf{h}(\mathbf{v}) \subseteq \Delta^{(1)} \cup \cdots \cup \Delta^{(n)}. \end{aligned} \tag{3.1}$$

Which is equivalent to requiring that the n simplices contain all possible values of the membership functions for all possible values of the premises $\mathbf{v}$.

An example of this partition is presented in Figure 3.3.

Particularly, we desire a sequence of partitions such that

$$\lim_{n \to \infty} \Delta^{(1)} \cup \cdots \cup \Delta^{(n)} = \mathbf{h}(\mathbf{v}). \tag{3.2}$$

In addition, we consider that there exists a transformation of the points interior to simplex $\Delta^{(m)}$ to another space, in which simplex $\Delta^{(m)}$ is the standard unit simplex, and that can be written as (Kruszewski et al. 2009, equation (17))

$$\mathbf{h} = G^{(m)} \tilde{\mathbf{h}}^{(m)}, \tag{3.3}$$

in which $G^{(m)}$ are matrices formed by grouping the column vectors $\mathbf{g}_j^{(m)}$ that represent the vertices of simplex $\Delta^{(m)}$, $\mathbf{h}$ is a point in the original space, and $\tilde{\mathbf{h}}^{(m)}$ is a point in the transformed space.

Remark 3.1

It is interesting to note that a sufficient condition for the feasibility of expressions (2.6) and (2.7) (single and double fuzzy summations respectively) is that they are simultaneously feasible for all the points inside of all of the $\Delta^{(m)}$ simplices. If, in addition, when $n \to \infty$ the union of the simplices contain only points that belong in the membership functions' image space, this conditions also becomes asymptotically necessary.

In that regard, making use of lemmas 2.5 and 2.6, we can enunciate the following theorems.

Theorem 3.1: Asymptotically necessary condition for a single fuzzy summation

Given a partition of the standard simplex Δ, composed of n simplices, according to equation (3.1), sufficient *conditions for* (2.6) *are that conditions*

$$\forall m \in \{1, \ldots, n\}, \begin{cases} \sum_{j=1}^{r} \tilde{h}_j^{(m)} \tilde{Q}_j^{(m)} > 0, \\ \tilde{Q}_j^{(m)} = \sum_{i=1}^{r} g_{ij}^{(m)} Q_i, \end{cases}$$

be feasible for every point of the simplicial partition. In which $g_{ij}^{(m)}$ is the ith element of vertex $\mathbf{g}_j^{(m)}$ from simplex $\Delta^{(m)}$. In the limit case, if the sequence of partitions is such that condition (3.2) *is satisfied, then these conditions are also* necessary.

Theorem 3.2: Asymptotically necessary condition for a double fuzzy summation

Given a partition of the standard simplex Δ, composed of n simplices, according to equation (3.1), sufficient *conditions for* (2.7) *are that conditions*

$$\forall m \in \{1, \ldots, n\}, \begin{cases} \sum_{k=1}^{r} \sum_{l=1}^{r} \tilde{h}_k^{(m)} \tilde{h}_l^{(m)} \tilde{Q}_{kl}^{(m)} > 0, \\ \tilde{Q}_{kl}^{(m)} = \sum_{i=1}^{r} \sum_{j=1}^{r} g_{ik}^{(m)} g_{jl}^{(m)} Q_{ij}, \end{cases} \tag{3.4}$$

be feasible for every point of the simplicial partition. In which $g_{ij}^{(m)}$ is the ith element of vertex $\mathbf{g}_j^{(m)}$ from simplex $\Delta^{(m)}$. In the limit case, if the sequence of partitions is such that condition (3.2) *is satisfied, then these conditions are also* necessary.

Remark 3.2

In some cases, conditions may occur with a higher number of fuzzy summations. Despite not being directly dealt with in this chapter, the generalization for such cases is obtained by applying the generalization of lemma 2.6 presented in (Bernal, Guerra, and Kruszewski 2009, *Remark 2*).

Unlike other asymptotically necessary conditions in the literature, the conditions above allow for the relaxation of the single fuzzy summation (Theorem 3.1) by making use of the membership functions' shape. Something that cannot be done in case we consider that the membership functions may assume any combination of values. In this case, the use of Theorem 3.1 is limited to verifying

$$\tilde{Q}_j^{(m)} > 0, \quad \forall m, j.$$

The use of Theorem 3.2, on the other hand, allows the use of any sufficient condition for double fuzzy summations (like lemmas 2.1, 2.2 and 2.3 presented in section 2.2) in order to guarantee that (3.4) is feasible. However, when the $\Delta^{(m)}$ simplices volume goes to zero, the difference between these conditions will also go to zero, and the choice of any sufficient condition will generate a sequence of asymptotically necessary conditions.

Remark 3.3

In the work of (Kruszewski et al. 2009), a sequence of sufficient and asymptotically necessary conditions and a sequence of necessary and asymptotically sufficient conditions for the double fuzzy summation are presented (the sequence of necessary and asymptotically sufficient conditions is found by using only the $\tilde{Q}_{kk} > 0$ conditions for the double summation). However, the sequence of necessary conditions proposed in (Kruszewski et al. 2009) is only necessary in cases where the membership functions may assume any combination of values over the standard unit simplex.

Remark 3.4

It is worth noting that Theorems 3.1 and 3.2 present asymptotically necessary conditions for (2.6) and (2.7), but that may not be necessary for conditions of the form

$$\mathbf{x}^T \left(\sum_{i=1}^{r} h_i Q_i \right) \mathbf{x} > 0,$$

and

$$\mathbf{x}^T \left(\sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j Q_{ij} \right) \mathbf{x} > 0,$$

if any of the premise variables is a state variable (*i.e.* the parametric LMIs may not be necessary conditions for the quadratic form, since they do not take into account any relationship the variables from the quadratic form may have with the premise variables).

Partition Strategies - Part I

Theorems 3.1 and 3.2 present sufficient conditions for any partition generating strategy, as long as the partitions satisfy condition (3.1). If, in the limit case, condition (3.2) is also satisfied, then the conditions on both theorems are also asymptotically necessary.

Note then, that the biggest problem of this approach is generating partitions in a way that, at each step, a more refined mesh of simplices is generated around the possible values of the membership functions, while simplices that cover invalid areas are discarded.

In order for the partition strategies to be implemented, the membership functions are approximated by piecewise constant functions, given by the zero order interpolation of samples over the domain of the premise variables (that can be represented as a point cloud in the membership functions' image space).

In this first part, we consider the approach presented in (Campos, Tôrres, and Palhares 2012), and disregard the approximation error between the piecewise constant functions and the membership functions. In the following parts, we propose extensions of the partitions methods presented here to deal with this approximation error.

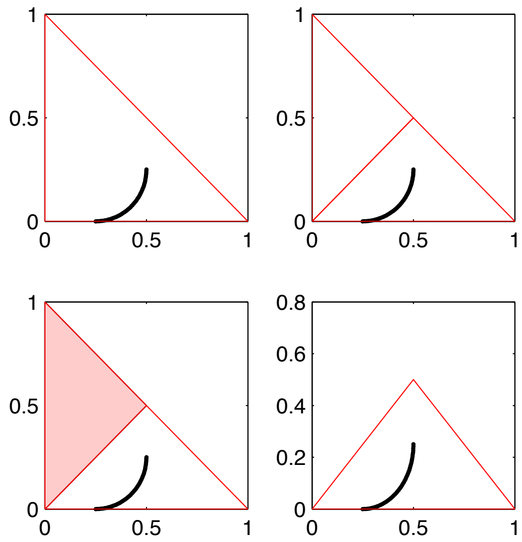

Figure 3.4: Steps from strategy *Dividing and Discarding*. Find the greatest volume simplex and split it. For each new simplex, check whether or not it contains at least one point. If it does, keep it in the partition. If it does not, discard it.

Remark 3.5

As presented in section 2.3, by making use of the Mean Value Theorem and assuming that the functions to be approximated are continuously differentiable, it is easy to show that this approximation error is limited and goes to zero when the sampling step goes to zero. In that regard, given a very dense sampling, it is reasonable to disregard the approximation error and make use of the partition strategies proposed in this part.

Dividing and Discarding

The first partition scheme, named *Dividing and Discarding*, can be seen as an adaptation of the first partition algorithm presented in (Kruszewski et al. 2009).

At each step, we search for the largest edge from the simplex with the greatest volume and add a new vertex in its middle. By doing so, the simplex with the greatest volume is split into new simplices.

For each of the new simplices, we check if they contain at least one of the points from the point cloud of the piecewise constant functions (representing the membership functions). If at least one point is in the interior of the simplex, we preserve it. Otherwise, it is discarded from the partition.

Figure 3.4 illustrates one step of this partition strategy.

In order to check if a point belongs to the interior of a simplex, note that, from equation (3.3)

$$\tilde{\mathbf{h}}^{(m)} = \left[G^{(m)}\right]^{-1} \mathbf{h}. \tag{3.5}$$

In addition, since $G^{(m)}$ defines a linear transformation in which simplex $\Delta^{(m)}$ is the standard unit simplex (in the transformed space), only its interior points will belong to the standard simplex in the transformed space.

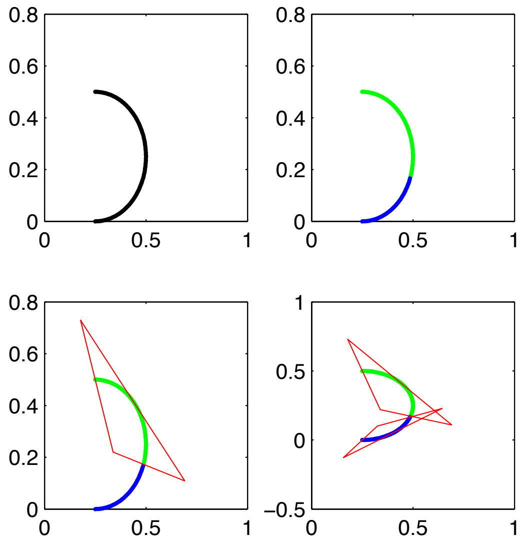

Figure 3.5: Steps from strategy *Grouping the membership points*. Find the largest grouping (the one with the greatest dispersion) and divide it into two new groupings. For each one of them, find a simplex that covers all of its points.

Therefore, a point, $\mathbf{h}$, belongs to the interior of a simplex $\Delta^{(m)}$ if and only if the transfomed point , $\tilde{\mathbf{h}}^{(m)}$, satisfy

$$\tilde{h}_i^{(m)} \geq 0, \quad \forall i$$
$$\sum_{i=1}^{r} \tilde{h}_i^{(m)} = 1.$$

Grouping the membership points

The second partition strategy, named *Grouping the membership points*, aims to solve the problem from a different angle. Instead of generating an increasingly finer mesh of simplices by their subdivision, we work with groupings of points in the image space.

At each iteration, choose a grouping (preferably the one with the greatest dispersion of its points) and divide it into two new groupings. For each of these groupings, find a simplex that covers all of its points.

Figure 3.5 illustrates one step of this partition strategy.

The regrouping step (the step in which a grouping is split into two new groupings) can be realized by making use of any clustering algorithm, like the *k-means* (MacQueen 1967). In this work, we made use of the MATLAB function *kmeans*, that in addition to providing a way to divide the group of points, also gives a measure of the size of the new groups.

Different ways of finding a simplex covering a set of points are commonly presented in the tensor product model transformation literature (Baranyi 2004). These transformations, similarly to equation (3.3), can be seen as different ways of finding simplices with desired properties that cover a set of points. The CNO and RNO-INO transformations, for instance, find, respectively, the smallest volume simplex and a simplex whose every face is touched.

Remark 3.6

Note that the transformations proposed in the tensor product model transformation literature are usually presented as a transposed version of equation (3.5). Therefore, the $G^{(m)}$ matrix, containing the vertices of the simplex that covers the points can be found transposing the inverse of the transformation matrix.

Remark 3.7

The default behaviour of the *k-means* implementation in MATLAB is to use random starting positions for the cluster's centroids. Therefore, different realizations of the same problem may lead to different partitions. However, once the number of steps from the partition increases, the difference between the realizations should decrease.

Partition Strategies - Part II

The partition strategies proposed above disregard the approximation error, and consider that the membership function can be perfectly represented as a point cloud in the image space. However, by making use of the content presented in section 2.3, we may consider an uncertain representation for the piecewise constant functions, using the error polytopes and balls as an added uncertainty that guarantees that this representation covers the original membership functions, and extend the strategies proposed to take the approximation error into consideration.

Note that, even then, we still have sufficient and asymptotically necessary conditions. However, in this new format, it is necessary for the number of simplices and the number of samples to go to infinity (so that the simplices volume and the approximation error go to zero).

Figure 3.6 illustrates the approximation of the membership functions by piecewise constant functions, and different possible representations for the approximation error (two different error polytope schemes and an error ball).

Remark 3.8

Note that, since we are dealing with membership functions, the sum of the errors must always be equal to zero in order for us to only consider points that belong to the same hyperplane as the membership functions (the hyperplane of sum equal to one). Therefore we must impose that the sum of the coordinates of the polytope vertices is equal to zero (which is equivalent to taking the intersection of the error polytope $\Delta\mathbf{h}(\mathbf{x})$ with the hyperplane $\sum_{i=1}^{r} \Delta h_i = 0$).

Another approach, which is a little more conservative, is the following: consider a system with r rules. Given the values of $r-1$ membership functions, the value of the other function is always exactly known (one less the sum of the other membership functions). Therefore we can take the maximum and minimum values of only $r-1$ approximation errors and complete the remaining dimension with the value that makes the sum of the coordinates equal to zero. In that regard, we reduce the number of vertices of the uncertainty polytope from 2^r to 2^{r-1}.

In the case presented in Figure 3.6 (c), we made use of only the bounds on the first and second membership function, and the values of the third are calculated as those that make the sum equal to zero, simplifying an 8 vertices polytope (or 6 vertices after the intersection with the zero sum plane, as presented in Figure 3.6 (b)) to 4 vertices.

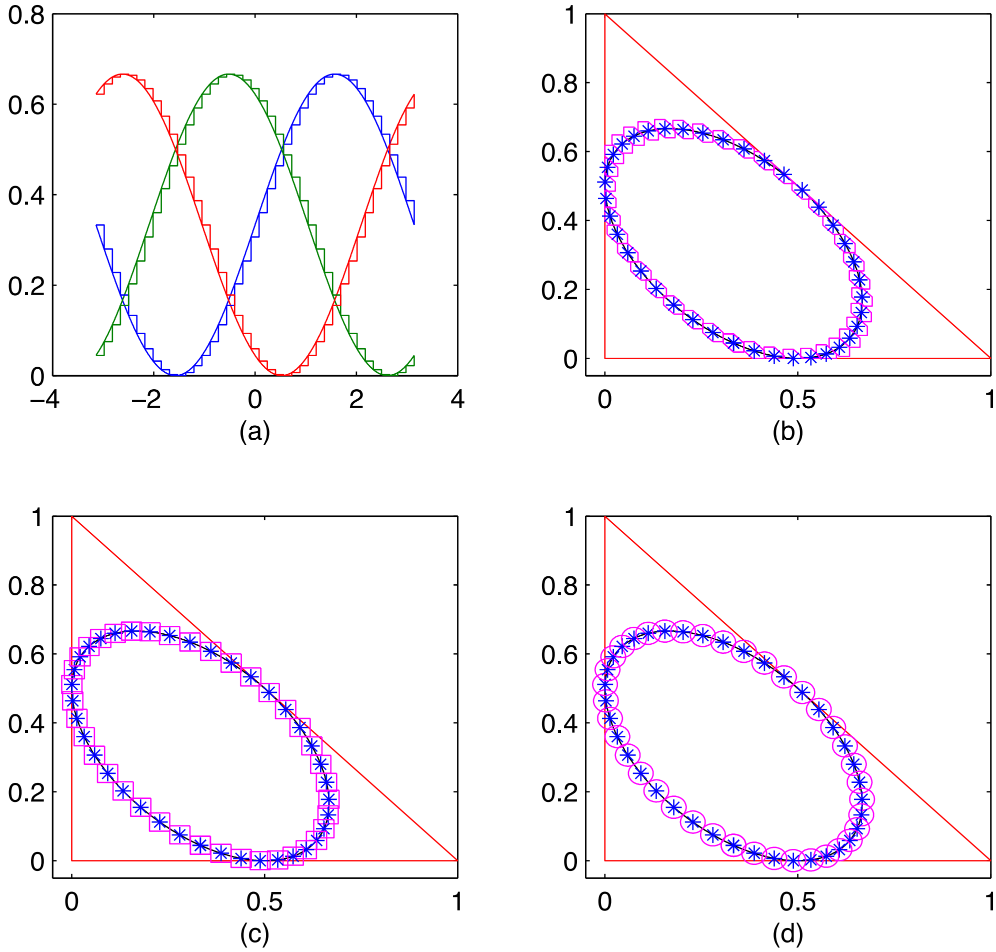

Figure 3.6: Approximation of the membership functions by piecewise constant functions. (a) presents the approximation by piecewise constant functions. (b) presents the corresponding point cloud in the image space using the error polytope obtained by the intersection of the error polytope with the zero sum plane and the standard simplex edges. (c) presents the error polytope described in Remark 3.8. (d) presents the error ball obtained using the Mean Value Theorem generalization for vector functions.

Dividing and discarding

In the first strategy proposed, we can make use of the error ball representation. In that regard, we substitute the tests of whether a point belongs to a simplex by tests of whether the simplex touches any uncertainty ball.

This test can be carried out in the following way: find the interior point of the simplex that is closest to the ball center (which is the point in the image space that was considered in the approach present in the previous subsection) and check if its distance to the center is less than or equal to the uncertainty radius. If it is, then the simplex contains points that belong to the uncertainty ball and cannot be discarded. Otherwise, test it with the other samples and uncertainty balls. If the test is negative for all of the samples, the simplex is discarded.

In order to find the simplex's closest interior point to the sample, note that a simplex is a convex polygon and that every point in its interior can be written as

$$\overline{\mathbf{v}} = \sum_{i=1}^{r} \alpha_i \mathbf{v}_i,$$
$$\alpha_i \geq 0, \quad \sum_{i=1}^{r} \alpha_i = 1$$

Given an external point $\mathbf{p}$, the interior point of a simplex closest to this point can be found by solving the optimization problem

$$\min_{\alpha_i} \left[(\sum_{i=1}^{r} \alpha_i \mathbf{v}_i) - \mathbf{p} \right]^T \left[(\sum_{i=1}^{r} \alpha_i \mathbf{v}_i) - \mathbf{p} \right]$$
$$\text{s.t.} \quad \alpha_i \geq 0, \quad \sum_{i=1}^{r} \alpha_i = 1$$

Note that the objective function can be rewritten as

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_r & -1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T\mathbf{v}_1 & \mathbf{v}_1^T\mathbf{v}_2 & \dots & \mathbf{v}_1^T\mathbf{v}_r & \mathbf{v}_1^T\mathbf{p} \\ * & \mathbf{v}_2^T\mathbf{v}_2 & \dots & \mathbf{v}_2^T\mathbf{v}_r & \mathbf{v}_2^T\mathbf{p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & \dots & \mathbf{v}_r^T\mathbf{v}_r & \mathbf{v}_r^T\mathbf{p} \\ * & * & \dots & * & \mathbf{p}^T\mathbf{p} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_r \\ -1 \end{bmatrix}$$

Which allows us to rewrite the optimization problem as a quadratic optimization problem subject to linear equality and inequality constraints

$$\min_{\alpha_i, \rho} f_0 = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_r & \rho \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T\mathbf{v}_1 & \mathbf{v}_1^T\mathbf{v}_2 & \dots & \mathbf{v}_1^T\mathbf{v}_r & \mathbf{v}_1^T\mathbf{p} \\ * & \mathbf{v}_2^T\mathbf{v}_2 & \dots & \mathbf{v}_2^T\mathbf{v}_r & \mathbf{v}_2^T\mathbf{p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & \dots & \mathbf{v}_r^T\mathbf{v}_r & \mathbf{v}_r^T\mathbf{p} \\ * & * & \dots & * & \mathbf{p}^T\mathbf{p} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_r \\ \rho \end{bmatrix}$$
$$\text{s.t.}$$
$$f_i \leq 0, \quad g_0 = 0, \quad g_1 = 0$$
$$f_i = -\alpha_i, \quad g_0 = (\sum_{i=1}^{r} \alpha_i) - 1, \quad g_1 = \rho + 1$$

Or, in vector form

$$\min_{\mathbf{x}} f_0 = \mathbf{x}^T A \mathbf{x}$$
$$\text{s.t.}$$
$$\mathbf{f} = H\mathbf{x} \leq 0$$
$$\mathbf{g} = G\mathbf{x} - \mathbf{b} = 0$$

with

$$A = \begin{bmatrix} \mathbf{v}_1^T\mathbf{v}_1 & \mathbf{v}_1^T\mathbf{v}_2 & \dots & \mathbf{v}_1^T\mathbf{v}_r & \mathbf{v}_1^T\mathbf{p} \\ * & \mathbf{v}_2^T\mathbf{v}_2 & \dots & \mathbf{v}_2^T\mathbf{v}_r & \mathbf{v}_2^T\mathbf{p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & \dots & \mathbf{v}_r^T\mathbf{v}_r & \mathbf{v}_r^T\mathbf{p} \\ * & * & \dots & * & \mathbf{p}^T\mathbf{p} \end{bmatrix},$$

$$H = \begin{bmatrix} -1 & 0 & \dots & 0 & 0 \\ 0 & -1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & \dots & -1 & 0 \end{bmatrix},$$

$$G = \begin{bmatrix} 1 & 1 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

The objective function's gradient is given by

$$\frac{\partial f_0}{\partial \mathbf{x}} = 2A\mathbf{x}$$

and its Hessian by

$$\frac{\partial^2 f_0}{\partial \mathbf{x}^2} = 2A$$

We can then make use of any method available to solving quadratic optimization problems (which are not usually that computationally expensive to solve), like the *quadprog* command from MATLAB or even a primal-dual interior point method (Boyd and Vandenberghe 2004, cap. 11.7) (which, in this problem, was implemented specifically for this problem).

Grouping the membership points

In the second strategy proposed, it suffices to do a simple modification: the clustering algorithm in the regrouping step still considers only the point cloud in the image space. However, the step of finding a simplex that cover the grouping's points uses the vertices of the uncertainty polytopes of each sample from the grouping. As a result of that, we find a simplex that covers the vertices of the polytopes and, thus, also covers the points in the interior of the polytope (since a simplex is convex).

Discussing the case with several premise variables

Since in this work we consider TS models with a tensor product structure in its membership functions (*i.e.* models whose membership functions satisfy equation (2.2)), two different approaches are possible for the partition strategies of the image space:

1. Separately dividing the simplices of the image spaces of $\boldsymbol{\mu}_i(v_i)$ and compounding the equivalent simplices in the image space of $\mathbf{h}(\mathbf{v})$. As can be inferred from the results presented in section 2.5, this composition is given by the tensor product of the vertices of the simplices for all possible combinations.

   The problem with this approach is that the number of simplices grows too fast and in neither of the two partition strategies proposed we have a clear way of choosing what image space is going to be split next.

   The advantage of this approach is that we can use a reduced number of samples.

2. Directly dividing the simplices in the image space of $\mathbf{h}(\mathbf{v})$.

   The problem of this approach is that, in this case, we will need a considerably higher number of samples (for the same approximation error as in the other approach), since the samples will be given by the tensor product of the samples of every function (in order for us to be able to find the error polytope, or error ball, according to the contents presented in section 2.3).

   The advantage of this approach is that we get more control over the growth of the number of simplices and the partition strategies presented can be directly applied (as long as one knows how to calculate the approximation error).

3.2 Switched representation

A different point of view

Despite the partition strategies presented having been conceived in order to generate a set of constraints in order to decrease the conservativeness of constraints commonly found in the literature (and reaching equivalency in the limit case), they can also be seen from another interesting point of view: *a way of finding exact switched nonlinear representations for a nonlinear system*.

In order to illustrate, consider the TS system defined by equation (2.4):

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{v})\left(A_i\mathbf{x} + B_i\mathbf{u}\right).$$

In each $\Delta^{(m)}$ simplex, we will have a description given by:

$$\begin{aligned}
\dot{\mathbf{x}} &= \sum_{i=1}^{r} \tilde{h}_i^{(m)} \left[A_i^{(m)}\mathbf{x} + B_i^{(m)}\mathbf{u}\right] \\
\tilde{\mathbf{h}}^{(m)} &= \left[G^{(m)}\right]^{-1}\mathbf{h} \\
A_j^{(m)} &= \sum_{i=1}^{r} g_{ij}A_i \\
B_j^{(m)} &= \sum_{i=1}^{r} g_{ij}B_i
\end{aligned}$$

in which $\mathbf{h}$ is the vector of the original membership functions, $\tilde{\mathbf{h}}^{(m)}$ is the vector of transformed membership functions, and $G^{(m)}$ defines a linear transformation in which simplex $\Delta^{(m)}$ is the standard unit simplex.

In addition, we have a switching signal, σ, given by

$$\sigma = m, \text{ if } \mathbf{h} \in \Delta^{(m)}.$$

In cases where there exists a conflict ($\mathbf{h}$ belongs to more than one simplex), the switching signal is free to be chosen as any of the conflicting simplices. In this work case, in order to resolve the conflicts, we choose the switching signal according to the active simplex whose centroid is closer, in terms of the 2-norm, to the current value of vector $\mathbf{h}$.

Switching control according to the partitions

The idea of using a switched control law according to partitions on the image space of the membership functions is not a novelty of this work. This type of control law was proposed in (Ko, Lee, and Park

2012), however the authors do not present any method in order to partition the image space of the membership functions.

In this work, we propose the use of a control law switched according to the partitions. However, since the idea of the partition strategies is having simplicial meshes whose simplices volume (in the image space) is ever smaller, the vertex systems are ever closer to each other (which by itself asymptotically implies in the partitions having a linear behaviour) and we choose to use a linear control law for each of the partitions.

This control law can be written as

$$\mathbf{u} = K^{(m)}\mathbf{x}, \text{ if } \sigma = m.$$

A different proposal of this work is the use of the tensor product model transformation (presented in section 2.4) to recover a continuous controller (simpler to implement, since ideally we would like to have a large number of partitions).

To that end, however, we must impose, on the synthesis conditions, that the controller of region (m) also stabilizes the neighboring regions. With those extra conditions, we can prove that the solutions of any switching between the controllers of neighboring regions stabilizes the system (because we will be proving the stability of the Filipov solutions). Therefore, by making use of (Liberzon 2003, corollary 2.3, page 25) or even the convexity property of the LMIs, we can show that any convex combination of the controllers will stabilize the system.

Remark 3.9

In this work, two regions are called neighbors if their simplices touch or intercept. Numerically, this can be verified by checking if at least one of the vertices of one of the simplices is contained in the other one.

In particular, we may take several samples from the controller gain (walking through the image space of the membership functions) and the controller given by the linear interpolation of these gains will stabilize the system (given that we draw at least one sample from each partition). These samples can then be used in the tensor product model transformation and we can recover the TS representation of the controller. It is interesting to note that, instead of generating a different set of samples from the membership functions' image space, the samples used to generate the partitions can be reused (guaranteeing that we draw at least one sample from each partition).

However, note that when using a large number of partitions (which is the idea when we are trying to achieve the necessity of the conditions), the difference between neighboring regions becomes ever smaller and the chance of one partition's controller also stabilizing its neighboring regions is high. Therefore, a possible synthesis methodology is to find the controller without adding the extra constraints to the optimization problems, and checking afterwards if they are valid for the controller values found. If they are, the procedure described above can be done without any problems. Otherwise, we try to solve the optimization problem a second time, but this time adding the extra conditions. In this way, the optimization problem (which is the longest step of the methodology) can be reduced.

Remark 3.10

By making use of this methodology to reconstruct a continuous controller, we find a fuzzy control law for the system. In doing so, we can think that this method allow us to search, at the same time, for the controller gains and its membership functions (unlike PDC control laws usually employed in the literature that employ the same membership functions as the system).

3.3 Examples

In this section, two possible use cases for the conditions in this chapter are presented. We consider, first, the problem of stabilizing a TS system and, in the following, the problem of finding a controller that minimizes an upper bound of the system's $\mathcal{H}_\infty$ norm.

In both cases we'll make use of a quadratic Lyapunov function, and the conditions used are largely known in the literature (or have some minor modifications)[1]. In that regard we do not present these conditions derivation (as that is not the focus of this section).

Stabilization

Consider a TS system defined by equation (2.4):

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{v}) A_i \mathbf{x} + B_i \mathbf{u}.$$

By making use of a linear state feedback control law, $\mathbf{u} = K\mathbf{x}$, we may look for a stabilizing controller by means of the conditions

$$\begin{aligned} \sum_{i=1}^{r} h_i Q_i &< 0, \\ P &> 0, \\ Q_i = A_i P + P A_i^T + B_i X + X^T B_i^T&. \end{aligned}$$

And the controller gains can be recovered by $K = XP^{-1}$.

By making use of a PDC state feedback control law, $\mathbf{u} = \sum_{j=1}^{r} h_j K_j \mathbf{x}$, we may look for a stabilizing controller by means of the conditions

$$\begin{aligned} \sum_{i=1}^{r}\sum_{j=1}^{r} h_i h_j Q_{ij} &< 0, \\ P &> 0, \\ Q_{ij} = A_i P + P A_i^T + B_i X_j + X_j^T B_i^T&. \end{aligned}$$

And the controller gains can be recovered by $K_j = X_j P^{-1}$.

Both of the above presented conditions can be solved by making use of the techniques presented in this chapter. Taking advantage of the partitions generated by these techniques, we may define a switched control law (as presented in the previous section). Such a control law can be written as

$$\mathbf{u} = K^{(m)}\mathbf{x}, \text{ if } \sigma = m.$$

In that regard, we may look for a stabilizing controller by means of the conditions

$$\begin{aligned} \sum_{j=1}^{r} \tilde{h}_j^{(m)} \sum_{i=1}^{r} g_{ij}^{(m)} Q_i^{(m)} &< 0, \forall m \\ P &> 0, \\ Q_i^{(m)} = A_i P + P A_i^T + B_i X^{(m)} + (X^{(m)})^T B_i^T&. \end{aligned}$$

[1]To the interested reader, the stabilization conditions are presented in (Tanaka and Wang 2001, Section 3.3) and (Tuan et al. 2001, Theorem 3.2). Some slightly different conditions for the $\mathcal{H}_\infty$ control are presented in (Tuan et al. 2001, Theorem 5.1)

And the controller gains will be given by $K^{(m)} = X^{(m)}P^{-1}$.

In addition, as explained in the previous section, we may look for a switched controller that can be represented in a continuous fashion (as discussed in the previous section) by imposing that the switched gains are also stabilizing for the neighbouring regions. Therefore, the conditions presented can be modified to

$$\sum_{j=1}^{r} \tilde{h}_j^{(m)} \sum_{i=1}^{r} g_{ij}^{(m)} Q_i^{(n)} < 0, \forall m, n = m \text{ or } n \text{ is a neighbour of } m$$
$$P > 0,$$
$$Q_i^{(n)} = A_i P + P A_i^T + B_i X^{(n)} + (X^{(n)})^T B_i^T.$$

In which the controller gains are given by $K^{(m)} = X^{(m)}P^{-1}$ and a continuous control law can be recoverd by making use of the Tensor Product Model Transformation.

Remark 3.11

Note that, the way the above conditions were defined, the quadratic Lyapunov function found is of the form

$$V(\mathbf{x}) = \mathbf{x}^T P^{-1} \mathbf{x}.$$

Example 3.1. Consider the TS model with a free b parameter presented in (Campos, Tôrres, and Palhares 2012):

$$\dot{\mathbf{x}} = \sum_{i=1}^{3} h_i(x_1)\left[A_i \mathbf{x} + B_i \mathbf{u}\right],$$

with

$$A_1 = \begin{bmatrix} 1.59 & -7.29 \\ 0.01 & 0 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$
$$A_2 = \begin{bmatrix} 0.02 & -4.64 \\ 0.35 & 0.21 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 8 \\ 0 \end{bmatrix},$$
$$A_3 = \begin{bmatrix} -2 & -4.33 \\ 0 & 0.05 \end{bmatrix}, \qquad B_3 = \begin{bmatrix} -b+6 \\ -1 \end{bmatrix},$$

and

$$h_1(x_1) = \frac{\sin(x_1)+1}{3},$$
$$h_2(x_1) = \frac{\sin(x_1+2\pi/3)+1}{3},$$
$$h_3(x_1) = \frac{\sin(x_1-2\pi/3)+1}{3}.$$

We would like to find the largest value of b for which we are able to find a quadratically stabilizing control law, and to that end, we will make use of the conditions presented above as well as the partition strategies presented in this chapter (taking the approximation error into account) and other similar approaches available in the literature. Particularly, we will compare the results from this chapter with (Sala and Ariño 2007, Proposition 2), (Sala and Ariño 2007, Theorem 5), (Kruszewski et al. 2009, Algorithm 1), all of these conditions as well as this chapter conditions augmented with the shape relaxation from (Sala and Ariño 2008, Proposition 3.1), and the conditions from (Lam and Narimani 2010, Theorem 2).

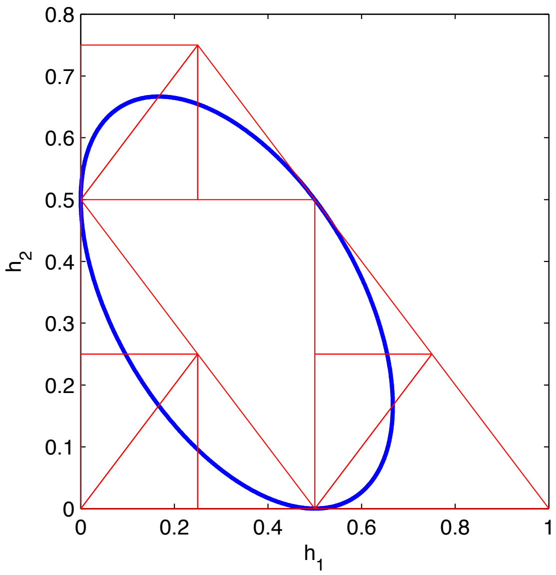

Figure 3.7: Partition generated by the *Divinding and Discarding* strategy after 12 steps.

In order to take the approximation error into consideration, note that the membership functions vector $\mathbf{h} = \begin{bmatrix} \frac{\sin(x_1)+1}{3} & \frac{\sin(x_1+2\pi/3)+1}{3} & \frac{\sin(x_1-2\pi/3)+1}{3} \end{bmatrix}^T$ derivative regarding the premise variable x_1 is given by

$$\frac{\partial}{\partial x_1}\mathbf{h} = \begin{bmatrix} \frac{1}{3}\cos(x_1) \\ \frac{1}{3}\cos(x_1+2\pi/3) \\ \frac{1}{3}\cos(x_1-2\pi/3) \end{bmatrix}.$$

Whose elements all have the same maximum, equal to $1/3$, and the same minimum, equal to $-1/3$. In addition, its vector 2-norm is equal to $\sqrt{6}/6$.

Note that the membership functions are periodic and share the same period, 2π. Thus, it suffices to take samples from the premise variable in the interval $[-\pi, \pi]$ in order to cover all of the membership functions' image space (this same observation is valid for the conditions from Lam and Narimani 2010, Theorem 2).

By taking 1001 samples, we have a sampling step equal to $\pi/500$, which leads to an uncertainty radius of $\frac{\pi\sqrt{6}}{6000}$ and a possible uncertainty polytope equal to $\begin{bmatrix} \frac{\pi}{3000} & \frac{\pi}{3000} & -\frac{\pi}{1500} \\ \frac{\pi}{3000} & -\frac{\pi}{3000} & 0 \\ -\frac{\pi}{3000} & \frac{\pi}{3000} & 0 \\ -\frac{\pi}{3000} & -\frac{\pi}{3000} & \frac{\pi}{1500} \end{bmatrix}$, in which every row is a polytope vertex.

Figures 3.7 and 3.8 illustrate the partition strategies after 12 iterations.

In order to make use of the shape relaxation from (Sala and Ariño 2008, Proposition 3.1), note that the membership functions are such that

$$\mathbf{h}^T S\mathbf{h} + \mathbf{w}^T\mathbf{h} + v \leq 0,$$

with $S = 6I$, $\mathbf{w} = \begin{bmatrix} -4 & -4 & -4 \end{bmatrix}^T$, and $v = 1$. It is interesting to note that when applied to single fuzzy summation conditions, this relaxation transforms it into double fuzzy summation conditions, and thus allows the use of sum relaxations that are not available for the single summation case. It is also interesting to note that this shape inequality is actually an equality in this example (meaning that all of the membership functions values lie over the curve given by $\mathbf{h}^T S\mathbf{h} + \mathbf{w}^T\mathbf{h} + v = 0$).

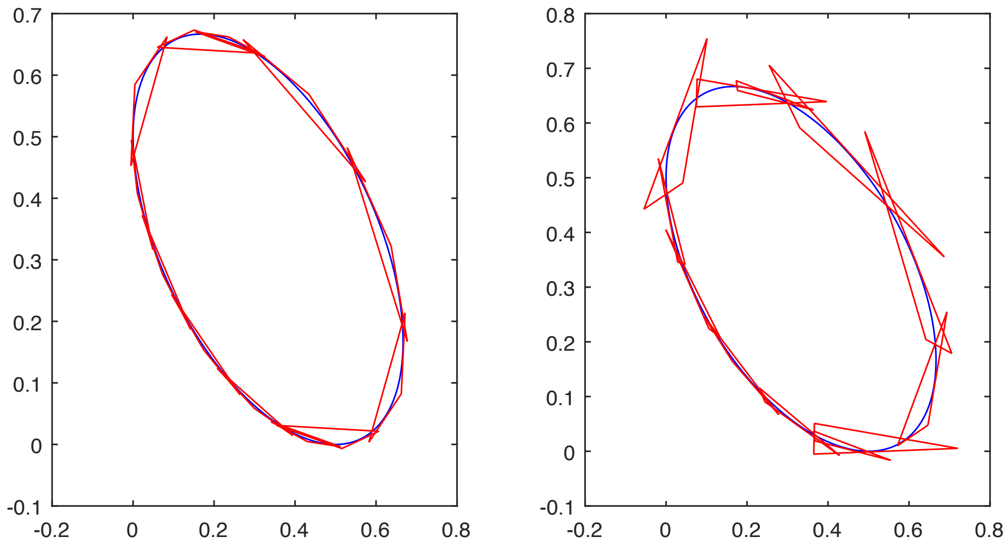

Figure 3.8: Partition generated by the *Grouping the membership points* strategy after 12 steps on Example 3.1. The left partition was generated using the **CNO** transformation, whereas the one on the right was generated by the **RNO-INO** transformation.

In addition, the Γ matrix representing the γ_{ij} elements used in the inequalities $h_i h_j - \bar{h}_i \bar{h}_j \geq \gamma_{ij}$, with $\bar{h}_i$ the staircase approximation of h_i used in the conditions in (Lam and Narimani 2010, Theorem 2), can be described by

$$\Gamma = \frac{-2\pi}{n_p} \begin{bmatrix} \frac{1.3}{9} & \frac{1.77}{18} & \frac{1.77}{18} \\ \frac{1.77}{18} & \frac{1.3}{9} & \frac{1.77}{18} \\ \frac{1.77}{18} & \frac{1.77}{18} & \frac{1.3}{9} \end{bmatrix},$$

with n_p the number of points used in the staircase approximation.

Tables 3.1, 3.2, 3.3 present the minimum number of rows of LMIs for each of the methods to find a stabilizing PDC, linear and switched controller for a given value of b. Figures 3.9 to 3.17 present these same comparisons, but showing the largest feasible value of b for a given number of rows of LMIs. Note that, due to the random initialization of the *k-means*, the values presented for the *Grouping the membership points* partition strategy may change. Finally, Table 3.4 presents the complexity analysis of each of the methods used for this example (by means of comparison we also present the complexity analysis for the usual conditions for a linear control law and a PDC control law).

Note that conditions (Sala and Ariño 2007, Proposition 2), (Sala and Ariño 2007, Theorem 5) and (Kruszewski et al. 2009, Algorithm 1) are *shape-independent*, and when used without the relaxation from (Sala and Ariño 2008, Proposition 3.1) are intrinsically more conservative than the other conditions (since they use less information). We can find an upper bound for *shape-independent* conditions by using the necessary and asymptotically sufficient conditions from (Kruszewski et al. 2009). In this example the upper bound found b was 6.661.

These same necessary and asymptotically sufficient conditions could be used in conjunction with (Sala and Ariño 2008, Proposition 3.1) to find an upper bound for the *shape-dependent* case. However, we were not able to find a value with the number of partitions we tried in this example. Interestingly though, we found that using Lemma 2.3 together with (Sala and Ariño 2008, Proposition 3.1) (which is the same as using Sala and Ariño 2007, Theorem 5 with 2 summations and Sala and Ariño 2008, Proposition 3.1) found the largest feasible value of b, which was the same for the PDC, the linear and

Table 3.1: Comparison of different methods of sum relaxation for a PDC control law in example 3.1 (minimum number of rows of LMIs required to find a stabilizing control law)

Method/b	4	5	5.5	6	6.3	6.6	6.7	6.72	6.75
Sala and Ariño 2007, Proposition 2	158	464	994	3194	12212	-	-	-	-
Sala and Ariño 2007, Theorem 5	34	34	34	34	34	60	-	-	-
Kruszewski et al. 2009, Algorithm 1	74	386	722	1538	-	-	-	-	-
Kruszewski et al. 2009, Algorithm 1 with Lemma 2.3	26	50	50	50	74	122	-	-	-
Sala and Ariño 2007, Proposition 2 with Sala and Ariño 2008, Proposition 3.1	24	24	34	46	94	760	3910	-	-
Sala and Ariño 2007, Theorem 5 with Sala and Ariño 2008, Proposition 3.1	32	32	32	32	32	32	32	32	32
Kruszewski et al. 2009, Algorithm 1 with Sala and Ariño 2008, Proposition 3.1	28	28	28	40	52	52	1720	1720	1720
Dividing and Discarding	98	170	170	170	170	170	-	-	-
Dividing and Discarding with Lemma 2.3	38	74	74	74	146	254	-	-	-
Dividing and Discarding with Sala and Ariño 2008, Proposition 3.1	28	28	28	40	52	88	628	760	760
Grouping the membership points-**CNO**	26	26	50	50	50	170	194	350	-
Grouping the membership points-**CNO** with Lemma 2.3	38	38	74	74	74	254	290	380	-
Grouping the membership points-**CNO** with Sala and Ariño 2008, Proposition 3.1	28	28	28	40	52	64	76	100	184
Grouping the membership points-**RNO-INO**	62	98	98	110	194	362	-	-	-
Grouping the membership points-**RNO-INO** with Lemma 2.3	74	92	146	164	290	542	-	-	-
Grouping the membership points-**RNO-INO** with Sala and Ariño 2008, Proposition 3.1	28	52	64	76	112	184	376	376	556
Lam and Narimani 2010, Theorem 2	54	74	94	134	194	494	1114	1474	2934

the switched control law, equal to 6.7803 (and that's the reason why the tables do not show any method using both Lemma 2.3 and (Sala and Ariño 2008, Proposition 3.1)).

From these tables and figures, we can see that, for this particular example, the conditions proposed performed well against the other conditions and can be used as an alternative to them.

Note that the *Grouping the membership points* partition strategy performed better than the *Dividing and Discarding* strategy, but, as can be seen from the Figures presented, it does not guarantee that it will be less conservative with every partition iteration (unlike the other sum relaxation strategies presented). This stems from the fact that a simplex found to cover a new group that has just been split can add points that were not being considered in the LMIs before. This problem is however mitigated as the simplices volume approaches zero.

It is also interesting to note that, for this example, after a reasonable number of iterations of the proposed partition strategies, the use of Lemma 2.3 instead of Lemma 2.1 did not reduce the conservativeness and only increased the computational cost (as can be more explicitly seen by comparing the results for the **CNO** transformation with and without Lemma 2.3 on Table 3.1).

Table 3.2: Comparison of different methods of sum relaxation for a linear control law in example 3.1 (minimum number of rows of LMIs required to find a stabilizing control law)

Method/b	4	5	5.5	6	6.3	6.6	6.7	6.72	6.75
Sala and Ariño 2007, Proposition 2 with Sala and Ariño 2008, Proposition 3.1	24	34	46	114	276	2760	-	-	-
Sala and Ariño 2007, Theorem 5 with Sala and Ariño 2008, Proposition 3.1	32	32	32	32	32	32	32	32	32
Kruszewski et al. 2009, Algorithm 1 with Sala and Ariño 2008, Proposition 3.1	28	40	124	124	148	472	1720	1720	-
Dividing and Discarding	86	134	248	398	-	-	-	-	-
Dividing and Discarding with Sala and Ariño 2008, Proposition 3.1	28	40	124	172	172	292	760	760	928
Grouping the membership points-**CNO**	20	26	26	26	38	86	176	-	-
Grouping the membership points-**CNO** with Sala and Ariño 2008, Proposition 3.1	28	28	28	40	52	64	172	172	-
Grouping the membership points-**RNO-INO**	50	50	86	98	128	278	-	-	-
Grouping the membership points-**RNO-INO** with Sala and Ariño 2008, Proposition 3.1	52	52	64	100	172	376	556	556	640

Table 3.3: Comparison of different methods of sum relaxation for a switched control law in example 3.1 (minimum number of rows of LMIs required to find a stabilizing control law)

Method/b	4	5	5.5	6	6.3	6.6	6.7	6.72	6.75
Dividing and Discarding	86	86	86	86	86	110	-	-	-
Dividing and Discarding with Sala and Ariño 2008, Proposition 3.1	28	28	52	52	100	136	760	760	760
Grouping the membership points-**CNO**	14	20	26	26	26	86	98	128	-
Grouping the membership points-**CNO** with Sala and Ariño 2008, Proposition 3.1	28	28	28	40	52	52	172	172	352
Grouping the membership points-**RNO-INO**	26	32	50	56	98	182	-	-	-
Grouping the membership points-**RNO-INO** with Sala and Ariño 2008, Proposition 3.1	52	52	52	64	112	196	376	544	640

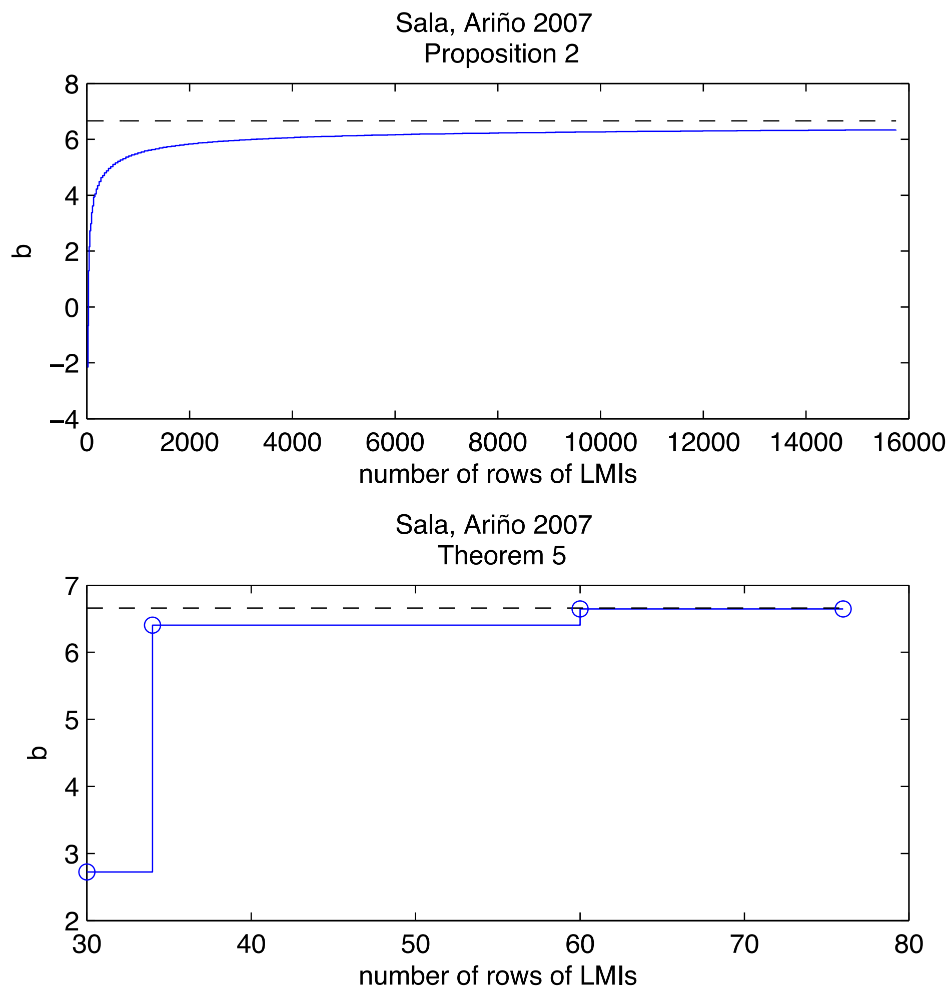

Figure 3.9: Maximum feasible b for a given number of rows of LMIs using the conditions in (Sala and Ariño 2007). Since these are *shape-independent* conditions, the maximum feasible value was found using the necessary and asymptotically sufficient conditions from (Kruszewski et al. 2009) and is displayed as the dashed line.

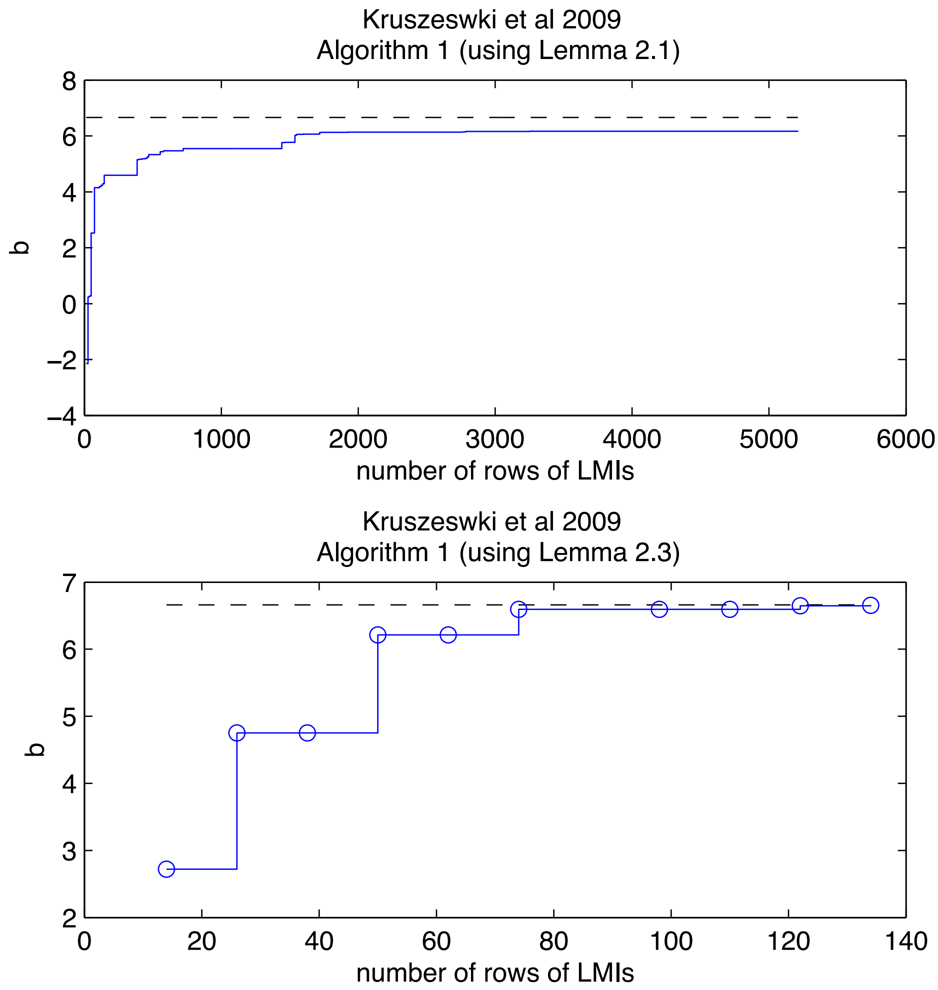

Figure 3.10: Maximum feasible b for a given number of rows of LMIs using the conditions in (Kruszewski et al. 2009, Algorithm 1). Since these are *shape-independent* conditions, the maximum feasible value was found using the necessary and asymptotically sufficient conditions from (Kruszewski et al. 2009) and is displayed as the dashed line.

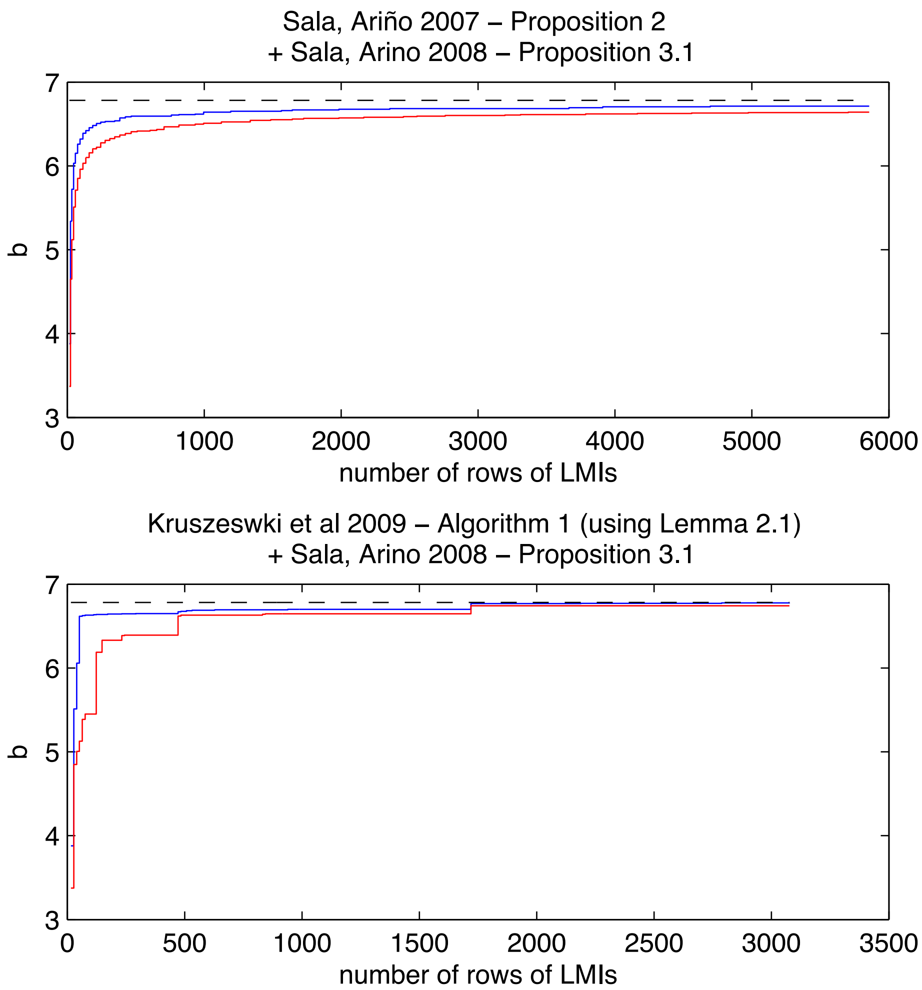

Figure 3.11: Maximum feasible b for a given number of rows of LMIs using the conditions in (Sala and Ariño 2007) and (Kruszewski et al. 2009, Algorithm 1) together with the shape relaxation from (Sala and Ariño 2008). The blue line represents a PDC control law, whereas the red one represents a linear control law. The maximum feasible value of b found with any of the methods in Example 3.1 is displayed as a dashed line.

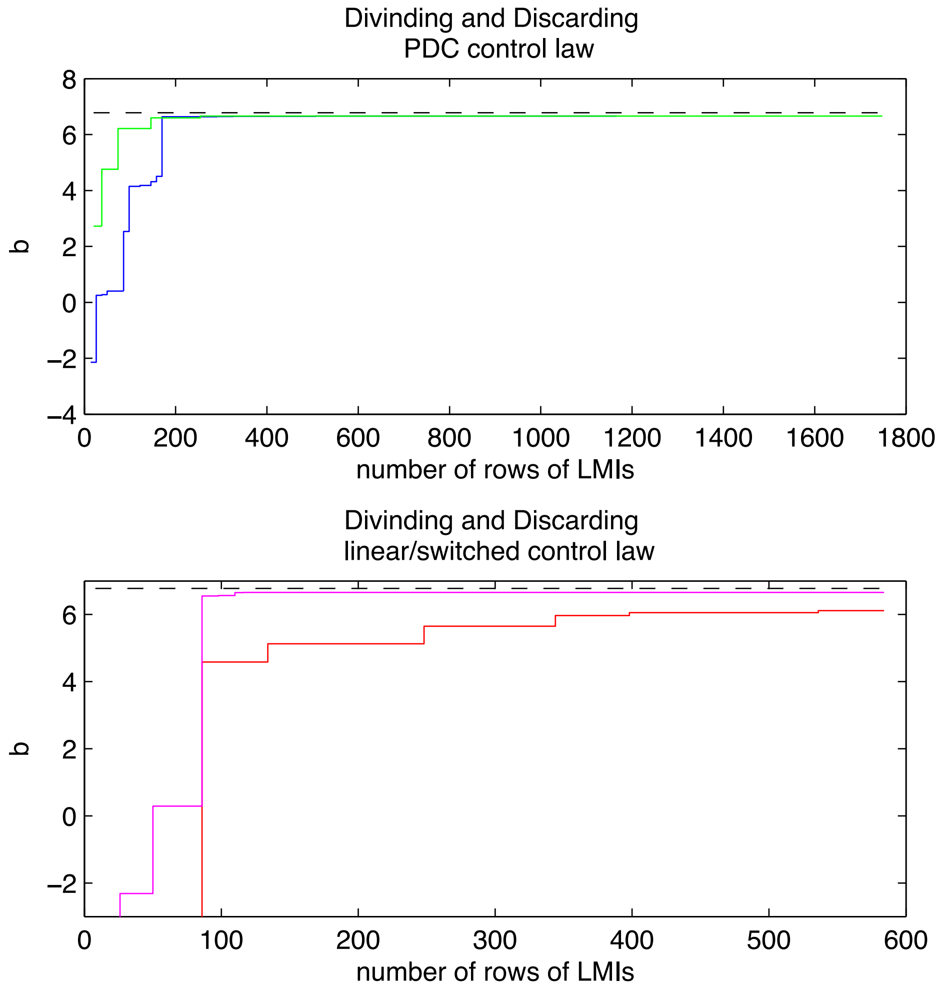

Figure 3.12: Maximum feasible b for a given number of rows of LMIs using the *Dividing and Discarding* partition strategy. The blue line represents the conditions for a PDC control law using Lemma 2.1, the green one represents the conditions for a PDC control law using Lemma 2.3, the red one represents a linear control law, and the magenta line represents a switched control law. The maximum feasible value of b found with any of the methods in Example 3.1 is displayed as a dashed line.

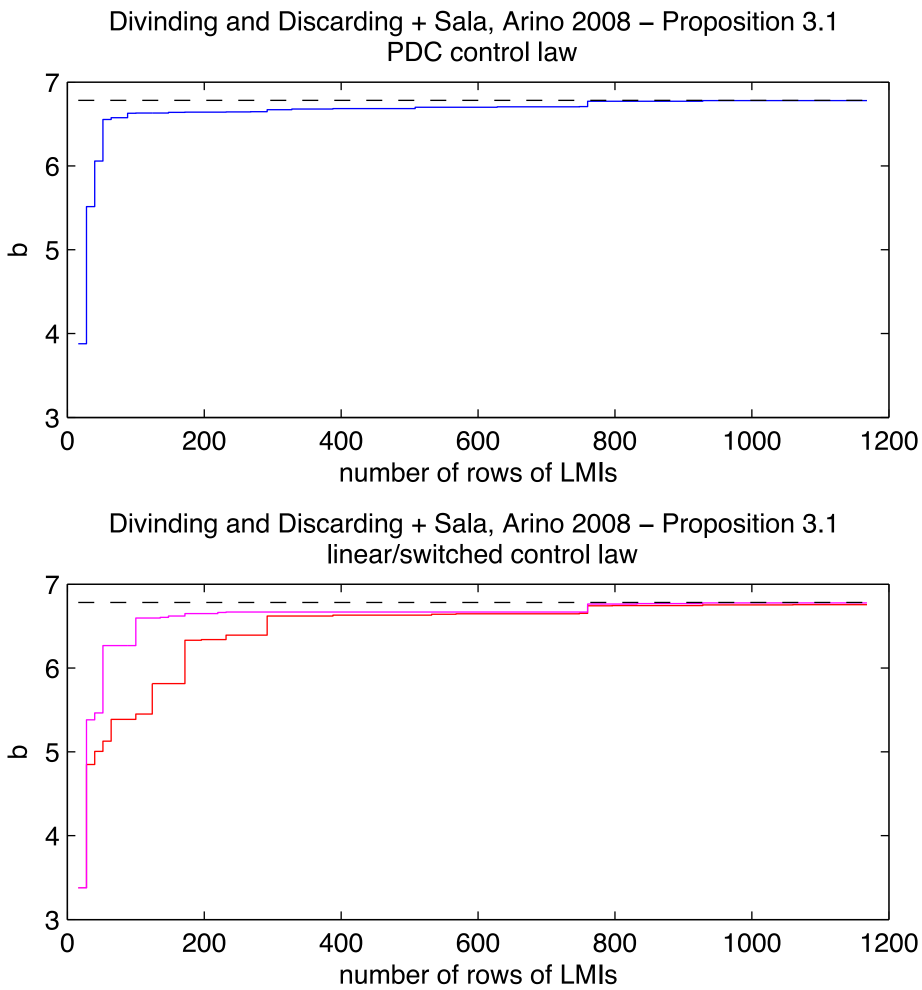

Figure 3.13: Maximum feasible b for a given number of rows of LMIs using the *Dividing and Discarding* partition strategy together with proposition 3.1 from (Sala and Ariño 2008). The blue line represents a PDC control law, the red one represents a linear control law, and the magenta line represents a switched control law. The maximum feasible value of b found with any of the methods in Example 3.1 is displayed as a dashed line.

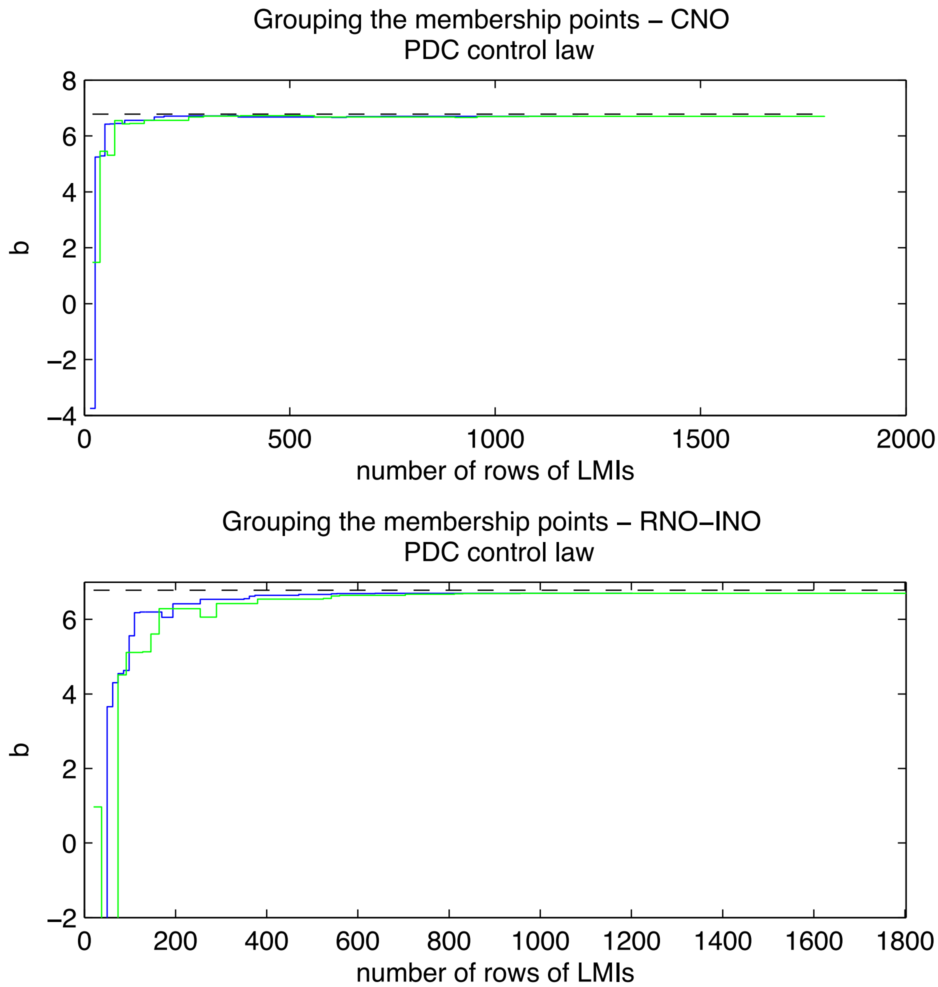

Figure 3.14: Maximum feasible b for a given number of rows of LMIs using the *Grouping the membership points* partition strategy. The blue line represents the conditions for a PDC control law using Lemma 2.1 and the green one represents the conditions for a PDC control law using Lemma 2.3. The maximum feasible value of b found with any of the methods in Example 3.1 is displayed as a dashed line.

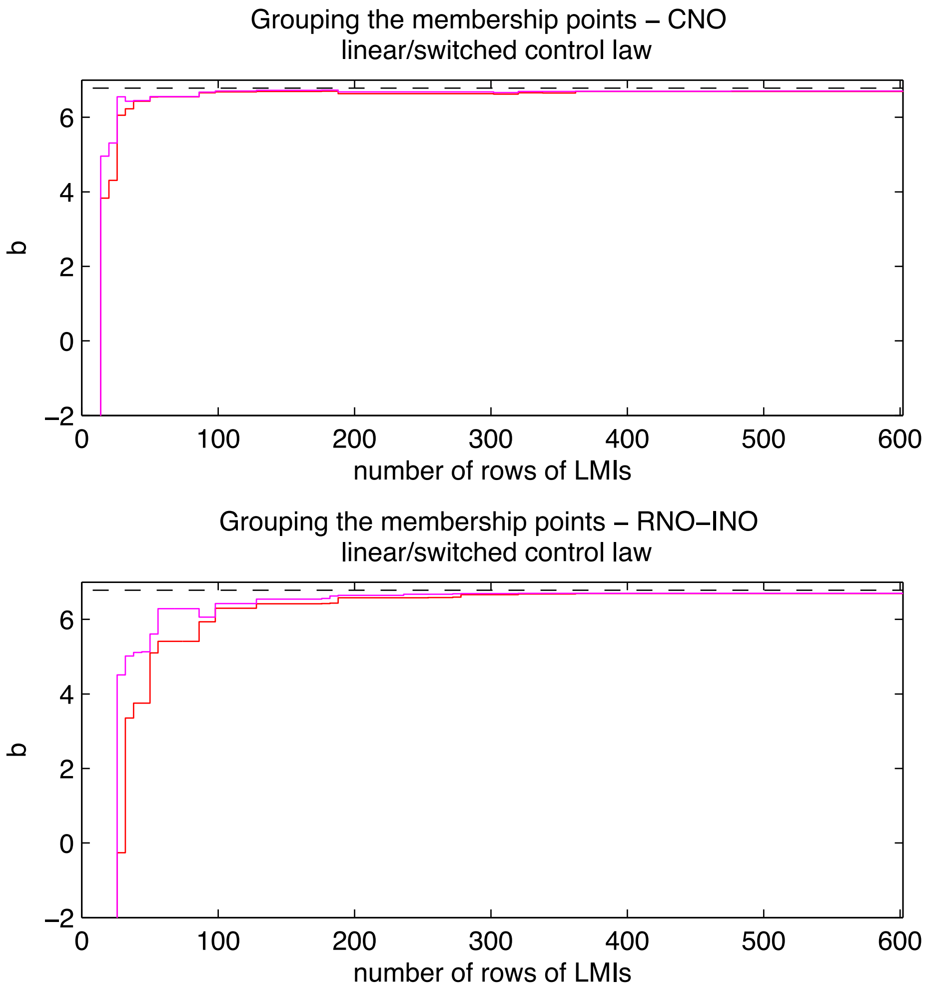

Figure 3.15: Maximum feasible b for a given number of rows of LMIs using the *Grouping the membership points* partition strategy. The red line represents a linear control law, and the magenta line represents a switched control law. The maximum feasible value of b found with any of the methods in Example 3.1 is displayed as a dashed line.

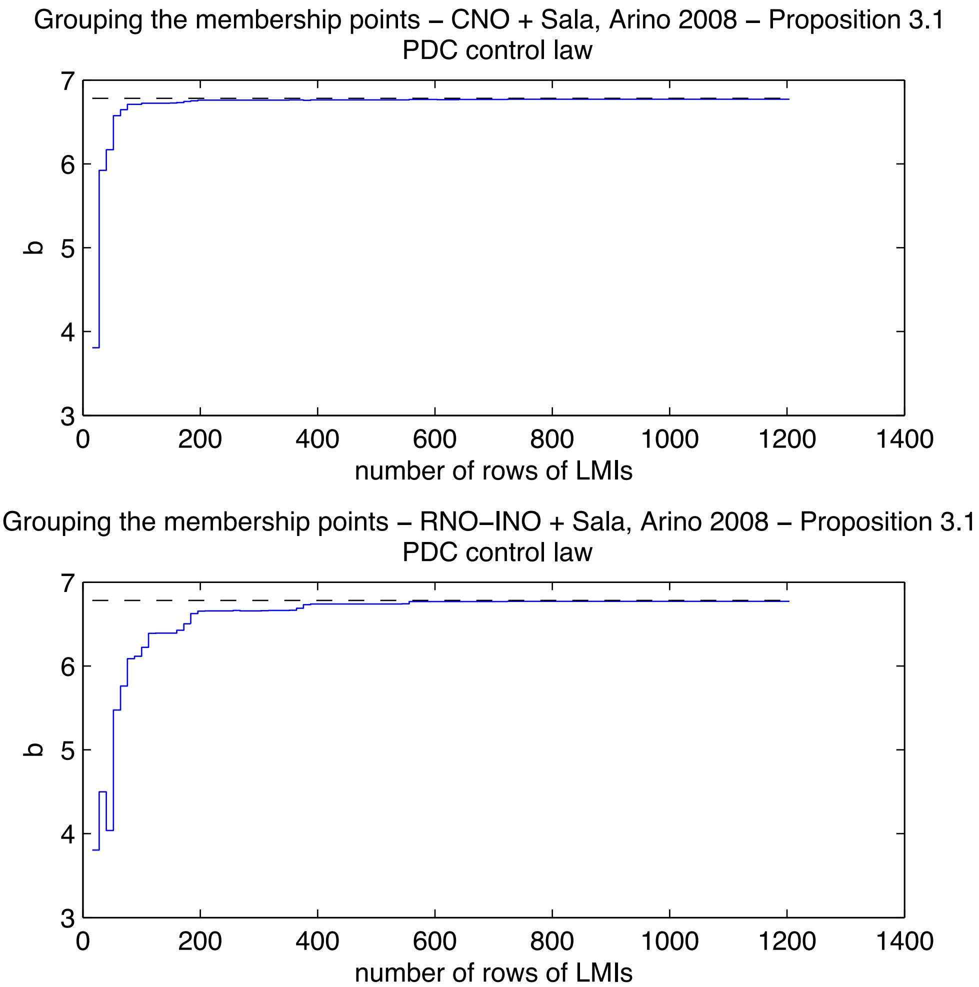

Figure 3.16: Maximum feasible b for a given number of rows of LMIs using the *Grouping the membership points* partition strategy together with proposition 3.1 from (Sala and Ariño 2008). The blue line represents the conditions for a PDC control law using Lemma 2.1. The maximum feasible value of b found with any of the methods in Example 3.1 is displayed as a dashed line.

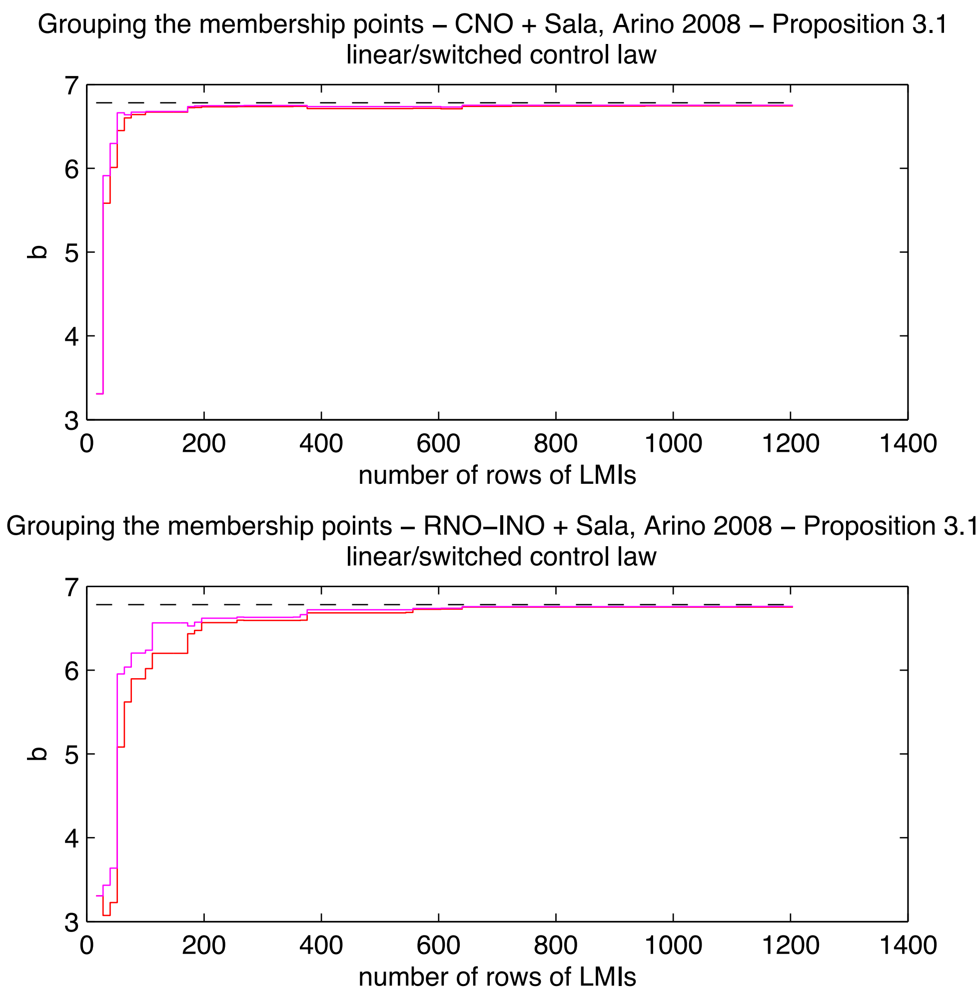

Figure 3.17: Maximum feasible b for a given number of rows of LMIs using the *Grouping the membership points* partition strategy together with proposition 3.1 from (Sala and Ariño 2008). The red line represents a linear control law, and the magenta line represents a switched control law. The maximum feasible value of b found with any of the methods in Example 3.1 is displayed as a dashed line.

Table 3.4: Complexity analysis for the different methods of sum relaxation used in example 3.1.

Method	Number of rows of LMIs	Number of scalar decision variables
Linear control law	8	5
PDC control law using Lemma 2.1	14	9
PDC - Sala and Ariño 2007, Proposition 2 with q summations	$2+\frac{(2+q)!}{q!}$	9
PDC - Sala and Ariño 2007, Theorem 5 with degree q	$\begin{cases} 16+\sum_{i=0}^{\text{floor}(q/2)} 2\binom{2+q-2i}{q-2i}, & \text{if } q \text{ is even,} \\ 8+\sum_{i=0}^{\text{floor}(q/2)} 2\binom{2+q-2i}{q-2i}, & \text{if } q \text{ is odd} \end{cases}$	$9+\sum_{i=0}^{\text{floor}(q/2)} 3^i(2\times 3^i+1)$
PDC - Kruszewski et al. 2009, Algorithm 1 with n_s simplices	$2+12n_s$	9
PDC - Kruszewski et al. 2009, Algorithm 1 with Lemma 2.3 and n_s simplices	$2+18n_s$	$9+15n_s$
Partition strategies proposed in this chapter with n_s simplices - PDC control	$2+12n_s$	9
Partition strategies proposed in this chapter with n_s simplices - linear control	$2+6n_s$	5
Partition strategies proposed in this chapter with n_s simplices - switched control	$2+6n_s$	$3+2n_s$
PDC - Lam and Narimani 2010, Theorem 2 with n_p points	$14+2n_p$	27
Using Sala and Ariño 2008, Proposition 3.1 for PDC control (any of the methods)	$+2$	$+3$
Using Sala and Ariño 2008, Proposition 3.1 for linear control (any of the methods)	same as PDC with (Sala and Ariño 2008)	4 less than PDC
Using Sala and Ariño 2008, Proposition 3.1 for switched control (Partition strategies proposed in this chapter)	same as PDC with (Sala and Ariño 2008)	$+3$

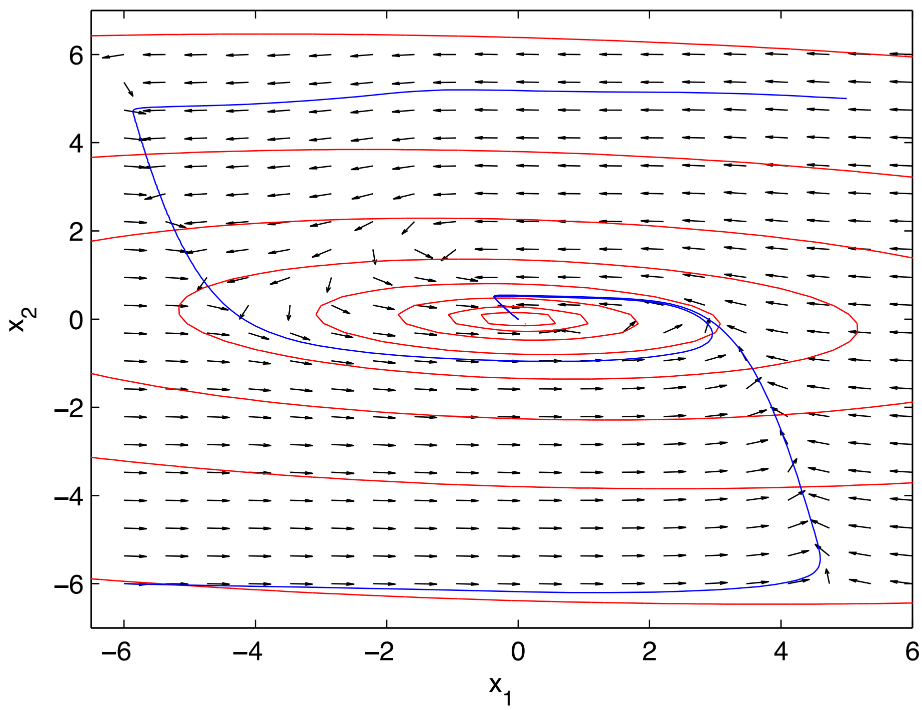

Figure 3.18: Linear stabilizing controller's phase plane for $b = 6.75$. The arrows represent the direction of the vector field in each point in space, the red level curves are the level curves of the Lyapunov function, and the blue lines are the system's trajectory from two different initial conditions.

As discussed earlier, none of the *shape-independent* conditions are able to find a stabilizing controller for $b > 6.661$ whereas some of the *shape-dependent* conditions were able to find stabilizing controllers up to 6.7803.

In order to illustrate the system's behaviour in closed loop, we obtained controllers using the *Grouping the membership points* strategy for $b = 6.75$.

In the linear control law case, we made use of the **RNO-INO** transformation to find the simplices, and the controller gains found were

$$K = \begin{bmatrix} -0.9260 & 0.2970 \end{bmatrix},$$

and the Lyapunov parameter matrix

$$P = \begin{bmatrix} 27.2336 & -1.1192 \\ -1.1192 & 1.8838 \end{bmatrix}.$$

The system's phase plane in closed loop is presented in Figure 3.18.

In the PDC control law case, we made use of the **CNO** transformation to find the simplices, and the controllers gains found were

$$K_1 = \begin{bmatrix} -1.2018 & 2.1324 \end{bmatrix},$$
$$K_2 = \begin{bmatrix} 0.3414 & -0.9099 \end{bmatrix},$$
$$K_3 = \begin{bmatrix} -0.5113 & 3.1824 \end{bmatrix},$$

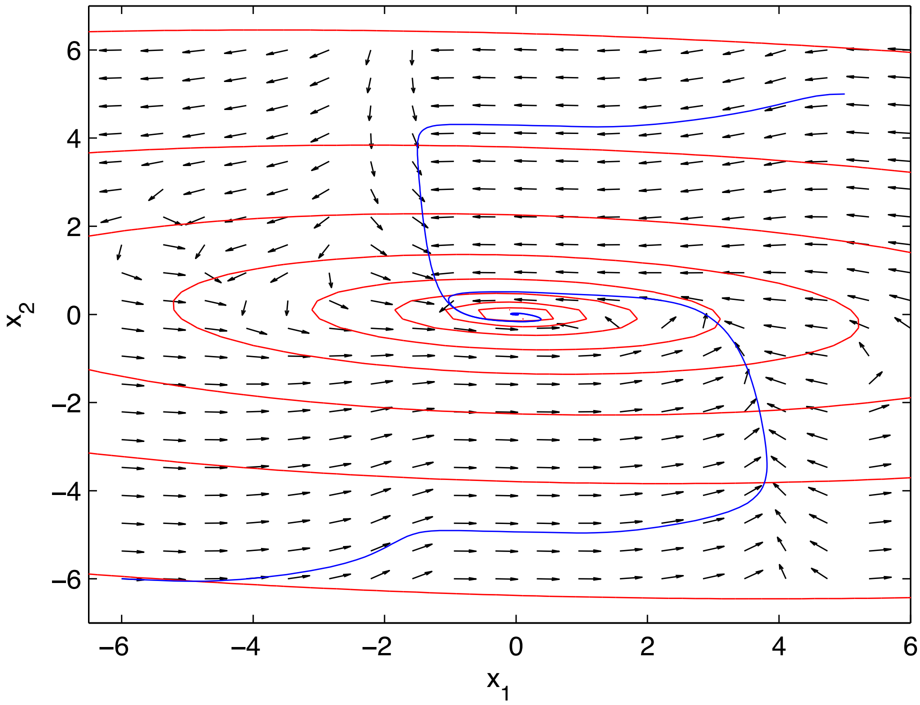

Figure 3.19: PDC stabilizing controller's phase plane for $b = 6.75$. The arrows represent the direction of the vector field in each point in space, the red level curves are the level curves of the Lyapunov function, and the blue lines are the system's trajectory from two different initial conditions.

and the Lyapunov parameter matrix

$$P = \begin{bmatrix} 27.0756 & -1.0860 \\ -1.0860 & 1.8352 \end{bmatrix}.$$

The system's phase plane in closed loop is presented in Figure 3.19.

In the switched control law case, we made use of the **CNO** transformation to find the simplices. The Lyapunov parameter matrix found was

$$P = \begin{bmatrix} 28.3290 & -1.1073 \\ -1.1073 & 1.8532 \end{bmatrix}.$$

The system's phase plane in closed loop is presented in Figure 3.20.

We are not presenting the controller gains found for this case since 100 partitions of the image space were used (correspoding to 100 different gain matrices). In addition, in order to fully describe the control law found, we would also need to present the simplices that describe each image space partition.

Finally, we also made use of modification to allow for a continuous controller to be found from the piecewise controller. The conditions were found to be feasible for $b = 6.75$ for the same simplices used in the switched control law synthesis.

An initial idea could be to make use of the original 1001 samples from the membership functions' image space to sample the controller gains (thus guaranteeing that at least one sample from each switching region is sampled) and consider that the intersample values are given by linear interpolation so that we can make use of the Tensor Product Model Transformation to recover a continuous more compact representation of the controller.

Note, however, that since we are using a lot more samples than switching regions we would be sampling the same value many times inside of each region, thus leading to a staircase-like membership

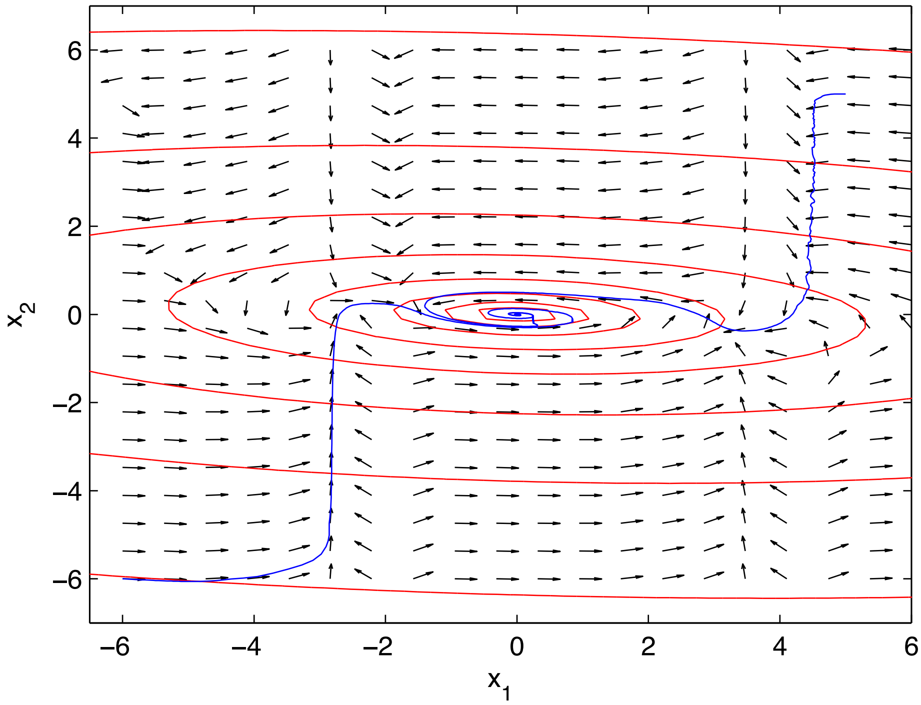

Figure 3.20: Stabilizing switched controller's phase plan for $b = 6.75$. The arrows represent the direction of the vector field in each point in space, the red level curves are the level curves of the Lyapunov function, and the blue lines are the system's trajectory from two different initial conditions.

function for the gains (implying that the membership values would change too fast from one switching region to the other).

In order to avoid this and "smooth" the controller's membership functions a little, we go through the 1001 samples and check their current switching region. By doing this we are capable of defining intervals in the premise variable corresponding to each switching region[2]. We then pick only one sample from each region. For the extremal regions, we pick the extremes, whereas for all the other regions we pick the middle point. In addition, we make use of the piecewise sinusoidal functions proposed in 2.4 to generate continuously differentiable membership functions.

The membership functions obtained from both approaches are presented in Figure 3.21.

The fuzzy controller gains found were

$$K_1 = \begin{bmatrix} 1.1617 & 11.2484 \end{bmatrix},$$
$$K_2 = \begin{bmatrix} -1.4358 & 11.0590 \end{bmatrix},$$
$$K_3 = \begin{bmatrix} -0.4847 & -5.8339 \end{bmatrix},$$

and the Lyapunov parameter matrix found was

$$P = \begin{bmatrix} 29.2791 & -1.1450 \\ -1.1450 & 1.9159 \end{bmatrix},$$

and the system's phase plane is presented in Figure 3.22.

[2]note that for a case with more than one premise variable, this would mean dealing with each of the image spaces of the tensor product separately - *i.e.* we would only be able to deal with the first case discussed previously in this chapter.

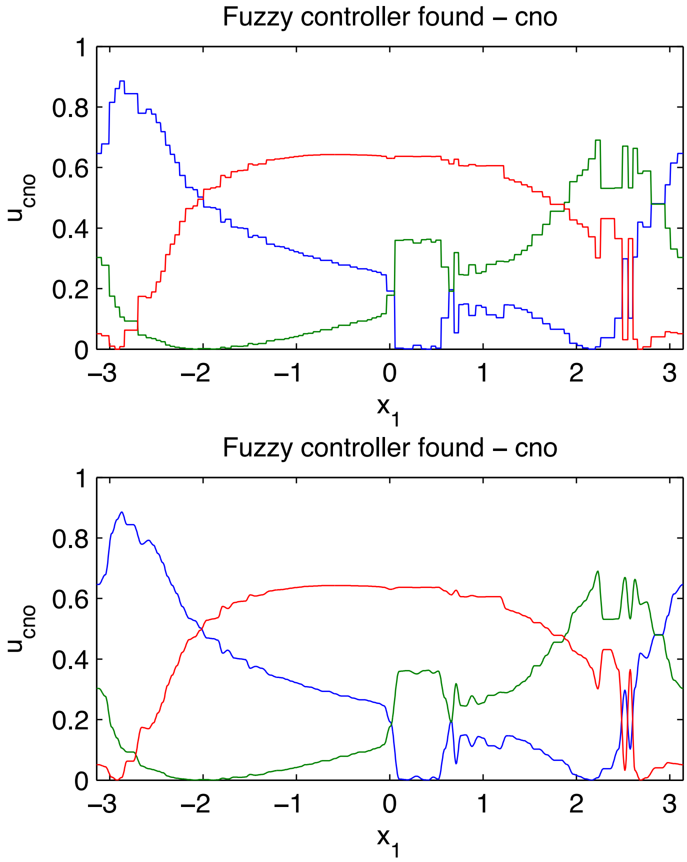

Figure 3.21: Fuzzy controller's membership functions. The plot on the top presents the membership functions using all 1001 samples and linear interpolation. The plot on the bottom presents a "smoothed" approach using less samples and a piecewise sinusoidal interpolation.

Guaranteed cost $\mathcal{H}_\infty$ controller

Consider a system described by:

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i \left[A_i \mathbf{x} + B_i \mathbf{u} + B_{w_i} \mathbf{w} \right]$$
$$\mathbf{z} = \sum_{i=1}^{r} h_i \left[C_i \mathbf{x} + D_i \mathbf{u} + D_{w_i} \mathbf{w} \right]$$

its $\mathcal{H}_\infty$ norm is the smallest γ such that

$$\|\mathbf{z}(t)\|_2 \leq \gamma \|\mathbf{w}\|_2,$$

for a system with zero initial conditions.

By making use of a linear state feedback control law, $\mathbf{u} = K\mathbf{x}$, we may search for a controller with

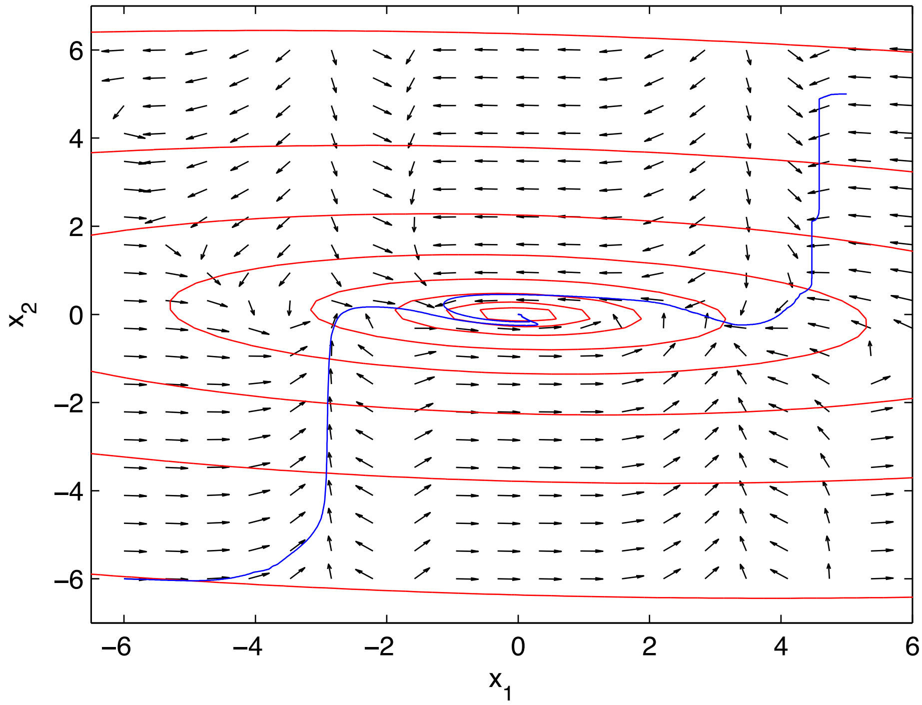

Figure 3.22: Stabilizing fuzzy (reconstructed from a switched controller) controller's phase plan for $b = 6.75$. The arrows represent the direction of the vector field in each point in space, the red level curves are the level curves of the Lyapunov function, and the blue lines are the system's trajectory from two different initial conditions.

minimum guaranteed cost by minimizing δ, subject to

$$\sum_{i=1}^{r} h_i Q_i < 0,$$

$$P > 0,$$

$$Q_i = \begin{bmatrix} A_i P + P A_i^T + B_i X + X^T B_i^T & B_{w_i} & P C_i^T + X^T D_i^T \\ * & -I & D_{w_i}^T \\ * & * & -\delta I \end{bmatrix}$$

and the $\mathcal{H}_\infty$ norm is guaranteed to be less than $\gamma = \sqrt{\delta}$ and the controller gains will be given by $K = XP^{-1}$.

This problem can be solved using the techniques proposed in this chapter. In addition to this, the partitions generated by these techniques can be taken advantage of and used to define a control law that switches according to each region of the membership functions' image space (as presented in the previous section). This control law will have the form

$$\mathbf{u} = K^{(m)}\mathbf{x}, \text{ if } \sigma = m.$$

In that regard, we may look for a guaranteed cost controller by minimizing δ, subject to

$$\sum_{j=1}^{r} \tilde{h}_j^{(m)} \sum_{i=1}^{r} g_{ij}^{(m)} Q_i^{(m)} < 0, \forall m$$

$$P > 0,$$

$$Q_i^{(m)} = \begin{bmatrix} A_iP + PA_i^T + B_iX^{(m)} + (X^{(m)})^T B_i^T & B_{w_i} & PC_i^T + (X^{(m)})^T D_i^T \\ * & -I & D_{w_i}^T \\ * & * & -\delta I \end{bmatrix}$$

and the $\mathcal{H}_\infty$ norm is guaranteed to be less than $\gamma = \sqrt{\delta}$ and the controller gains will be given by $K^{(m)} = X^{(m)}P^{-1}$.

In addition, as explained previously, by changing the LMI conditions such that the switched gains are also valid in the neighbouring regions, we will be able to reconstruct a continuous controller from the conditions. In that regard we modify the LMI to

$$\sum_{j=1}^{r} \tilde{h}_j^{(m)} \sum_{i=1}^{r} g_{ij}^{(m)} Q_i^{(n)} < 0, \forall m, n = m \text{ or } n \text{ is a neighbour of } m$$

$$P > 0,$$

$$Q_i^{(n)} = \begin{bmatrix} A_iP + PA_i^T + B_iX^{(n)} + (X^{(n)})^T B_i^T & B_{w_i} & PC_i^T + (X^{(n)})^T D_i^T \\ * & -I & D_{w_i}^T \\ * & * & -\delta I \end{bmatrix}$$

so that the $\mathcal{H}_\infty$ norm is guaranteed to be less than $\gamma = \sqrt{\delta}$, the controller gains will be given by $K^{(m)} = X^{(m)}P^{-1}$, and a continuous controller may be reconstructed by means of the Tensor Product Model Transformation (in a similar way as done in the previous example).

Example 3.2. Consider the 3 rule system described by matrices:

$$A_1 = \begin{bmatrix} 1.59 & -7.29 \\ 0.01 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad B_{w_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad D_1 = 0, \quad D_{w_1} = 0$$

$$A_2 = \begin{bmatrix} 0.02 & -4.64 \\ 0.35 & 0.21 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 8 \\ 0 \end{bmatrix}, \quad B_{w_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad D_2 = 0, \quad D_{w_2} = 0$$

$$A_3 = \begin{bmatrix} -2 & -4.33 \\ 0 & 0.05 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad B_{w_3} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad D_3 = 0, \quad D_{w_3} = 0$$

with the membership functions

$$h_1(\theta) = \frac{\cos(10x_1) + 1}{4}, \qquad h_2(\theta) = \frac{\sin(10x_1) + 1}{4}, \qquad h_3(\theta) = 0.5 - \frac{\cos(10x_1) + \sin(10x_1)}{4}$$

By taking 1001 samples from $x_1 \in [-\pi/10, \pi/10]$ (since the membership functions are periodic, this interval covers all of their image space), we find an approximation error polytope (in the image space) that will be smaller than $\begin{bmatrix} \frac{\pi}{4000} & \frac{\pi}{4000} & -\frac{\pi}{2000} \\ \frac{\pi}{4000} & -\frac{\pi}{4000} & 0 \\ -\frac{\pi}{4000} & \frac{\pi}{4000} & 0 \\ -\frac{\pi}{4000} & -\frac{\pi}{4000} & \frac{\pi}{2000} \end{bmatrix}$, with everyone row representing a vertex. In addition, we get an approximation error radius equal to $\frac{\pi}{2000}$.

We made use of the *Dividing and Discarding* partition strategy and the *Grouping the membership points* partition strategy with **RNO-INO** transformation to find both a linear control law and a switched control

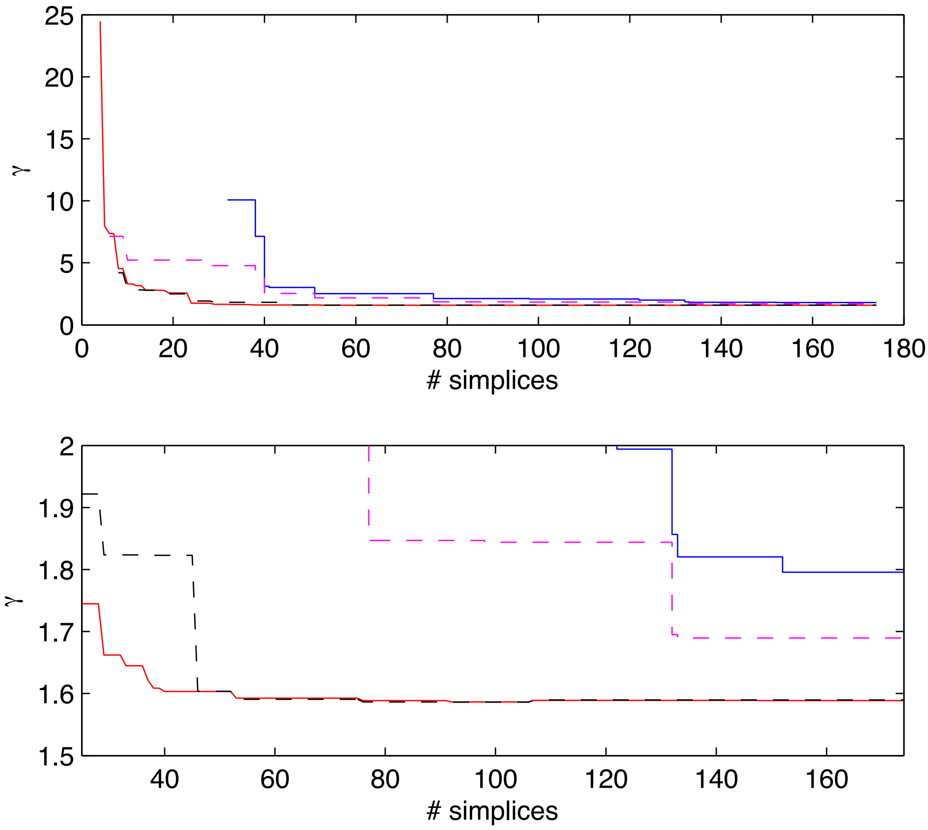

Figure 3.23: Guaranteed $\mathcal{H}_\infty$ norm for a given number of simplices. The top plot presents the results obtained for the number of simplices tested, whereas the bottom plot presents a zoom in of the upper plot. The blue line represents the guaranteed cost found by the linear control law using the *Dividing and Discarding* partition strategy. The red line represents the linear control law using the *Grouping the membership points* partition strategy with **RNO-INO** transformation. The dashed magenta line represents the guaranteed cost found by the switched control law using the *Dividing and Discarding* partition strategy. The black dashed line represents the linear control law using the *Grouping the membership points* partition strategy with **RNO-INO** transformation.

law. Figure 3.23 presents a comparison of the results obtained from these methods by checking the guaranteed $\mathcal{H}_\infty$ norm they were able to find according to the number of simplices used.

In addition to this, we also made use of the *Grouping the membership points* partition strategy with **RNO-INO** transformation to find a continuous controller, reconstructed from a switched one by means of the Tensor Product Model Transformation.

We made use of 4001 samples of the membership functions from $x_1 \in [-\pi/10, \pi/10]$, resulting in an error polytope of $\begin{bmatrix} \frac{\pi}{16000} & \frac{\pi}{16000} & -\frac{\pi}{8000} \\ \frac{\pi}{16000} & -\frac{\pi}{16000} & 0 \\ -\frac{\pi}{16000} & \frac{\pi}{16000} & 0 \\ -\frac{\pi}{16000} & -\frac{\pi}{16000} & \frac{\pi}{8000} \end{bmatrix}$.

We made use of 600 groupings of the samples and the controller found had a guaranteed cost of 1.5374. Just as in the previous example, we are able to reconstruct a fuzzy controller from the switched controller found by making use of the Tensor Product Model Transformation (and the same observations are valid here as to how to obtain a smoother set of membership functions). In that regard the

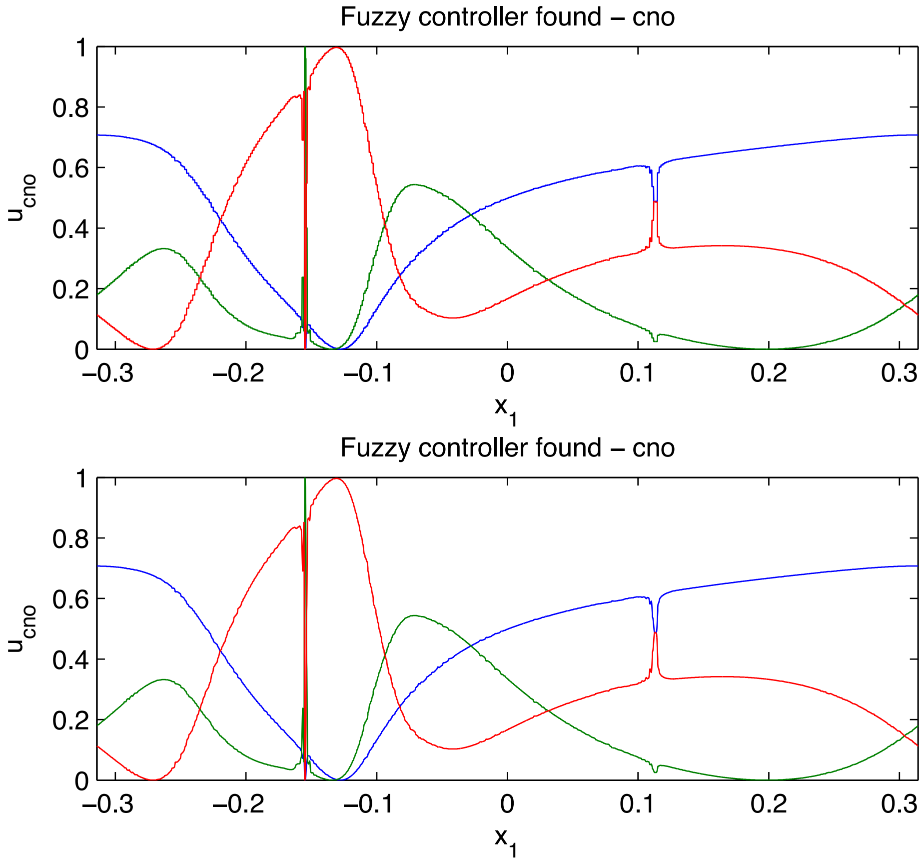

Figure 3.24: Membership functions found in Example 3.2. A single period of the function is presented. The plot on the top presents the membership functions using all 4001 samples and linear interpolation. The plot on the bottom presents a "smoothed" approach using less samples and a piecewise sinusoidal interpolation.

membership functions are presented in Figure 3.24. The controller gains found were

$$K_1 = \begin{bmatrix} 1.1011 & 1.0872 \end{bmatrix},$$
$$K_2 = \begin{bmatrix} -2.3099 & 2.9830 \end{bmatrix},$$
$$K_3 = \begin{bmatrix} -2.9565 & -0.0449 \end{bmatrix}.$$

In order to check the controller's performance in closed loop we simulated it with zero initial condition and a disturbance signal $w(t) = \cos(\frac{\pi}{10}t)$. The system's output $z(t)$, as well as the disturbance signal are presented in Figure 3.25.

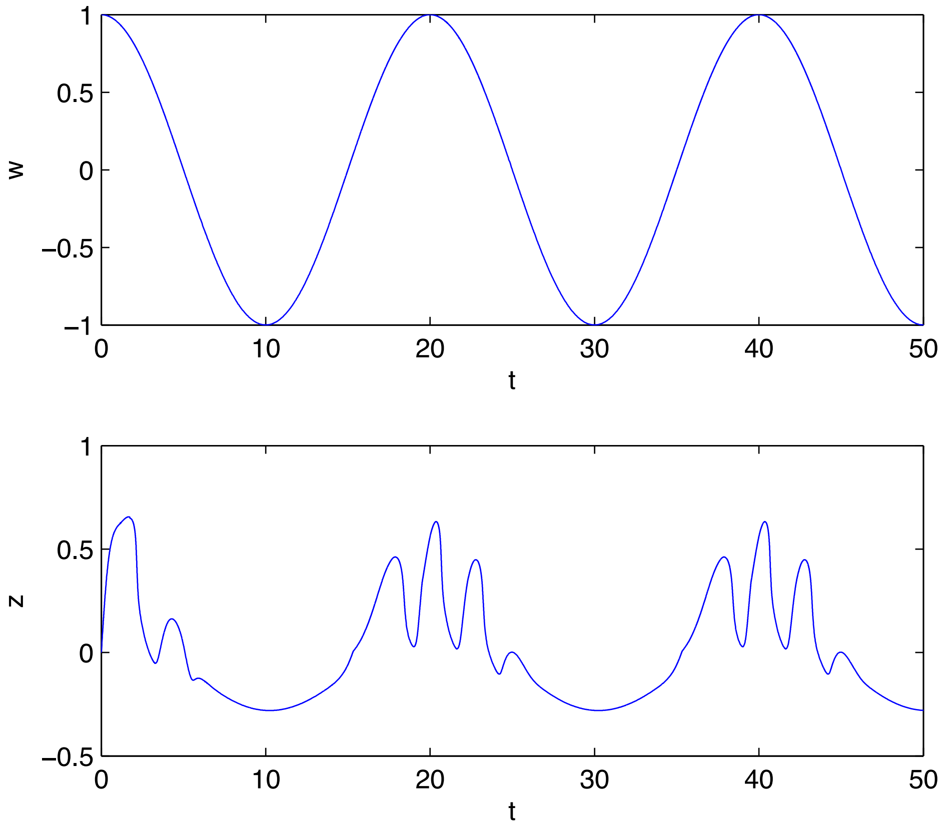

Figure 3.25: System's disturbance rejection in Example 3.2. The top plot is the disturbance signal $w(t) = \cos(\frac{\pi}{10}t)$, whereas the bottom plot is the system's output z.

4 Rule Reduction

Quite often, a model with a large number of rules might be undesirable. Be it because their analysis/synthesis conditions require a too large number of LMIs that make them prohibitly costly, be it because we are searching for a more convenient representation, and thus require a model with a reduced number of rules.

In this chapter, we propose the use of the tensor product model transformation technique for rule reduction. We aim to show that, even without the sampling step, the HOSVD and the convex hull manipulation steps could be used to reduce the number of rules of a given TS fuzzy system. In addition, we also present how a polytopic uncertainty could be added to the reduced system such that this uncertain reduced system represents the original one. Afterwards, we present LMI conditions that allow the transformation of this polytopic uncertainty into other kinds of uncertainty that are more common in the TS literature.

The presentation and the results in this chapter were based on (Campos, Tôrres, and Palhares 2015).

4.1 Rule reduction by means of tensor approximation

As stated in Section 2.4 (in the equivalence between equations (2.10) and (2.11)), a tensor product TS representation of a matrix function $L(\mathbf{v}) : \mathbb{R}^{\ell} \to \mathbb{R}^{\kappa_1 \times \kappa_2}$

$$L(\mathbf{v}) = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \cdots \sum_{i_\ell=1}^{r_\ell} \prod_{n=1}^{\ell} \mu_{i_n}^{(n)}(v_n) L_{i_1 i_2 \ldots i_\ell},$$

with $L_{i_1 i_2 \ldots i_\ell} \in \mathbb{R}^{\kappa_1 \times \kappa_2}$, can be written with tensor notation as

$$L(\mathbf{v}) = \mathcal{L} \mathop{\times}_{n=1}^{\ell} \boldsymbol{\mu}_n^T(v_n), \tag{4.1}$$

in which $\mathcal{L} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_\ell \times \kappa_1 \times \kappa_2}$ is a tensor whose element $l_{i_1 i_2 \ldots i_\ell k_1 k_2}$ corresponds to the element in row k_1 and column k_2 of matrix $L_{i_1 i_2 \ldots i_\ell}$, and $\boldsymbol{\mu}_n(v_n)$ is a vector formed by stacking the membership functions related to premise variable v_n.

By applying the HOSVD over tensor $\mathcal{L}$, we arrive at an approximation

$$\mathcal{L} \approx \mathcal{S} \mathop{\times}_{n=1}^{\ell} U_i.$$

Note that the HOSVD step may be used simply to find a minimal TS representation, by discarding no nonzero higher order singular values. In this case, the equation above is actually an equality.

In cases where singular values are discarded, we can write a tensor describing the approximation error

$$\hat{\mathcal{L}} = \mathcal{L} - \left(\mathcal{S} \overset{\ell}{\underset{n=1}{\times}} U_i \right),$$

whose Frobenius norm will be limited by (De Lathauwer, De Moor, and Vandewalle 2000a, equation 24)

$$\|\hat{\mathcal{L}}\| \leq \sum_{i_1=1}^{k_1} \overline{\sigma}_{i_1}^{(1)^2} + \sum_{i_2=1}^{k_2} \overline{\sigma}_{i_2}^{(2)^2} + \cdots + \sum_{i_\ell=1}^{k_\ell} \overline{\sigma}_{i_\ell}^{(\ell)^2},$$

in which $\overline{\sigma}_{i_m}^{(m)^2}$ denotes the square of the discarded *m-mode* singular values, and k_m the number of *m-mode* singular values discarded.

This reduction step can be seen as searching for a tensor with rank restrictions that approximates $\mathcal{L}$. Note, however, that if we discard columns in more than one direction (*i.e.* if we impose a rank constraint in more than one direction), the solution given by the truncation of the HOSVD will not usually be optimal. This solution can be refined, for instance, by using the Higher Order Orthogonal Iteration (HOOI) (De Lathauwer, De Moor, and Vandewalle 2000b) which is an alternating least squares method that converges to a *locally* optimal solution.

Note that, the weight matrices found by the HOSVD (or HOOI) do not posses membership functions' characteristics (SN and NN) and induce transformations over the original membership functions that would make them loose these characteristics. In that regard, just like in the usual tensor product model transformation technique, we make use of the convex hull manipulation step in order to impose desired characteristics over the weight matrices. Since this step does not really change the tensor approximation, we have

$$\mathcal{L} \approx \widetilde{\mathcal{L}} \overset{\ell}{\underset{n=1}{\times}} \tilde{U}_i = \mathcal{S} \overset{\ell}{\underset{n=1}{\times}} U_i.$$

A naïve method to find a norm bounded uncertainty that could be added to the reduced model so that the uncertain model covers the original one, would be to use the upper bound of the Frobenius norm of the error between the original tensor and the approximated one. That approach would, however, completely ignore the structure of the rule reduction error.

Instead of doing that, note that we can rewrite equation (4.1) as

$$\begin{aligned} &L(\mathbf{v}) \\ &= \left(\left(\widetilde{\mathcal{L}} \overset{\ell}{\underset{n=1}{\times}} \tilde{U}_i \right) + \hat{\mathcal{L}} \right) \overset{\ell}{\underset{n=1}{\times}} \boldsymbol{\mu}_n^T(v_n), \\ &= \left(\widetilde{\mathcal{L}} \overset{\ell}{\underset{n=1}{\times}} \tilde{U}_i \right) \overset{\ell}{\underset{n=1}{\times}} \boldsymbol{\mu}_n^T(v_n) + \hat{\mathcal{L}} \overset{\ell}{\underset{n=1}{\times}} \boldsymbol{\mu}_n^T(v_n), \\ &= \widetilde{\mathcal{L}} \overset{\ell}{\underset{n=1}{\times}} \boldsymbol{\mu}_n^T(v_n)\tilde{U}_i + \hat{\mathcal{L}} \overset{\ell}{\underset{n=1}{\times}} \boldsymbol{\mu}_n^T(v_n), \\ &= \widetilde{\mathcal{L}} \overset{\ell}{\underset{n=1}{\times}} \tilde{\boldsymbol{\mu}}_n^T(v_n) + \hat{\mathcal{L}} \overset{\ell}{\underset{n=1}{\times}} \boldsymbol{\mu}_n^T(v_n), \end{aligned}$$

with $\tilde{\boldsymbol{\mu}}_n^T(v_n) = \boldsymbol{\mu}_n^T(v_n)\tilde{U}_i$.

This last equation, in turn, can be rewritten using the summations as

$$L(\mathbf{v}) = \sum_{i_1=1}^{\tilde{r}_1} \sum_{i_2=1}^{\tilde{r}_2} \cdots \sum_{i_\ell=1}^{\tilde{r}_\ell} \prod_{n=1}^{\ell} \tilde{\mu}_{i_n}^{(n)}(v_n) \tilde{L}_{i_1 i_2 \ldots i_\nu} + \Delta L,$$

with

$$\Delta L = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \cdots \sum_{i_\ell=1}^{r_\ell} \prod_{n=1}^{\ell} \mu_{i_n}^{(n)}(v_n) \hat{L}_{i_1 i_2 \ldots i_\nu}. \tag{4.2}$$

Given the discussion above, we may state the following result.

Theorem 4.1

The matrix function

$$L(\mathbf{v}) = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \cdots \sum_{i_\ell=1}^{r_\ell} \prod_{n=1}^{\ell} \mu_{i_n}^{(n)}(v_n) L_{i_1 i_2 \ldots i_\ell}$$

can be represented by the uncertain matrix function

$$\tilde{L}(\mathbf{v}) + \Delta L$$

found by the procedure described in this section, with

$$\tilde{L}(\mathbf{v}) = \sum_{i_1=1}^{\tilde{r}_1} \sum_{i_2=1}^{\tilde{r}_2} \cdots \sum_{i_\ell=1}^{\tilde{r}_\ell} \prod_{n=1}^{\ell} \tilde{\mu}_{i_n}^{(n)}(v_n) \tilde{L}_{i_1 i_2 \ldots i_\nu},$$

$$\Delta L = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \cdots \sum_{i_\ell=1}^{r_\ell} \prod_{n=1}^{\ell} \alpha_{i_n}^{(n)} \hat{L}_{i_1 i_2 \ldots i_\nu},$$

and $\alpha_{i_n}^{(n)}$ *unknown functions with the properties*

$$\sum_{i_n=1}^{r_n} \alpha_{i_n}^{(n)} = 1, \qquad \alpha_{i_n}^{(n)} \geq 0.$$

Proof. Note that ΔL in equation (4.2) is a polytopic description of the approximation error of the procedure described in this chapter, and thus the original matrix function is recovered in the uncertain matrix function by taking $\alpha_{i_n}^{(n)} = \mu_{i_n}^{(n)}(v_n)$. □

As presented in this section, we see that, in cases where we have a tensor product TS representation, the HOSVD and convex hull manipulation steps can be used to find an approximate model with a reduced number of rules. As a byproduct, we are also able to find a polytopic uncertainty, represented by equation (4.2), that, when added to the approximate model, covers the original one.

Note that, even though the rule reduction steps were presented for Tensor Product TS models, they could also be used for generic TS models by means of the change of representation

$$\sum_{i=1}^{r} h_i(\mathbf{v}) L_i = \mathcal{L} \times_1 \mathbf{h}^T(\mathbf{v}),$$

instead of equation (4.1).

Remark 4.1

Comparing the strategy proposed in this section with the usual tensor product model transformation, we can think that when we find the $\tilde{U}_i$ matrices, we are finding membership functions for the membership functions.

Remark 4.2

Note that the strategy proposed in this section is different from the usual strategies found in the literature (Baranyi 1999; Taniguchi et al. 2001), since we are dealing with approximating the tensor $\mathcal{L}$ instead of directly dealing with aproximating function $L(\mathbf{v})$.

This rule reduction strategy is best used in cases where all the membership functions attain the value 1 at some point, as is the case of the tensor product TS representations found by the sampling step. In these cases, we can think that most of the information regarding the membership functions are present in the tensor $\mathcal{L}$ and this model reduction scheme will yield good results. When that is not the case, the sampling step could always be used in order to generate tensor product TS representations with this property, reducing the conservativeness of the rule reduction.

4.2 Uncertainty transformation

Even though we were able to find an uncertain model with a reduced number of rules, the polytopic uncertainty is not very common in the TS literature since it will usually yield an even larger number of LMIs than the original model. In that regard, in the following we present LMI conditions that allow us to transform this polytopic uncertainty into a norm bounded uncertainty or into a structured linear fractional form. The conditions presented here were inspired by those in (Balas, Packard, and Seiler 2009).

Before we present the conditions, note that equation (4.2), can be rewritten, by a simple change of indices, as

$$\Delta L = \sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i,$$

with $r = \prod_{i=1}^{\ell} r_i$ and $h_i(\mathbf{v}) = \prod_{n=1}^{\ell} \mu_{i_n}^{(n)}(v_n)$.

Norm bounded uncertainty

We seek a norm bounded representation for the uncertainty such that

$$\Delta L \subset D\Delta E$$
$$\|\Delta\|_2 \leq 1.$$

We need, therefore, conditions that satisfy

$$\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i \subset D\Delta E,$$
$$||\Delta||_2 \leq 1.$$

In order for these conditions to be true, one has that

$$\Delta = D^{-1}\left(\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i\right) E^{-1},$$

for some Δ respecting the norm condition and assuming that matrices D and E are square and invertible. This condition can be rewritten as

$$\begin{aligned}
&\Delta\Delta^T \leq I \\
&I - \Delta\Delta^T \geq 0 \\
&I - D^{-1}\left(\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i\right) E^{-1}E^{-T}\left(\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i^T\right) D^{-T} \geq 0 \\
&\sum_{i=1}^{r} h_i(\mathbf{v}) \begin{bmatrix} I & D^{-1}\hat{L}_i \\ * & E^T E \end{bmatrix} \geq 0, \qquad (4.3)
\end{aligned}$$

where the last step comes from Schur's complement.

The problem is then reduced to finding matrices D and E that satisfy (4.3). We may then state the following result.

Theorem 4.2

Given a polytopic model

$$\begin{aligned}
&\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i, \\
&\sum_{i=1}^{r} h_i(\mathbf{v}) = 1, \quad h_i(\mathbf{v}) \geq 0,
\end{aligned}$$

the norm bounded uncertain model

$$\begin{aligned}
&D\Delta E, \\
&||\Delta||_2 \leq 1,
\end{aligned}$$

is such that

$$\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i \subset D\Delta E,$$

if the LMIs

$$\begin{aligned}
Y = Y^T &> 0, \\
Z = Z^T &> I, \\
\sum_{i=1}^{r} h_i(\mathbf{v}) \begin{bmatrix} I & Z\hat{L}_i \\ * & Y \end{bmatrix} &\geq 0,
\end{aligned}$$

are feasible with $D = Z^{-1}$, $E^T E = Y$.

From the theorem, we see that a way to search for a "small" uncertainty representation is through the optimization problem

$$\begin{aligned}
\min \quad & \operatorname{tr}(Y) \\
s.t. \quad & Y = Y^T > 0 \\
& Z = Z^T > I \\
& \sum_{i=1}^{r} h_i(\mathbf{v}) \begin{bmatrix} I & Z\hat{L}_i \\ * & Y \end{bmatrix} \geq 0
\end{aligned}$$

and matrices D and E are recovered by

$$D = Z^{-1}, \qquad E^T E = Y.$$

Structured linear fractional form

We seek a structured linear fractional form representation for the uncertainty such that

$$\begin{aligned}
&\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i \subset D\Delta E,\\
&\Delta = F\left[I - JF\right]^{-1}\\
&I - JJ^T > 0 \quad \text{and } J \text{ known}\\
&||F||_2 \leq 1.
\end{aligned}$$

In order for these conditions to be true, one has that

$$\Delta = D^{-1}\left(\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)E^{-1},$$

for some Δ and assuming that matrices D and E are square and invertible. However, $\Delta = F\left[I - JF\right]^{-1}$, which implies that

$$\begin{aligned}
F &= D^{-1}\left(\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)E^{-1}\left[I - JF\right],\\
F &= \Lambda^{-1}D^{-1}\left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)E^{-1},
\end{aligned}$$

with

$$\Lambda = I + D^{-1}\left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)E^{-1}J.$$

Since $F^T F \leq I$, we may write

$$E^{-T}\left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)^T D^{-T}\Lambda^{-T}\Lambda^{-1}D^{-1}\left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)E^{-1} \leq I.$$

By making use of Schur's complement,

$$\begin{aligned}
&\begin{bmatrix} I & E^{-T}\left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)^T \\ * & D\Lambda\Lambda^T D^T \end{bmatrix} \geq 0,\\
&\begin{bmatrix} I & E^{-T}\left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)^T \\ * & \tilde{\Lambda}\tilde{\Lambda}^T \end{bmatrix} \geq 0,
\end{aligned} \tag{4.4}$$

with

$$\tilde{\Lambda} = D + \left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)E^{-1}J.$$

By taking $J = ERD^{-T}$,

$$\tilde{\Lambda}\tilde{\Lambda}^T = DD^T + \left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)R + R^T\left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)^T + \left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)RD^{-T}D^{-1}R^T\left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)^T.$$

We then use the fact that

$$\tilde{\Lambda}\tilde{\Lambda}^T \geq DD^T + \left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right) R + R^T \left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)^T,$$

to write the following sufficient condition to (4.4)

$$\begin{bmatrix} I & E^{-T}\left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)^T \\ * & DD^T + \left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right) R + R^T \left(\sum_{j=1}^{r} h_i(\mathbf{v})\hat{L}_i\right)^T \end{bmatrix} \geq 0. \tag{4.5}$$

Another condition imposed by this uncertainty representation model is that $JJ^T \leq I$, which can be rewritten as

$$\begin{bmatrix} I & ER \\ * & DD^T \end{bmatrix} \geq 0,$$

which, by making use of Schur's Complement, is equivalent to

$$\begin{bmatrix} E^{-1}E^{-T} & R \\ * & DD^T \end{bmatrix} \geq 0. \tag{4.6}$$

By noting that

$$E^{-1}E^{-T} \geq E^{-1} + E^{-T} - I,$$

we can write the following sufficient condition to (4.6)

$$\begin{bmatrix} E^{-1} + E^{-T} - I & R \\ * & DD^T \end{bmatrix} \geq 0. \tag{4.7}$$

It then suffices to find matrices E, D and R satisfying equations (4.5) and (4.7). We may then state the following result.

Theorem 4.3

Given a polytopic model

$$\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i,$$
$$\sum_{i=1}^{r} h_i(\mathbf{v}) = 1, \quad h_i(\mathbf{v}) \geq 0,$$

the structured linear fractional uncertain model

$$\begin{aligned} &D\Delta E, \\ &\Delta = F\left[I - JF\right]^{-1} \\ &||F||_2 \leq 1, \end{aligned}$$

is such that

$$\sum_{i=1}^{r} h_i(\mathbf{v})\hat{L}_i \subset D\Delta E,$$

if the LMIs

$$\begin{aligned} Y = Y^T > 0, \\ \sum_{j=1}^{r} h_i(\mathbf{v}) \begin{bmatrix} I & X^T\hat{L}_i^T \\ * & Y + \hat{L}_i R + R^T\hat{L}_i^T \end{bmatrix} \geq 0, \\ \begin{bmatrix} X + X^T - I & R \\ * & Y \end{bmatrix} \geq 0, \end{aligned}$$

are feasible with $DD^T = Y$, $E = X^{-1}$, *and* $J = ERD^{-T}$.

From the theorem, we see that a way to search for a "small" uncertainty representation is through the optimization problem

$$\begin{aligned} \min \quad & \operatorname{tr}(Y) \\ s.t. \quad & Y = Y^T > 0 \\ & \sum_{j=1}^{r} h_i(\mathbf{v}) \begin{bmatrix} I & X^T\hat{L}_i^T \\ * & Y + \hat{L}_i R + R^T\hat{L}_i^T \end{bmatrix} \geq 0 \\ & \begin{bmatrix} X + X^T - I & R \\ * & Y \end{bmatrix} \geq 0 \end{aligned}$$

and matrices E, D and J are recovered by

$$\begin{aligned} E &= X^{-1}, \\ DD^T &= Y, \\ J &= ERD^{-T}. \end{aligned}$$

4.3 Numerical Example

In this section we apply the uncertainty modelling presented above to find a controller for a TS fuzzy model with a reduced number of rules that will still work for the original TS model.

We consider the TORA benchmark system (Baranyi et al. 2006; Bupp, Bernstein, and Coppola 1998; Petres, Baranyi, and Hashimoto 2010; Petres et al. 2007), described by equations

$$\dot{\mathbf{x}} = A(\mathbf{x})\mathbf{x} + B(\mathbf{x})u,$$

with $\mathbf{x}$ the system states, u the input signal, and

$$A(\mathbf{x}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-1}{1-\varepsilon^2\cos^2 x_3} & 0 & 0 & \frac{\varepsilon x_4 \sin x_3}{1-\varepsilon^2\cos^2 x_3} \\ 0 & 0 & 0 & 1 \\ \frac{\varepsilon \cos x_3}{1-\varepsilon^2\cos^2 x_3} & 0 & 0 & \frac{-\varepsilon^2 x_4 \cos x_3 \sin x_3}{1-\varepsilon^2\cos^2 x_3} \end{bmatrix},$$

$$B(\mathbf{x}) = \begin{bmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1-\varepsilon^2\cos^2 x_3} \\ 0 \\ \frac{1}{1-\varepsilon^2\cos^2 x_3} \end{bmatrix}.$$

We find an approximate model by means of the TP model transformation sampling step. We consider that $x_3 \in [-\pi/4, \pi/4]$, $x_4 \in [-0.5, 0.5]$ and we use a sampling grid with 25 samples in x_3 and 25 samples in x_4 and triangular shaped membership functions. These considerations lead us to a tensor $\mathcal{S} \in \mathbb{R}^{25\times25\times4\times5}$ when approximating the matrix function $S(\mathbf{x}) = [A(\mathbf{x})B(\mathbf{x})]$.

When applying the HOSVD to this tensor, we find $\sigma_1^{(1)} = 51.2576, \sigma_2^{(1)} = 0.7258, \sigma_3^{(1)} = 0.6903, \sigma_4^{(1)} = 0.0111$ and $\sigma_5^{(1)} = 0.0074$ as singular values for the x_3 direction, and $\sigma_1^{(2)} = 51.263$ and $\sigma_2^{(2)} = 0.69043$ as singular values for the x_4 direction.

We decide to keep only the first singular value in the x_3 and x_4 directions. Since we are discarding nonzero singular values, this rule reduction will generate an uncertain model. Ideally, that would lead us to an uncertain system with a single rule. However, when applying the convex hull manipulation step to the weight matrices, we were forced to add an extra column to the x_3 weight matrix in order to attain the SN property. After imposing the CNO property the weight matrices, we arrive at an uncertain TS model with 2 rules.

The reduced order model is represented by

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1.0416 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0.20898 & 0 & 0 & 0 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 0 \\ -0.20898 \\ 0 \\ 1.0416 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1.0202 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0.14551 & 0 & 0 & 0 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ -0.14551 \\ 0 \\ 1.0202 \end{bmatrix}$$

and membership functions presented in Figure 4.1.

In addition, by making use of the LMI conditions presented in the previous section, we are able to find a norm bounded uncertainty that, when used in conjunction with the reduced TS model, is capable of covering the TS model obtained in the sampling step. This uncertainty can be represented by

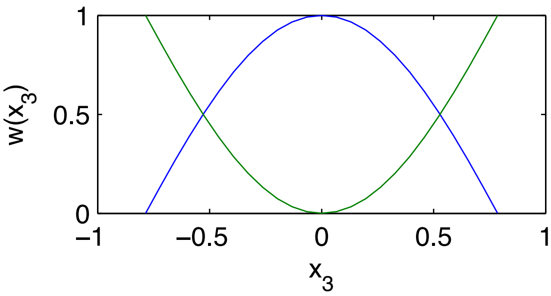

Figure 4.1: Membership functions found for the TORA system with a reduced number of rules.

$\Delta [A|B] = D\Delta E$, with

$$D = \begin{bmatrix} 0.69314 & 0 & 0 & 0 \\ 0 & 0.9999 & 0 & 0 \\ 0 & 0 & 0.69295 & 0 \\ 0 & 0 & 0 & 0.9999 \end{bmatrix},$$

$$E = \begin{bmatrix} 0.003407 & 0 & 0 & 0 & -0.0011 \\ 0 & 0.003162 & 0 & 0 & 0 \\ 0 & 0 & 0.003162 & 0 & 0 \\ 0 & 0 & 0 & 0.073518 & 0 \\ 0 & 0 & 0 & 0 & 0.0097257 \end{bmatrix}.$$

By following similar steps to ones deriving the LMI conditions in (Tanaka and Wang 2001, chapter 5), we know that a robustly stabilizing controller

$$\mathbf{u} = \sum_{i=1}^{2} w_i(x_3) K_i \mathbf{x}$$

can be found by solving the LMI feasibility problem

$$P > 0, \qquad \sum_{i=1}^{2}\sum_{j=1}^{2} w_i(x_3) w_j(x_3) \begin{bmatrix} A_i P + P A_i^T + B_i S_j + S_j^T B_i^T + D D^T & P E_a^T + S_j^T E_b^T \\ * & -I \end{bmatrix} < 0,$$

with E_a the first 4 columns of E and E_b the last column, and the controller gains recovered by $K_i = S_i P^{-1}$.

In the case of this example, we have found the gains

$$K_1 = \begin{bmatrix} 1.5565 & -0.63924 & -1.6553 & -2.1105 \end{bmatrix},$$
$$K_2 = \begin{bmatrix} 1.6325 & -0.61135 & -1.6619 & -2.1293 \end{bmatrix}$$

for the system. Finally, the system's closed loop behaviour with initial conditions $[0.1, 0.1, 0.1, 0.1]^T$ and the control signal are presented in Figure 4.2.

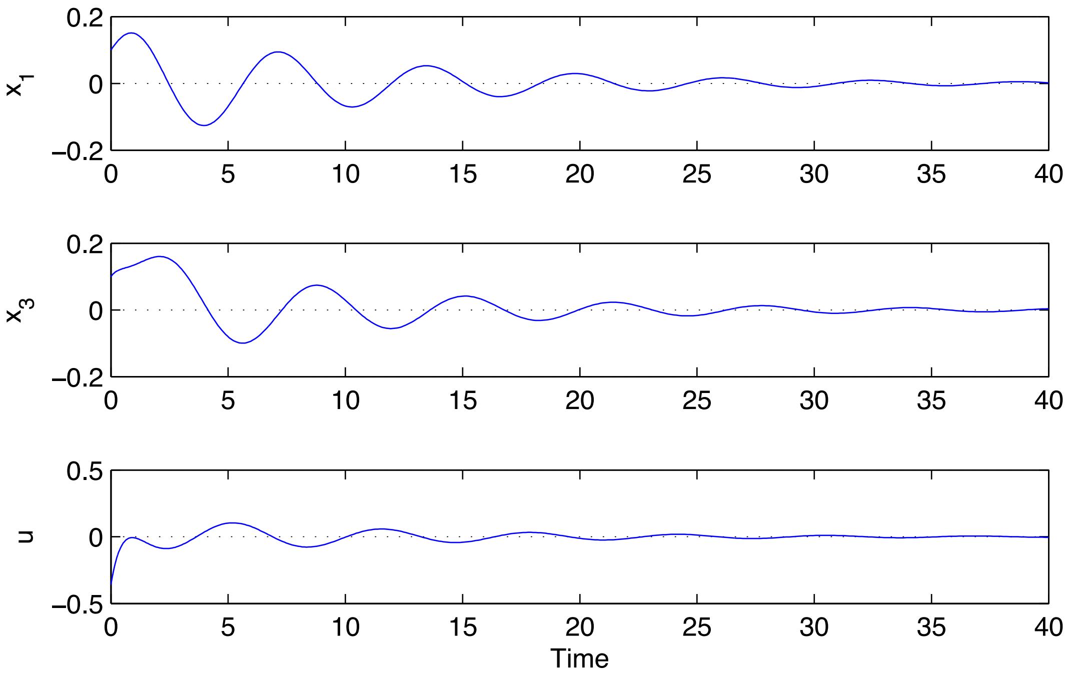

Figure 4.2: Evolution of the states and input signal through time for the closed loop TORA system.

Part II

Premise variables are state variables

5 New stability analysis conditions using piecewise fuzzy Lyapunov functions

As it is known in the literature (Sala 2009), LMI conditions for TS systems have many sources of conservativeness. In this chapter, we try to deal with two of these sources: the fact that, in general, the conditions are independent of the membership functions shape, and the choice of the family of candidate Lyapunov functions to be used.

The presentation and the results in this chapter were based on (Campos et al. 2013).

5.1 Introduction

Recent results in the literature have shown how to relax fuzzy summation conditions by incorporating the ordering relations of the membership functions by means of local transformations (Bernal, Guerra, and Kruszewski 2009). Taking these results as motivation, we propose the use of local transformations according to the shape of the membership functions such that the corresponding LMIs need to be valid for a smaller class of systems, thus reducing the conservativeness of the conditions. In addition, it is important to note that the search for these transformations can be done in a way that does not increase the number of rules of the corresponding local fuzzy model (*e.g.* by making use of the transformations from the tensor product model transformation literature presented in section 2.4).

Regarding the choice of the family of candidate Lyapunov functions, despite the use of a common quadratic Lyapunov function usually leading to simpler conditions, they are inherently conservative since they are sufficient for a larger class of systems (Johansson, Rantzer, and Arzen 1999). With that in mind, different families of candidate Lyapunov families have been proposed for TS systems (*e.g.* fuzzy Lyapunov functions and piecewise quadratic Lyapunov functions).

Fuzzy Lyapunov functions (Bernal and Guerra 2010; Faria, Silva, and Oliveira 2012; Guedes et al. 2013; Guerra et al. 2012; Ko and Park 2012; Ko, Lee, and Park 2012; Lee, Park, and Joo 2012; Mozelli, Palhares, and Avellar 2009; Mozelli et al. 2009; Rhee and Won 2006; Tanaka, Hori, and Wang 2003; Tognetti, Oliveira, and Peres 2011; Zhang and Xie 2011) are those composed by the fuzzy mixture of quadratic Lyapunov functions, using the same membership functions as the system. Some of these works (Faria, Silva, and Oliveira 2012; Guedes et al. 2013; Ko and Park 2012; Ko, Lee, and Park 2012; Lee, Park, and Joo 2012; Mozelli et al. 2009; Tanaka, Hori, and Wang 2003; Tognetti, Oliveira, and Peres 2011; Zhang and Xie 2011) make use of bounds over the time derivative of the membership functions, others (Mozelli, Palhares, and Avellar 2009; Rhee and Won 2006) impose a special structure over the parameter matrices in order to obtain conditions that are independent of these bounds, still others (Bernal and Guerra 2010; Guerra et al. 2012) make use of bounds over the partial derivatives of the membership functions and generate a set of conditions that guarantee the time derivative bound will be respected in a local region.

Piecewise quadratic Lyapunov functions (Bernal, Guerra, and Kruszewski 2009; Chen et al. 2009; Feng et al. 2005; Johansson, Rantzer, and Arzen 1999; Zhang, Li, and Liao 2005) divide the system into regions and make use of a different quadratic Lyapunov functions for each region. Some additional considerations over each of these functions need to be made in order for their combination to generate a proper candidate Lyapunov function.

In this chapter, two different piecewise fuzzy Lyapunov functions are proposed, depending on whether or not bounds on the time derivatives of the membership functions are known, together with the use of local transformations of the membership functions according to their image space distribution (instead of using their ordering relations as in Bernal, Guerra, and Kruszewski 2009).

The main contribution of this chapter is the fact that this nontrivial combination of strategies, in the sense that new theorems had to be demonstrated in order to prove these methodologies, has shown itself to be very interesting in the stability analysis of TS fuzzy models. Specifically, by applying the proposed methodologies to a standard exampled found in many recent works in the literature (Faria, Silva, and Oliveira 2012; Guedes et al. 2013; Ko and Park 2012; Ko, Lee, and Park 2012; Lee, Park, and Joo 2012; Mozelli, Palhares, and Avellar 2009; Rhee and Won 2006; Tognetti, Oliveira, and Peres 2011; Zhang and Xie 2011), significant improvement was noticed, as presented in section 5.4.

Throughout this chapter, we consider TS models whose premise variables are state variables. Specifically, we consider models that can be represented like the one in equation (2.5):

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{x})\overline{A}_i\overline{\mathbf{x}},$$

in which

$$\overline{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \qquad \overline{A}_i = \begin{bmatrix} A_i & \mathbf{a}_i \\ 0 & 0 \end{bmatrix},$$

and relationships (2.1), (2.2) and (2.3) are valid.

5.2 Local Transformation Matrices

The main motivation of the transformation matrices presented in (Campos et al. 2013) is finding a transformation matrix that reduces the conservativeness of the conditions, by means of lemma 2.5, but without increasing the number of conditions (differently from what happens when we make use of the transformation matrices proposed in Bernal, Guerra, and Kruszewski 2009, that always increase the number of conditions).

The main point behind this reduction is the fact that, most of the LMI conditions presented in the literature prove some property (like stability) for all the systems that can be represented by the local linear models of the TS model. In that way, this property is not proven only for the desired system, but for a whole family of system, and reducing the “size” of this family leads to a reduction in the conservativeness of the conditions.

As stated previously, in this chapter we consider TS systems that can be written with a tensor product structure. This class of systems covers those found using the sector nonlinearity approach (Tanaka and Wang 2001), as well as those found by making use of the tensor product model transformation technique (Baranyi 2004).

As presented in section 2.5, whenever models with a tensor product structure are used, the transformations of the normalized premise membership functions (which are functions of one variable) can be carried out separately and combined later to generate a transformation of the membership functions $h_i(\mathbf{x})$ according to equation (2.13).

In addition to this, in this chapter, the CNO transformation (proposed in (Campos et al. 2013) and presented in section 2.4) is used to find the transformation matrices of the normalized premise membership functions.

Note that, the way we calculate the transformation matrices proposed in this work is not capable of dealing directly with functions (since the CNO transformation was proposed to deal with a point cloud instead of a function). Therefore, just like in chapter 3, the membership functions we would like to transform are sampled and represented as points in the image space. In this chapter, just like in the original work presented in (Campos et al. 2013), we simply consider that we have a large enough number of sample points and disregard the approximation error between the original functions and their piecewise constant approximations to find the transformations. However, this approximation error could be taken into consideration by making use of the error polytope from section 2.3 in the same way it was considered for the *Grouping the membership points* partition strategy in chapter 3.

The method used to obtain the G^q transformation matrices is summarized in Algorithm 1. Note that this algorithm makes use of equation (2.21) instead of equation (2.13) due to the way the K_j matrices are defined in the CNO transformation in equation (2.9).

Algorithm 1: Algorithm used to calculate the local G^q transformation matrices using the CNO approach.

for all *premise variables of the model* **do**

-Sample the normalized premise variable membership functions $\mu_j^{\alpha_{ij}}(x_j)$ inside of the desired region and use each point (or the vertices of each point polytope) as rows in the U_j;

-Solve the optimization problem:

$$\begin{aligned} \min_{T_j} \quad & -\ln(\det(T_j)) \\ \text{s.t.} \quad & \text{sum}(U_j T_j) = \mathbf{1}, \\ & U_j T_j \succeq 0. \end{aligned}$$

-Find matrix $K_j = T_j^{-1}$;

endFor

-Calculate the local transformation $G^q = (K_1 \otimes \cdots \otimes K_n)^T$;

Remark 5.1

As previously stated, contrarily to the transformation matrices proposed in (Bernal, Guerra, and Kruszewski 2009). the methodology presented in this section *does not increase the number of rules* of the transformed systems, and therefore does not incur in an increase of the conditions generated when using lemma 2.5.

Remark 5.2

It is worth noting that, when we are dealing with tensor product models whose premise variables possess only two fuzzy sets, like those obtained via sector nonlinearity (Tanaka and Wang 2001) if we only have at most one nonlinearity for each state, the convex hull of the sample points (or the uncertainty vertices) is always a simplex with 2 vertices. In this case, finding the K_j matrices reduces to finding the extrema of these points, and the optimization problem in algorithm 1, albeit costly, can be seen as an alternative to doing this search.

Remark 5.3

Despite the discussion and the algorithm presented for the local transformation matrices have been directed towards models with a tensor product structure, they can also be applied to find a transformation when that is not the case. In that case, we search directly for the G^q matrices, instead of the K_j matrices, and samples of the $h_i(\mathbf{x})$ functions, instead of the $\mu_j^{\alpha_{ij}}(x_j)$ functions. Note however that, in order to find an approximation of these functions, we usually need a much larger number of samples, leasing to a possibly costlier optimization problem.

5.3 Relaxed Stability Conditions

In order to make the most out of the relaxations introduced by the transformations discussed in the previous section, it is interesting to apply them over local regions (as suggested by the name of the previous section) instead of the whole state space. In this case, an interesting methodology (similar to the one presented in (Bernal, Guerra, and Kruszewski 2009)) is to use piecewise Lyapunov functions. In that regard, we propose the use of two different piecewise fuzzy Lyapunov functions (leading to two different sets of LMI stability conditions) together with the local transformation matrices from the previous section. Note as well that the results presented in this section could be used regardless of the transformations, but are more relaxed when the transformations are considered.

In order to apply the new stability conditions, the state space is partitioned into polyhedral regions Ω_q, $q = 1, 2, \ldots, n_q$, such that the following assumptions are valid.

Assumption 5.1

In each Ω_q region, the relation

$$\overline{E}^q \overline{\mathbf{x}} \succeq 0, \quad \forall\, \overline{\mathbf{x}} \in \Omega_q, \tag{5.1}$$

is valid, and $\overline{E}^q = \begin{bmatrix} E^q & \mathbf{e}^q \end{bmatrix}$, with $\mathbf{e}^q = 0$ for regions that include the origin of the state space. In addition, given two neighboring regions Ω_q and Ω_p, we have that

$$\overline{F}^q \overline{\mathbf{x}} = \overline{F}^p \overline{\mathbf{x}}, \quad \forall\, \overline{\mathbf{x}} \in \Omega_q \cap \Omega_p, \tag{5.2}$$

in which $\overline{F}^q = \begin{bmatrix} F^q & \mathbf{f}^q \end{bmatrix}$, with $\mathbf{f}^q = 0$ for regions that include the origin.

It is important to note that, given the vertices of each polyhedral region, matrices $\overline{E}^q$, $\overline{F}^q$ and $\overline{F}^p$ from Assumption 5.1 can be found using the procedures described in (Johansson, Rantzer, and Arzen 1999).

In a different way, the $\overline{E}^q$ are easily found by stacking the inequalities that describe the region, and the $\overline{F}^q$ matrices can be found by solving a set of homogeneous linear equations for each state variable.

Assumption 5.2

In each Ω_q region there are $G^q \in \mathbb{R}^{r\times\tilde{r}_q}$ matrices describing local transformations of the $h_i(\mathbf{x})$ membership functions such that

$$\mathbf{h} = G^q\tilde{\mathbf{h}}^q,$$

in which $\mathbf{h}$ is the vector formed by stacking the $h_i(\mathbf{x})$, $i \in 1,\dots,r$ membership functions, and $\tilde{\mathbf{h}}$ is the vector formed by stacking the $\tilde{h}_j^q(\mathbf{x})$, $j \in 1,\dots,\tilde{r}_q$ membership functions. In addition, the transformed membership functions are such that (2.1), (2.2) and (2.3) are still valid.

Note that, if this is not the case, we can take G^q to be the $r \times r$ identity matrix.

Besides the transformation matrices discussed in the previous section, an alternative, from (Bernal, Guerra, and Kruszewski 2009), to find transformed membership functions that satisfy Assumption 5.2 is found by splitting the state space such that the possible ordering relations among the membership functions can be taken into consideration. As a consequence of this procedure, it is possible to find G^q matrices with the additional property that $G^q \succeq 0$.

Membership derivative independent conditions

For each Ω_q region, it is possible to write a fuzzy candidate Lyapunov function (Rhee and Won 2006) of the form:

$$V^q(\overline{\mathbf{x}}) = 2\int_{\Gamma[0,\overline{\mathbf{x}}]} \langle \sum_{i=1}^{r} h_i(\mathbf{s})\overline{P}_i^q\overline{\mathbf{s}}, d\overline{\mathbf{s}}\rangle, \tag{5.3}$$

in which $\Gamma[0,\overline{\mathbf{x}}]$ is an arbitrary trajectory from the origin to $\overline{\mathbf{x}}$, with $\overline{\mathbf{s}} \in \mathbb{R}^{n+1}$ being the trajectory variable and $\mathbf{s} \in \mathbb{R}^n$ equal to the first n components of $\overline{\mathbf{s}}$.

The $\overline{P}_i^q \in \mathbb{R}^{(n+1)\times(n+1)}$ matrices parametrize the candidate Lyapunov function, and in order for $V^q(\overline{\mathbf{x}})$ to be similar to potential energy functions, such that the integral in (5.3) is path independent, it is sufficient that the $\overline{P}_i^q$ matrices be symmetric and have the following structure (Rhee and Won 2006):

$$\begin{aligned}
\overline{P}_i^q &= \begin{bmatrix} P_i^q & \mathbf{p}_1^q \\ * & p_2^q \end{bmatrix}, \\
P_i^q &= D_0^q + D_i^q, \\
D_0^q &= \begin{bmatrix} 0 & d_{12q} & \cdots & d_{1nq} \\ d_{12q} & 0 & \cdots & d_{2nq} \\ \vdots & \vdots & \ddots & \vdots \\ d_{1nq} & d_{2nq} & \cdots & 0 \end{bmatrix}, \\
D_i^q &= \begin{bmatrix} d_{11q}^{\alpha_{i1}} & 0 & \cdots & 0 \\ 0 & d_{22q}^{\alpha_{i2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nnq}^{\alpha_{in}} \end{bmatrix}.
\end{aligned} \tag{5.4}$$

The structure in (5.4) indicates that, in each region Ω_q, the elements of P_i^q outside of the diagonal, and the elements $\mathbf{p}_1^q$ and p_2^q have to be the same for all of the rules. Nevertheless, the diagonal elements of

P_i^q, $d_{jjq}^{\alpha_{ij}}$, may change according to the rules, but need to be the same for two rules sharing the same fuzzy set regarding variable x_j; *i.e.* given two rules i and k, if $\alpha_{ij} = \alpha_{kj}$, then $d_{jjq}^{\alpha_{ij}} = d_{jjq}^{\alpha_{kj}}$.

For regions Ω_q that contain the origin, $\mathbf{p}_1^q = 0$ and $p_2^q = 0$, and equation (5.3) can be rewritten as

$$V^q(\bar{\mathbf{x}}) = 2\int_{\Gamma[0,\mathbf{x}]} \langle \sum_{i=1}^{r} h_i(\mathbf{s}) P_i^q \mathbf{s}, \mathbf{ds} \rangle, \tag{5.5}$$

with the P_i^q matrices following the structure presented in equation (5.4). Having made these assumptions, the following theorem can be stated.

Theorem 5.1: Membership functions' time derivative independent stability conditions

Given matrices $\overline{E}^q = \begin{bmatrix} E^q & \mathbf{e}^q \end{bmatrix}$ *and* $\overline{F}^q = \begin{bmatrix} F^q & \mathbf{f}^q \end{bmatrix}$ *satisfying conditions* (5.1) *and* (5.2)*, together with matrices* $G^q \in \mathbb{R}^{r \times \tilde{r}_q}$ *describing membership functions transformations such that Assumption 5.2 is valid, system* (2.5) *is asymptotically stable if there are matrices* $\overline{M} = \begin{bmatrix} M & \mathbf{m}_1 \\ \mathbf{m}_2^T & m_3 \end{bmatrix}$ *and* $\overline{N} = \begin{bmatrix} N & \mathbf{n}_1 \\ \mathbf{n}_2^T & n_3 \end{bmatrix}$*, symmetric matrices* $U_i^q \succeq 0$ *and* $W_i^q \succeq 0$*, and symmetric matrices* T_i*, such that the following conditions are met:*

$$\overline{P}_i^q - (\overline{E}^q)^T U_i^q \overline{E}^q > 0, \tag{5.6}$$

$$\sum_{i=1}^{r} g_{ij}^q \overline{L}_{iq} < 0, \quad j \in 1, \ldots, \tilde{r}_q, \tag{5.7}$$

with $\overline{P}_i^q = (\overline{F}^q)^T T_i \overline{F}^q$ *having the structure in* (5.4)*, and*

$$\overline{L}_{iq} = \begin{bmatrix} \overline{M}^T \overline{A}_i + \overline{A}_i^T \overline{M} + (\overline{E}^q)^T W_i^q \overline{E}^q & \overline{P}_i^q - \overline{M}^T + \overline{A}_i^T \overline{N} \\ * & -\overline{N} - \overline{N}^T \end{bmatrix}, \tag{5.8}$$

for regions that do not contain the origin of the state space. In a similar way,

$$P_i^q - (E^q)^T U_i^q E^q > 0, \tag{5.9}$$

$$\sum_{i=1}^{r} g_{ij}^q L_{iq} < 0, \quad j \in 1, \ldots, \tilde{r}_q, \tag{5.10}$$

with $P_i^q = (F^q)^T T_i F^q$ *having the structure in* (5.4)*, and*

$$L_{iq} = \begin{bmatrix} M^T A_i + A_i^T M + (E^q)^T W_i^q E^q & P_i^q - M^T + A_i^T N \\ * & -N - N^T \end{bmatrix}, \tag{5.11}$$

for regions that contain the origin of the state space.

Proof. Consider that the state space is divided into Ω_q polyhedral regions described by (5.1). If we would like to use a piecewise combination of (5.3) and (5.5) as a candidate Lyapunov function, we must guarantee the continuity of this combination when we switch from one region to the other. Therefore, we would like that

$$V^q(\bar{\mathbf{x}}) = V^p(\bar{\mathbf{x}}),$$
$$2\int_{\Gamma[0,\bar{\mathbf{x}}]} \langle \sum_{i=1}^{r} h_i(\mathbf{s}) \overline{P}_i^q \mathbf{s}, \mathbf{ds} \rangle = 2\int_{\Gamma[0,\bar{\mathbf{x}}]} \langle \sum_{i=1}^{r} h_i(\mathbf{s}) \overline{P}_i^p \mathbf{s}, \mathbf{ds} \rangle, \tag{5.12}$$

be true for two neighboring regions Ω_q and Ω_p, whenever $\bar{\mathbf{x}}$ is on the border of these two regions.

Since the structure imposed by equation (5.4) over the $\overline{P}_i^q$ and P_i^q matrices guarantees that the line integral is path independent (Rhee and Won 2006), choosing the path $\bar{\mathbf{s}} = \tau\bar{\mathbf{x}}$, with $\tau \in [0,1]$, condition (5.12) can be rewritten as

$$\begin{aligned}\int_0^1 \tau\bar{\mathbf{x}}^T \sum_{i=1}^r h_i(\tau\mathbf{x})\overline{P}_i^q\bar{\mathbf{x}}d\tau &= \int_0^1 \tau\bar{\mathbf{x}}^T \sum_{i=1}^r h_i(\tau\mathbf{x})\overline{P}_i^p\bar{\mathbf{x}}d\tau,\\ \int_0^1 \tau \sum_{i=1}^r h_i(\tau\mathbf{x})\bar{\mathbf{x}}^T\overline{P}_i^q\bar{\mathbf{x}}d\tau &= \int_0^1 \tau \sum_{i=1}^r h_i(\tau\mathbf{x})\bar{\mathbf{x}}^T\overline{P}_i^p\bar{\mathbf{x}}d\tau. \end{aligned} \tag{5.13}$$

A sufficient condition for (5.13) is that

$$\sum_{i=1}^r h_i(\tau\mathbf{x})\bar{\mathbf{x}}^T\overline{P}_i^q\bar{\mathbf{x}} = \sum_{i=1}^r h_i(\tau\mathbf{x})\bar{\mathbf{x}}^T\overline{P}_i^p\bar{\mathbf{x}},$$

which, in turn, can be assured by

$$\bar{\mathbf{x}}^T\overline{P}_i^q\bar{\mathbf{x}} = \bar{\mathbf{x}}^T\overline{P}_i^p\bar{\mathbf{x}}, \tag{5.14}$$

whenever $\bar{\mathbf{x}}$ is on the intersection of regions Ω_q and Ω_p.

Assuming that there are $\overline{F}^q$ matrices such that, whenever $\bar{\mathbf{x}}$ lies on the switching surface of the two regions, condition (5.2) is valid, matrices $\overline{P}_i^q$ can be written as

$$\overline{P}_i^q = (\overline{F}^q)^T T_i \overline{F}^q$$

to satisfy condition (5.14). By doing this,

$$\bar{\mathbf{x}}^T(\overline{F}^q)^T T_i \overline{F}^q\bar{\mathbf{x}} = \bar{\mathbf{x}}^T(\overline{F}^p)^T T_i \overline{F}^p\bar{\mathbf{x}}.$$

On the other hand, the condition that the candidate Lyapunov function must be positive definite implies that, over region Ω_q,

$$\bar{\mathbf{x}}^T\overline{P}_i^q\bar{\mathbf{x}} > 0, \quad \text{for regions not including the origin,} \tag{5.15}$$

$$\mathbf{x}^T P_i^q\mathbf{x} > 0, \quad \text{for regions including the origin.} \tag{5.16}$$

The inequalities above need be satisfied only when $\mathbf{x} \in \Omega_q$; *i.e.* only when expression (5.1) is true. In order to deal with this problem, the following quadratic constraints are considered

$$\bar{\mathbf{x}}^T(\overline{E}^q)^T U_i^q \overline{E}^q\bar{\mathbf{x}} \geq 0, \tag{5.17}$$

$$\mathbf{x}^T(E^q)^T U_i^q E^q\mathbf{x} \geq 0, \tag{5.18}$$

which, from (5.1) and picking *copositive* U_i^q matrices, are satisfied for every $\mathbf{x} \in \Omega_q$. Note that it is possible that these inequalities are also valid for some $\mathbf{x} \notin \Omega_q$; *i.e.* the inequalities above are only sufficient for (5.15) and (5.16).

In order to simplify the problem, we use non-negative matrices (matrices whose elements are non-negative), which is a subset of the *copositive* matrices (Bundfuss 2009).

Therefore, a sufficient condition to ensure that the candidate Lyapunov function is positive definite over each region Ω_q is that (5.15)) and (5.16) be valid whenever (5.17) and (5.18), respectively, are valid. And by applying the S-procedure we get conditions (5.6) and (5.9).

In regions that do not include the origin, the time derivative of the Lyapunov function can be written as

$$\dot{V}(\bar{\mathbf{x}}) = \begin{bmatrix}\bar{\mathbf{x}}\\ \dot{\bar{\mathbf{x}}}\end{bmatrix}^T \sum_{i=1}^r h_i(\mathbf{x}) \begin{bmatrix}0 & \overline{P}_i^q\\ * & 0\end{bmatrix} \begin{bmatrix}\bar{\mathbf{x}}\\ \dot{\bar{\mathbf{x}}}\end{bmatrix}, \tag{5.19}$$

whereas in regions that include the origin, it can be written as

$$\dot{V}(\bar{\mathbf{x}}) = \begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix}^T \sum_{i=1}^{r} h_i(\mathbf{x}) \begin{bmatrix}0 & P_i^q\\ * & 0\end{bmatrix} \begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix}. \tag{5.20}$$

Consider arbitrary real matrices $\overline{M} = \begin{bmatrix}M & \mathbf{m}_1\\ \mathbf{m}_2^T & m_3\end{bmatrix}$ and $\overline{N} = \begin{bmatrix}N & \mathbf{n}_1\\ \mathbf{n}_2^T & n_3\end{bmatrix}$. The following null-term (Mozelli, Palhares, and Avellar 2009)

$$\begin{bmatrix}\bar{\mathbf{x}}\\ \dot{\bar{\mathbf{x}}}\end{bmatrix}^T \sum_{i=1}^{r} h_i(\mathbf{x}) \begin{bmatrix}\overline{M}^T\overline{A}_i + \overline{A}_i^T\overline{M} & -\overline{M}^T + \overline{A}_i^T\overline{N}\\ * & -\overline{N} - \overline{N}^T\end{bmatrix} \begin{bmatrix}\bar{\mathbf{x}}\\ \dot{\bar{\mathbf{x}}}\end{bmatrix} = 0, \tag{5.21}$$

can be added to the time derivative in regions that do not include the origin. And the null-term

$$\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix}^T \sum_{i=1}^{r} h_i(\mathbf{x}) \begin{bmatrix}M^T A_i + A_i^T M & -M^T + A_i^T N\\ * & -N - N^T\end{bmatrix} \begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} = 0, \tag{5.22}$$

in regions that include the origin.

By making use of quadratic constraints, similar to the one presented above, to represent regions that contain each region Ω_q, from (5.1), and using W_i^q matrices whose elements are non-negative, we can write

$$\begin{bmatrix}\bar{\mathbf{x}}\\ \dot{\bar{\mathbf{x}}}\end{bmatrix}^T \sum_{i=1}^{r} h_i(\mathbf{x}) \begin{bmatrix}(\overline{E}^q)^T W_i^q \overline{E}^q & 0\\ * & 0\end{bmatrix} \begin{bmatrix}\bar{\mathbf{x}}\\ \dot{\bar{\mathbf{x}}}\end{bmatrix} \geq 0,$$
$$\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix}^T \sum_{i=1}^{r} h_i(\mathbf{x}) \begin{bmatrix}(E^q)^T W_i^q E^q & 0\\ * & 0\end{bmatrix} \begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} \geq 0.$$

Therefore, adding each term to the respective time derivative in (5.19) and (5.20), we assure that the time derivative of the Lyapunov function is negative definite over regions Ω_q, and at the same time, allow for a relaxation of this constraint outside of these regions (where this conditions is not really necessary). The final expression can be written as

$$\sum_{i=1}^{r} h_i(\mathbf{x})\overline{L}_{iq} < 0, \quad \text{for regions not including the origin,}$$
$$\sum_{i=1}^{r} h_i(\mathbf{x}) L_{iq} < 0, \quad \text{for regions including the origin,}$$

in which $\overline{L}_{iq}$ and L_{iq} are given by (5.8) and (5.11), respectively.

If there are $G^q \in \mathbb{R}^{r\times\tilde{r}_q}$ matrices defining valid transformations of the membership functions according to Assumption 5.2, we can make use of lemma 2.5 in the conditions above to write

$$\sum_{j=1}^{\tilde{r}_q} \tilde{h}_j^q \left(\sum_{i=1}^{r} g_{ij}^q \overline{L}_{iq}\right) < 0, \tag{5.23}$$

for regions not including the origin,

$$\sum_{j=1}^{\tilde{r}_q} \tilde{h}_j^q \left(\sum_{i=1}^{r} g_{ij}^q L_{iq}\right) < 0, \tag{5.24}$$

for regions including the origin.

And it is clear that conditions (5.7) and (5.10) are sufficient for (5.23) and (5.24). □

Remark 5.4

Since the $\overline{F}^q$ matrices change according to each Ω_q polyhedral region considered, a generic way of imposing structure (5.4) to matrices $\overline{P}_i^q$ is by means of linear equality constraints (since, in this case, the decision variables are the T_i matrices instead of the $\overline{P}_i^q$ matrices). However, it is possible to remove the equality constraints from the optimization problem by means of a reduction of the set of decision variables, which also reduces the computational time associated with the optimization. In the case here presented, these constraints (if well posed) determine a well behaved transformation basis, such that numerical problems should not occur from eliminating the constraints [a].

[a]this can be done, for instance, using YALMIP (Löfberg 2004) and asking the interpreter to reduce the decision variables based on the equality constraints.

Membership derivative dependent conditions

For each region, it is also possible to write a fuzzy candidate Lyapunov function (Tanaka, Hori, and Wang 2003):

$$V^q(\overline{\mathbf{x}}) = \sum_{i=1}^{r} h_i(\mathbf{x})\overline{\mathbf{x}}^T \overline{P}_i^q \overline{\mathbf{x}}, \tag{5.25}$$

in which $\overline{P}_i^q \in \mathbb{R}^{(n+1)\times(n+1)}$ are matrices parametrizing the Lyapunov function. In a similar way to what was previously presented, these matrices can be written as

$$\overline{P}_i^q = \begin{bmatrix} P_i^q & \mathbf{p}_{1_i}^q \\ * & p_{2_i}^q \end{bmatrix}.$$

However, unlike the previous case, there are no restrictions over the structure of the $\overline{P}_i^q$ matrices regarding the rules of the TS model. Instead, we assume that the membership functions' time derivatives are bounded such that[1]

$$|\dot{h}_i(\mathbf{x})| \leq \phi_i, \quad i = 1, 2, \ldots, r. \tag{5.26}$$

For regions containing the origin, $\mathbf{p}_{1_i}^q = 0$ and $p_{2_i}^q = 0$, and the corresponding candidate Lyapunov function can be rewritten as

$$V^q(\mathbf{x}) = \sum_{i=1}^{r} h_i(\mathbf{x})\mathbf{x}^T P_i^q \mathbf{x}. \tag{5.27}$$

Given these assumptions, the following theorem can be stated.

[1]In order to simplify the demonstration, we make use of the same bounds for every region in the state space. Note however that we are not forced to do so, and different bounds could be used for each region.

Theorem 5.2: Membership functions' time derivative dependent stability conditions

Given matrices $\overline{E}^q = \begin{bmatrix} E^q & \mathbf{e}^q \end{bmatrix}$ *and* $\overline{F}^q = \begin{bmatrix} F^q & \mathbf{f}^q \end{bmatrix}$ *satisfying conditions* (5.1) *and* (5.2)*, together with matrices* $G^q \in \mathbb{R}^{r\times\tilde{r}_q}$ *describing membership functions transformations such that Assumption 5.2 is valid, and considering that the time derivative of the membership functions is bounded according to* (5.26)*, system* (2.5) *is asymptotically stable if there are matrices* $\overline{M} = \begin{bmatrix} M & \mathbf{m}_1 \\ \mathbf{m}_2^T & m_3 \end{bmatrix}$ *and* $\overline{N} = \begin{bmatrix} N & \mathbf{n}_1 \\ \mathbf{n}_2^T & n_3 \end{bmatrix}$*, symmetric matrices* $U_i^q \succeq 0$ *and* $W_i^q \succeq 0$*, and symmetric matrices* T_i *and* $\overline{Q}^q$ *(or* Q^q*), such that the following conditions are met:*

$$\overline{P}_i^q - (\overline{E}^q)^T U_i^q \overline{E}^q > 0, \tag{5.28}$$
$$\overline{P}_i^q + \overline{Q}^q > 0, \tag{5.29}$$
$$\sum_{i=1}^{r} g_{ij}^q \overline{L}_{iq} < 0, \quad j \in 1,\ldots,\tilde{r}_q, \tag{5.30}$$

with $\overline{P}_i^q = (\overline{F}^q)^T T_i \overline{F}^q$*, and*

$$\overline{L}_{iq} = \begin{bmatrix} \overline{P}_\phi^q + \overline{M}^T\overline{A}_i + \overline{A}_i^T\overline{M} + (\overline{E}^q)^T W_i^q \overline{E}^q & \overline{P}_i^q - \overline{M}^T + \overline{A}_i^T\overline{N} \\ * & -\overline{N} - \overline{N}^T \end{bmatrix}, \tag{5.31}$$
$$\overline{P}_\phi^q = \sum_{i=1}^{r} \phi_i(\overline{P}_i^q + \overline{Q}^q),$$

for regions not containing the origin of the state space. In a similar way,

$$P_i^q - (E^q)^T U_i^q E^q > 0, \tag{5.32}$$
$$P_i^q + Q^q > 0, \tag{5.33}$$
$$\sum_{i=1}^{r} g_{ij}^q L_{iq} < 0, \quad j \in 1,\ldots,\tilde{r}_q, \tag{5.34}$$

with $P_i^q = (F^q)^T T_i F^q$*, and*

$$L_{iq} = \begin{bmatrix} P_\phi^q + M^T A_i + A_i^T M + (E^q)^T W_i^q E^q & P_i^q - M^T + A_i^T N \\ * & -N - N^T \end{bmatrix}, \tag{5.35}$$
$$P_\phi^q = \sum_{i=1}^{r} \phi_i(P_i^q + Q^q),$$

for regions containing the origin of the state space.

Proof. Consider that the state space is divided into Ω_q polyhedral regions described by (5.1). If we would like to use a piecewise combination of (5.25) and (5.27) as a candidate Lyapunov function, we must guarantee the continuity of this combination when switching from one region to another. Therefore,

$\forall \overline{\mathbf{x}} \in \Omega_q \cap \Omega_p$,

$$V^q(\overline{\mathbf{x}}) = V^p(\overline{\mathbf{x}}),$$
$$\sum_{i=1}^{r} h_i(\mathbf{x})\overline{\mathbf{x}}^T\overline{P}_i^q\overline{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{x})\overline{\mathbf{x}}^T\overline{P}_i^p\overline{\mathbf{x}}.$$

A sufficient condition for that is that

$$\overline{\mathbf{x}}^T\overline{P}_i^q\overline{\mathbf{x}} = \overline{\mathbf{x}}^T\overline{P}_i^p\overline{\mathbf{x}}, \quad \forall \overline{\mathbf{x}} \in \Omega_q \cap \Omega_p.$$

On the other hand, from (5.2), this can be guaranteed by rewriting the $\overline{P}_i^q$ matrices as

$$\overline{P}_i^q = (\overline{F}^q)^T T_i \overline{F}^q,$$

as was shown in the proof of Theorem 5.1.

Conditions (5.28) and (5.32) are sufficient to guarantee that the piecewise combination of these Lyapunov functions will be positive definite over regions Ω_q. This can be proved by following the same development done in the proof of Theorem 5.1.

In regions not including the origin, the time derivative of the Lyapunov function can be written as

$$\dot{V}(\overline{\mathbf{x}}) = \begin{bmatrix} \overline{\mathbf{x}} \\ \dot{\overline{\mathbf{x}}} \end{bmatrix}^T \sum_{i=1}^{r} h_i(\mathbf{x}) \begin{bmatrix} \sum_{k=1}^{r} \dot{h}_k(\mathbf{x})\overline{P}_k^q & \overline{P}_i^q \\ * & 0 \end{bmatrix} \begin{bmatrix} \overline{\mathbf{x}} \\ \dot{\overline{\mathbf{x}}} \end{bmatrix}, \tag{5.36}$$

whereas, for regions including the origin, it can be written as

$$\dot{V}(\overline{\mathbf{x}}) = \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix}^T \sum_{i=1}^{r} h_i(\mathbf{x}) \begin{bmatrix} \sum_{k=1}^{r} \dot{h}_k(\mathbf{x})P_k^q & P_i^q \\ * & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix}. \tag{5.37}$$

Taking into account the bounds in (5.26) over the membership functions time derivatives, and considering that conditions (5.29) and (5.33) are satisfied, we can make use of the idea presented in (Mozelli et al. 2009) to write

$$\sum_{k=1}^{r} \dot{h}_k(\mathbf{x})\overline{P}_k^q = \sum_{k=1}^{r} \dot{h}_k(\mathbf{x})(\overline{P}_k^q + \overline{Q}^q) \leq \overline{P}_\phi^q = \sum_{k=1}^{r} \phi_k(\overline{P}_k^q + \overline{Q}^q),$$
$$\sum_{k=1}^{r} \dot{h}_k(\mathbf{x})P_k^q = \sum_{k=1}^{r} \dot{h}_k(\mathbf{x})(P_k^q + Q^q) \leq P_\phi^q = \sum_{k=1}^{r} \phi_k(P_k^q + Q^q),$$

which, from (5.36) and (5.37), leads to

$$\dot{V}(\overline{\mathbf{x}}) \leq \begin{bmatrix} \overline{\mathbf{x}} \\ \dot{\overline{\mathbf{x}}} \end{bmatrix}^T \sum_{i=1}^{r} h_i(\mathbf{x}) \begin{bmatrix} \overline{P}_\phi^q & \overline{P}_i^q \\ * & 0 \end{bmatrix} \begin{bmatrix} \overline{\mathbf{x}} \\ \dot{\overline{\mathbf{x}}} \end{bmatrix},$$

for regions not including the origin, and

$$\dot{V}(\overline{\mathbf{x}}) \leq \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix}^T \sum_{i=1}^{r} h_i(\mathbf{x}) \begin{bmatrix} P_\phi^q & P_i^q \\ * & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix},$$

for regions including the origin.

By following the same steps of using the null-terms (5.21) and (5.22), and the same relaxations regarding the W_i^q variables that were used in the proof of Theorem 5.1, sufficient conditions for the time derivative of the Lyapunov function to negative definite can be written as

$$\sum_{i=1}^{r} h_i(\mathbf{x})\overline{L}_{iq} < 0, \quad \text{for regions not including the origin,} \tag{5.38}$$
$$\sum_{i=1}^{r} h_i(\mathbf{x})L_{iq} < 0, \quad \text{for regions including the origin,} \tag{5.39}$$

in which $\overline{L}_{iq}$ and L_{iq} are given by (5.31) and (5.35), respectively.

As in Theorem 5.1, considering that there exist valid G^q transformation matrices according to assumption 5.2, conditions (5.30) and (5.34) are sufficient for (5.38) and (5.39) to be true. □

Remark 5.5

It is important to note that, when dealing with piecewise Lyapunov functions, the stability will only be proven for the largest Lyapunov sublevel set that is contained in the union of all of the state space regions considered. However, if the regions are written and used as in (Johansson, Rantzer, and Arzen 1999) (which is the case in this chapter) and all of the regions considered contain the origin, the conditions will be valid for all of the state space. However, in the case of Theorem 5.2, we will only get global stability if the bound used for the time derivative of the membership functions is global. In the other cases, the stability region proven will be the largest sublevel set of the Lyapunov function that is contained in the region where the bound is valid. In case we would like to use local conditions, an interesting approach is to employ instead bounds the partial derivatives of the membership functions, as in (Bernal and Guerra 2010).

Remark 5.6

Note that Theorems 5.1 and 5.2 are more general than (Bernal, Guerra, and Kruszewski 2009, Theorem 2), if the bound used for the time derivative of the membership functions considered in Theorem 5.2 is valid in all of the regions considered. In both cases, if the $\overline{P}_i^q$ matrices are chosen to change only with q, conditions in Theorems 5.1 are 5.2 relaxed versions of (Bernal, Guerra, and Kruszewski 2009, Theorem 2).

Proof. In order to prove the remark above, we will show only the proof for the regions that contain the origin of the state space. The proof for the other regions follows by analogy.

Note that conditions from (Bernal, Guerra, and Kruszewski 2009, Theorem 2) can be written as

$$P^q - (E^q)^T U^q E^q > 0, \quad \forall q, \tag{5.40}$$

$$\sum_{i=1}^{r} h_i \left(A_i^T P^q + P^q A_i + (E^q)^T W_i^q E^q \right) < 0, \quad \forall q. \tag{5.41}$$

By taking $P_i^q = P^q$ and $U_i^q = U^q$ in (5.9) and (5.32), we recover (5.40).

By making use of Finsler's lemma (Mozelli, Palhares, and Mendes 2010), the constraint (5.41) can be written as

$$\sum_{i=1}^{r}\sum_{j=1}^{r} h_i h_j \begin{bmatrix} A_i^T M_j + M_j^T A_i + (E^q)^T W_i^q E^q & P^q - M_j^T + A_i^T N_j \\ * & -N_j - N_j^T \end{bmatrix} < 0, \quad \forall q.$$

However, note that, if the above constraint is satisfied, there are matrices M and N such that

$$\sum_{i=1}^{r} h_i L_{iq} < 0, \quad \forall q,$$

$$L_{iq} = \begin{bmatrix} A_i^T M + M^T A_i + (E^q)^T W_i^q E^q & P^q - M^T + A_i^T N \\ * & -N - N^T \end{bmatrix}.$$

Applying the local transformation matrices, we can write

$$\sum_{i=1}^{r} g_{ij}^q L_{iq} < 0, \forall q, j \in 1, \ldots, \tilde{r}_q. \tag{5.42}$$

Note that, by taking $P_i^q = P^q$ in (5.10), we recover (5.42). In addition, by taking $P_i^q = P^q$ and $Q^q = -P^q$ in (5.34), we also recover (5.42).

Therefore, the conditions in Theorems 5.1 and 5.2 will always be satisfied whenever the conditions in (Bernal, Guerra, and Kruszewski 2009, Theorem 2) are. In that regard, the conditions presented in 5.1 are more general than those presented in (Bernal, Guerra, and Kruszewski 2009, Theorem 2), and conditions in Theorem 5.2 are also more general, given a bound that is valid in all of the regions considered. □

5.4 Example

In this section, a numerical example is presented to illustrate that the proposed stability conditions give better results than previous conditions available in the literature.

Example 5.1 (Example 1 from (Rhee and Won 2006))**.** The following TS fuzzy system is considered:

Rule 1: If $x_1 \in \mathcal{M}_1^1$ and $x_2 \in \mathcal{M}_2^1$ Then $\dot{\mathbf{x}} = A_1\mathbf{x}$,

Rule 2: If $x_1 \in \mathcal{M}_1^1$ and $x_2 \in \mathcal{M}_2^2$ Then $\dot{\mathbf{x}} = A_2\mathbf{x}$,

Rule 3: If $x_1 \in \mathcal{M}_1^2$ and $x_2 \in \mathcal{M}_2^1$ Then $\dot{\mathbf{x}} = A_3\mathbf{x}$,

Rule 4: If $x_1 \in \mathcal{M}_1^2$ and $x_2 \in \mathcal{M}_2^2$ Then $\dot{\mathbf{x}} = A_4\mathbf{x}$,

with

$$A_1 = \begin{bmatrix} -5 & -4 \\ -1 & a \end{bmatrix},$$
$$A_2 = \begin{bmatrix} -4 & -4 \\ \frac{1}{5}(3b-2) & \frac{1}{5}(3a-4) \end{bmatrix},$$
$$A_3 = \begin{bmatrix} -3 & -4 \\ \frac{1}{5}(2b-3) & \frac{1}{5}(2a-6) \end{bmatrix},$$
$$A_4 = \begin{bmatrix} -2 & -4 \\ b & -2 \end{bmatrix},$$

for constants a and b. The normalized premise membership functions of $\mathcal{M}_i^1$ and $\mathcal{M}_i^2$ are given by

$$\mu_i^1(x_i) = \begin{cases} (1-\sin(x_i))/2 & \text{for } |x_i| \le \pi/2, \\ 0 & \text{for } x_i > \pi/2, \\ 1 & \text{for } x_i < -\pi/2, \end{cases}$$
$$\mu_i^2(x_i) = 1 - \mu_i^1(x_i).$$

The membership functions associated with the fuzzy rules are then written as

$$h_1(\mathbf{x}) = \mu_1^1(x_1)\mu_2^1(x_2), \qquad h_2(\mathbf{x}) = \mu_1^1(x_1)\mu_2^2(x_2),$$
$$h_3(\mathbf{x}) = \mu_1^2(x_1)\mu_2^1(x_2), \qquad h_4(\mathbf{x}) = \mu_1^2(x_1)\mu_2^2(x_2).$$

The system's stability was checked for different values of a and b by using the proposed conditions, as well as Theorems 1 and 4 from (Mozelli, Palhares, and Avellar 2009), Theorem 2 from (Bernal, Guerra, and Kruszewski 2009), Theorem 1 from (Guedes et al. 2013), Theorem 4.3 from (Faria, Silva, and Oliveira 2012), Theorem 2 from (Lee, Park, and Joo 2012), Theorem 1 from (Ko, Lee, and Park 2012), and Theorem 1 from (Tognetti, Oliveira, and Peres 2011) with $g = 3$ and $d = 3$.

In order to apply Theorem 5.1 and 5.2, as well as Theorem 2 from (Bernal, Guerra, and Kruszewski 2009), the system was divided into 4 regions according to the ordering relations of the membership functions (this was done to allow the use, for comparison purposes, of the local transformation matrices proposed in Bernal, Guerra, and Kruszewski 2009). The following regions were chosen

Region 1: $x_1 < 0, \quad x_2 < 0$
$h_1 > h_2 > h_4$ and $h_1 > h_3 > h_4$

Region 2: $x_1 < 0, \quad x_2 > 0$
$h_2 > h_4 > h_3$ and $h_2 > h_1 > h_3$

Region 3: $x_1 > 0, \quad x_2 < 0$
$h_3 > h_1 > h_2$ and $h_3 > h_4 > h_2$

Region 4: $x_1 > 0, \quad x_2 > 0$
$h_4 > h_2 > h_1$ and $h_4 > h_3 > h_1$

The following E^q and F^q matrices satisfying conditions (5.1) and (5.2) were used:

$$E^1 = \begin{bmatrix} -0.1 & 0 \\ 0 & 0 \\ 0 & -0.1 \\ 0 & 0 \end{bmatrix}, \qquad F^1 = \begin{bmatrix} 0 & 0 \\ -0.1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -0.1 \\ 0 & 0 \end{bmatrix},$$

$$E^2 = \begin{bmatrix} -0.1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix}, \qquad F^2 = \begin{bmatrix} 0 & 0 \\ -0.1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix},$$

$$E^3 = \begin{bmatrix} 0 & 0 \\ 0.1 & 0 \\ 0 & -0.1 \\ 0 & 0 \end{bmatrix}, \qquad F^3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \\ 0 & 0 \\ 0 & -0.1 \\ 0 & 0 \end{bmatrix},$$

$$E^4 = \begin{bmatrix} 0 & 0 \\ 0.1 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix}, \qquad F^4 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

In order to allow the use of Theorem 5.2, Theorem 1 from (Mozelli, Palhares, and Avellar 2009), Theorem 1 from (Guedes et al. 2013), Theorem 4.3 from (Faria, Silva, and Oliveira 2012), Theorem 2 from (Lee, Park, and Joo 2012), Theorem 1 from (Ko, Lee, and Park 2012), and Theorem 1 from (Tognetti, Oliveira, and Peres 2011), the following bound on the time derivative of the membership functions was used:

$$\phi_i = 0.85, \ \forall i.$$

Note that, most likely, this is not a global bound for the membership functions time derivative in this example (and given large values of a and b, most likely, is valid only for a very small region around the origin). Therefore, as stated in Section 5.3, conditions based on this bound will only assert the system's local stability. Nevertheless, this is a very common value for this bound found on the literature (Faria, Silva, and Oliveira 2012; Guedes et al. 2013; Ko and Park 2012; Ko, Lee, and Park 2012; Lee, Park, and Joo 2012; Mozelli, Palhares, and Avellar 2009; Rhee and Won 2006; Tognetti, Oliveira, and Peres 2011; Zhang and Xie 2011) and will be used for comparison purposes.

In order to compare the relaxation brought about by the transformation matrices, both the method proposed in (Bernal, Guerra, and Kruszewski 2009) as well as the method discussed in Section 5.2 were used.

Using the constructing method detailed in (Bernal, Guerra, and Kruszewski 2009), the transformation matrices are

$$G^1 = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix},$$

$$G^3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

$$G^4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1 \end{bmatrix}.$$

Using the method presented in Section 5.2, the transformation matrices are:

$$G^1 = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.25 \\ 0 & 0.5 & 0 & 0.25 \\ 0 & 0 & 0.5 & 0.25 \\ 0 & 0 & 0 & 0.25 \end{bmatrix},$$

$$G^2 = \begin{bmatrix} 0.5 & 0 & 0.25 & 0 \\ 0.5 & 1 & 0.25 & 0.5 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0.25 & 0.5 \end{bmatrix},$$

$$G^3 = \begin{bmatrix} 0.5 & 0.25 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0.5 & 0.25 & 1 & 0.5 \\ 0 & 0.25 & 0 & 0.5 \end{bmatrix},$$

$$G^4 = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0.25 & 0.5 & 0 & 0 \\ 0.25 & 0 & 0.5 & 0 \\ 0.25 & 0.5 & 0.5 & 1 \end{bmatrix}.$$

The results are presented in Tables 5.1 and 5.2, where a line search method was used to find the largest value of b, given a fixed a, in which the conditions were still feasible. Table 5.1 shows the results obtained using different derivative independent conditions (Theorem 5.1, Theorem 4 from Mozelli, Palhares, and Avellar 2009, and Theorem 2 from (Bernal, Guerra, and Kruszewski 2009)), whereas Table 5.2 shows the results obtained using different derivative dependent conditions (Theorem 5.2, Theorem 1 from Mozelli, Palhares, and Avellar 2009, Theorem 1 from (Guedes et al. 2013), Theorem 4.3 from Faria, Silva, and Oliveira 2012, Theorem 2 from Lee, Park, and Joo 2012, Theorem 1 from Ko, Lee, and Park 2012, and Theorem 1 from Tognetti, Oliveira, and Peres 2011).

From these tables, we can see that Theorem 5.1 has a better performance than the other derivative independent conditions, as well as better performance than many derivative dependent conditions for $a = -2$ (with the exception of Theorem 5.2). In addition, note that, when Theorem 5.1 is used in conjunction with the CNO transformation it behaves better than Theorem 1 from (Mozelli, Palhares, and Avellar 2009) (which is a derivative dependent condition). From Table 5.2, we also see that Theorem 5.2 had a better perfomance than the other derivative dependent conditions.

Note that, in general, the derivative dependent conditions presented a superior performance than the independent conditions for this example. However, in this case, the larger the values of a and b, the smaller the corresponding region on the state space for which these conditions assure the stability. Whereas, in this example, the derivative independent conditions always assure the global stability of the system when they are feasible (and for that reason they are valid for a smaller set of a and b parameters than the derivative dependent conditions).

Another interesting observation that can be made regarding the tables is that, in this example, the use of the CNO transformation against the use of the transformation presented in (Bernal, Guerra, and Kruszewski 2009) yielded better results.

Figures 5.1 to 5.5 illustrate the conditions and transformations proposed for $a \in [-18, 0]$ and $b \in [0, 1000]$.

Figure 5.1 compares Theorem 5.1, using the transformation matrices from (Bernal, Guerra, and Kruszewski 2009), with other conditions from the literature.

Table 5.1: Maximum feasible values of b, in the example, for the derivative independent conditions. Compares Theorem 5.1 used in conjunction with the transformation proposed in (Bernal, Guerra, and Kruszewski 2009) and the CNO transformation, (Bernal, Guerra, and Kruszewski 2009, Theorem 2) used in conjunction with the transformation proposed in (Bernal, Guerra, and Kruszewski 2009) and the CNO transformation, and (Mozelli, Palhares, and Avellar 2009, Theorem 4).

Method	a				
	-16	-12	-8	-4	-2
Mozelli, Palhares, and Avellar 2009, Th.4	441	339	233	120	65
Bernal, Guerra, and Kruszewski 2009, Th.2 + Bernal, Guerra, and Kruszewski 2009	317	222	139	74	48
Bernal, Guerra, and Kruszewski 2009, Th.2 + **CNO**	421	292	186	103	70
Th.5.1 + Bernal, Guerra, and Kruszewski 2009	780	638	390	288	201
Th.5.1 + **CNO**	1012	782	558	352	253

Table 5.2: Maximum feasible values of b, in the example, for the derivative dependent conditions. Compares Theorem 5.2 used in conjunction with the transformation proposed in (Bernal, Guerra, and Kruszewski 2009) and the CNO transformation, (Mozelli, Palhares, and Avellar 2009, Theorem 1), (Guedes et al. 2013, Theorem 1), (Faria, Silva, and Oliveira 2012, Theorem 4.3), (Lee, Park, and Joo 2012, Theorem 2), (Ko, Lee, and Park 2012, Theorem 1), and (Tognetti, Oliveira, and Peres 2011, Theorem 1) with $g = 3$ and $d = 3$.

Method	a				
	-16	-12	-8	-4	-2
Mozelli, Palhares, and Avellar 2009, Th.1	964	603	323	123	51
Guedes et al. 2013, Th.1	1284	805	431	161	66
Faria, Silva, and Oliveira 2012, Th.4.3	1360	850	452	167	66
Lee, Park, and Joo 2012, Th.2	1914	1072	547	202	88
Ko, Lee, and Park 2012, Th.1	2836	1704	853	286	104
Tognetti, Oliveira, and Peres 2011, Th.$1_{(3,3)}$	4727	2789	1383	469	188
Th.5.2 + Bernal, Guerra, and Kruszewski 2009	37244	39911	4979	463	219
Th.5.2 + **CNO**	983122	992028	726248	4529	846

Figure 5.2 compares Theorem 5.2, using the transformation matrices from (Bernal, Guerra, and Kruszewski 2009), with other conditions from the literature.

Figures 5.3, 5.4 and 5.5 compare the results obtained with the transformation discussed in Section 5.2 with those obtained with the transformation proposed in (Bernal, Guerra, and Kruszewski 2009). They deal, respectively, with Theorem 2 from (Bernal, Guerra, and Kruszewski 2009), Theorem 5.1, and Theorem 5.2. There are no guarantees that the transformation matrices discussed in Section 5.2 will have a better behaviour than the transformation matrices from (Bernal, Guerra, and Kruszewski 2009). However, as can be seen from the figures presented, that was the case in this example.

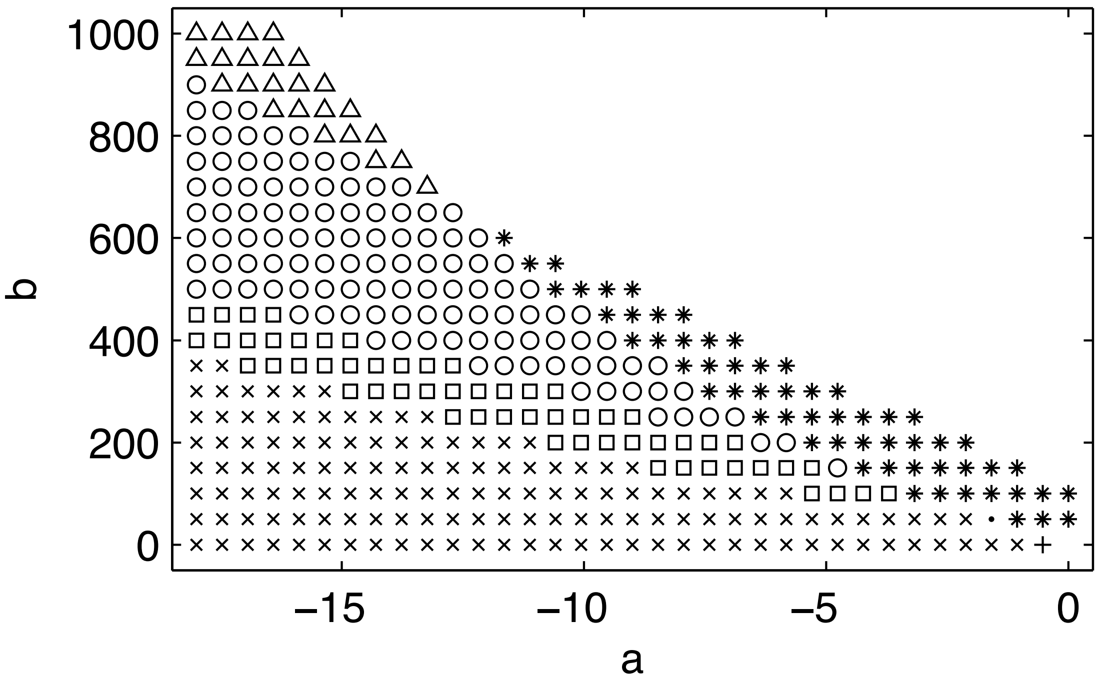

Figure 5.1: Feasibility space for different conditions using the local transformation matrices in (Bernal, Guerra, and Kruszewski 2009). Theorem 2 from (Bernal, Guerra, and Kruszewski 2009) is feasible in the $\times$ and $+$ region. Theorem 4 from (Mozelli, Palhares, and Avellar 2009) is feasible in the $\times$, $\square$ and $\cdot$ region. Theorem 1 from (Mozelli, Palhares, and Avellar 2009) is feasible in the $\times$, $\square$, $\bigcirc$ and $\triangle$ region. Theorem 5.1 is feasible in the $\times$, $\square$, $\bigcirc$, $*$, $+$ and $\cdot$ region.

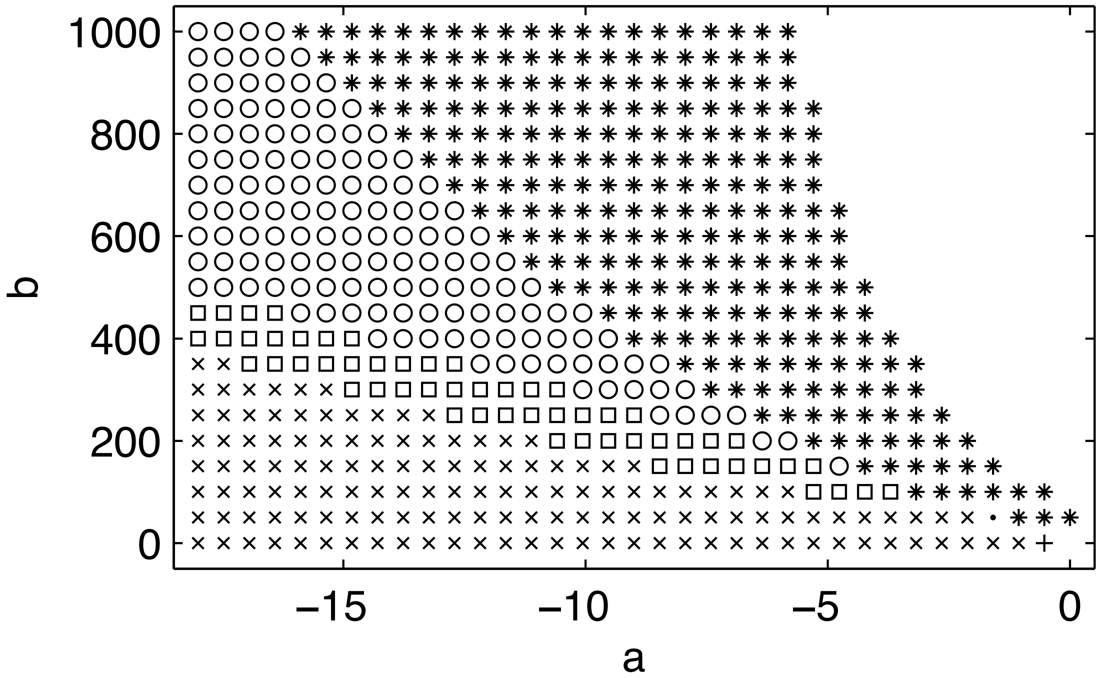

Figure 5.2: Feasibility space for different conditions using the local transformation matrices in (Bernal, Guerra, and Kruszewski 2009). Theorem 2 from (Bernal, Guerra, and Kruszewski 2009) is feasible in the × and + region. Theorem 4 from (Mozelli, Palhares, and Avellar 2009) is feasible in the ×, □ and ·. Theorem 1 from (Mozelli, Palhares, and Avellar 2009) is feasible in the ×, □ and ◯ region. Theorem 5.2 is feasible for all the points shown in the figure.

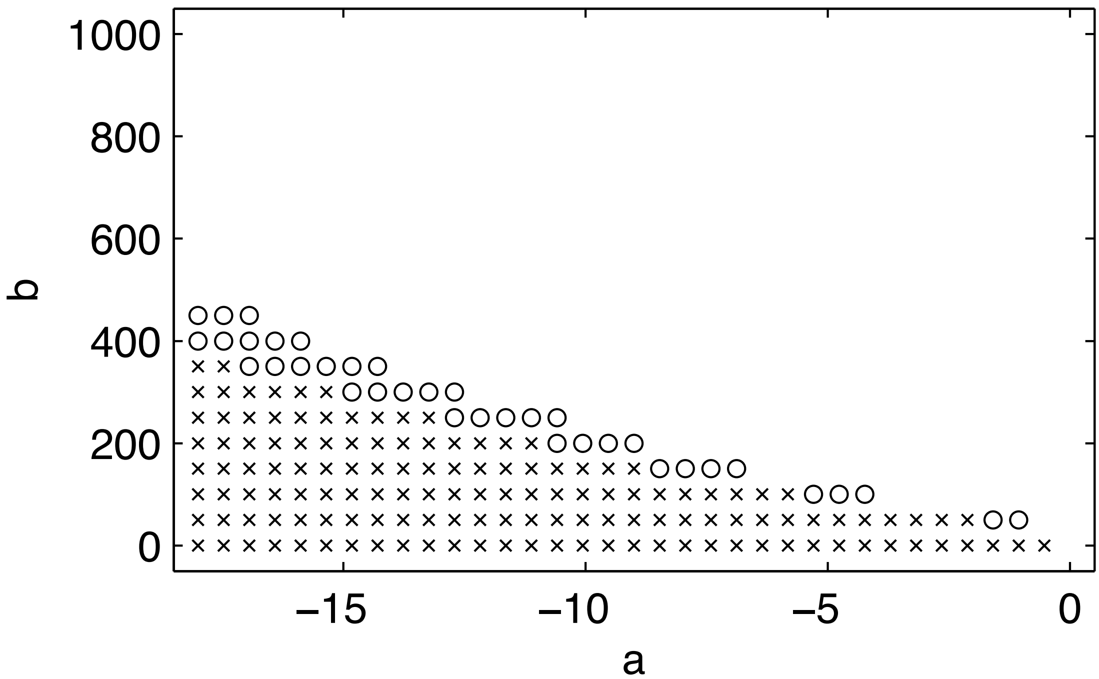

Figure 5.3: Feasibility space for different transformation matrices using Theorem 2 from (Bernal, Guerra, and Kruszewski 2009). With the transformation proposed in (Bernal, Guerra, and Kruszewski 2009), conditions are feasible in the × region. With the transformation discussed in Section 5.2, conditions are valid for all the points shown in the figure.

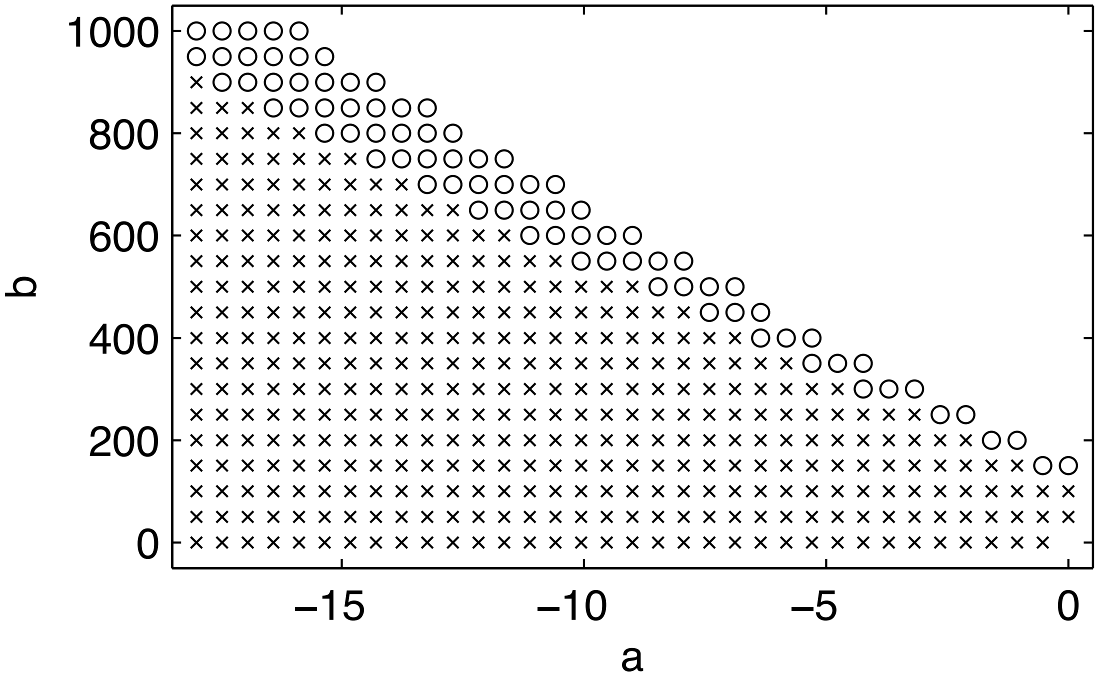

Figure 5.4: Feasibility space for different transformation matrices using Theorem 5.1. With the transformation proposed in (Bernal, Guerra, and Kruszewski 2009), conditions are feasible in the × region. With the transformation discussed in Section 5.2, conditions are valid for all the points shown in the figure.

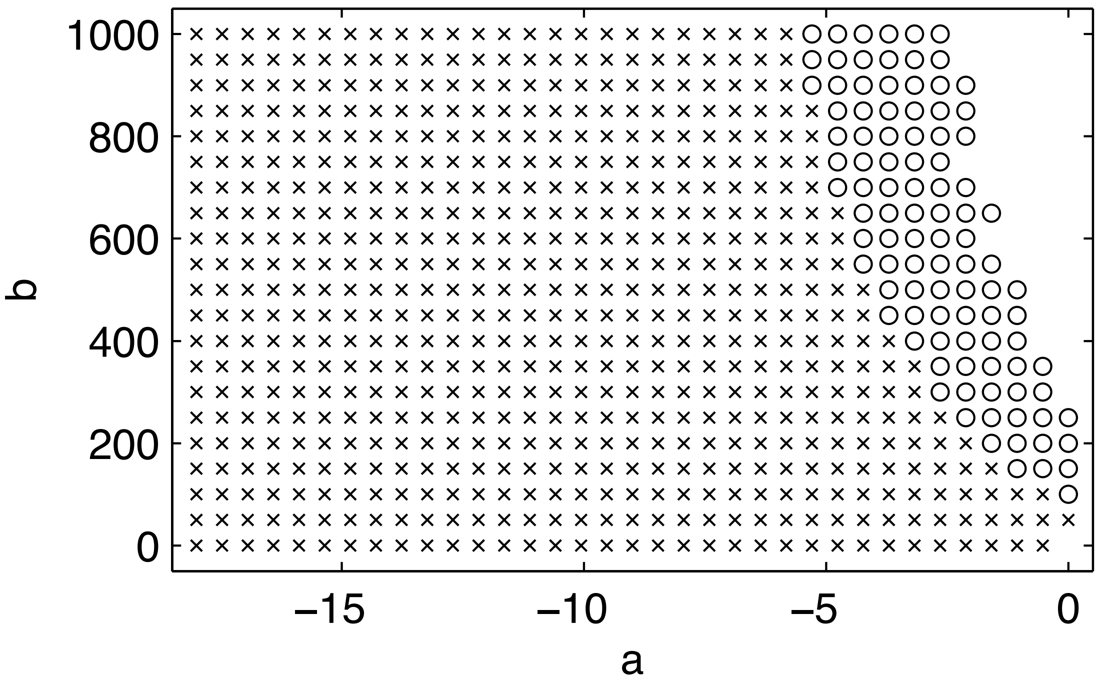

Figure 5.5: Feasibility space for different transformation matrices using Theorem 5.2. With the transformation proposed in (Bernal, Guerra, and Kruszewski 2009), conditions are feasible in the × region. With the transformation discussed in Section 5.2, conditions are valid for all the points shown in the figure.

6 A Novel Way to Deal with Nonquadratic Lyapunov Functions in Continuous Time TS Systems: Stability and Stabilization Conditions

A novel way to deal with the time-derivative of nonquadratic Lyapunov functions in local conditions is proposed. This new proposition allow us to avoid the use of direct bounds over the time-derivative of the membership functions, and therefore avoid the need of a separate set of conditions to guarantee those bounds. New stability and stabilization conditions are presented as well as examples to illustrate their uses.

The work presented in this chapter was done in collaboration with Raymundo Márquez under the supervision of Prof. Thierry Marie Guerra.

6.1 Introduction

Since TS systems are nonlinear systems, an important source of conservatism in the LMIs comes from the family of candidate Lyapunov functions used in the conditions (Sala 2009). Among these families of functions, quadratic Lyapunov functions yield the simplest LMI conditions and can be readily applied in most of the cases due to their simplicity. However, they usually yield the most conservative results (since other families of functions are usually constructed in a way that they cover quadratic functions).

Nonquadratic, or fuzzy Lyapunov functions (Tanaka, Hori, and Wang 2003) use a quadratic function for each of the system's rules with the same membership functions, yielding a convex sum of quadratic functions. When deriving stability conditions for continuous-time systems using these functions, the time derivative of the membership functions usually appears.

Many conditions in the literature (Campos et al. 2013; Faria, Silva, and Oliveira 2012; Guedes et al. 2013; Ko and Park 2012; Ko, Lee, and Park 2012; Mozelli, Palhares, and Avellar 2009; Mozelli et al. 2009; Tognetti, Oliveira, and Peres 2011; Zhang and Xie 2011) make use of previously known upper bounds on these time derivatives in order to present LMI analysis and synthesis conditions. In the analysis case, these upper bounds can usually be found beforehand and this does not usually represent a restriction. In the synthesis case, however, if any of the premise variables are state variables, this assumption is violated and stability is only assured for the largest sublevel-set of the Lyapunov function inside the region given by the upper bound used.

Trying to get rid of a priori bounds on the time-derivatives, recent works in the literature (Bernal and Guerra 2010; Guerra et al. 2012; Lee, Joo, and Tak 2014; Lee and Kim 2014; Lee, Park, and Joo 2012; Pan et al. 2012) give up on asserting global stability and instead pursue a local certification. Some (Bernal and Guerra 2010; Guerra et al. 2012) use the state space region the model is confined in, as well as bounds on the membership functions partial derivatives regarding the states and bounds on the control effort to enforce previously specified bounds on the time derivatives. Others (Lee and Kim 2014; Lee, Park, and Joo 2012; Pan et al. 2012), present conditions that need not directly bound the control efforts, while some others (Lee, Joo, and Tak 2014) also remove the need to specify the time derivative bounds (in the analysis case).

By making use of a polytopic representation of a local region in state space and some novel manipulations, this chapter avoids the use of bounds on the membership functions time derivative by explicitly including the state vector into the stability conditions. In the stabilization case, we make use of a non-PDC control law and some sufficient conditions to adapt the results from the stability case. The stabilization conditions are, in a way, conservative but are still guaranteed to give better results than those with a quadratic Lyapunov function.

6.2 Considerations and special notation

Throughout this chapter, we consider continuous-time TS systems described by

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{x}) A_i \mathbf{x}, \tag{6.1}$$

for the stability conditions, or

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{x}) \left(A_i \mathbf{x} + B_i \mathbf{u} \right), \tag{6.2}$$

for the stabilization conditions. In these equations, $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the input vector, $A_i \in \mathbb{R}^{n \times n}$ and $B_i \in \mathbb{R}^{n \times m}$ are the system matrices, $h_i(\mathbf{x})$ are the membership functions (that we assume may only depend on the system's states), and r is the number of rules.

Since we are dealing with TS models, the membership functions respect condition (2.3).

Throughout this chapter, we also consider a TS model for the Jacobian of the membership functions' vector $\mathbf{h} = \begin{bmatrix} h_1 & h_2 & \dots & h_r \end{bmatrix}^T \in \mathbb{R}^r$ (which can be found either by sector nonlinearity (Tanaka and Wang 2001), or by the Tensor Product Model Transformation (Baranyi 2004)). We consider this model to be described by

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \sum_{j=1}^{n_g} g_j(\mathbf{x}) J_j, \tag{6.3}$$

with $J_j \in \mathbb{R}^{r \times n}$ the matrices that represent the Jacobian, n_g the number of rules that describe its TS model, and $g_j(\mathbf{x})$ its membership functions with properties

$$g_j \geq 0, \qquad \sum_{j=1}^{n_g} g_j = 1.$$

For the stability case, we restrict our analysis to a polytopic region around the origin given by

$$\mathbf{x} = \sum_{k=1}^{n_\alpha} \alpha_k \mathbf{x}_k, \qquad \sum_{k=1}^{n_\alpha} \alpha_k = 1, \qquad \alpha_k \geq 0, \tag{6.4}$$

with $\mathbf{x}_k$ the vertices of the polytopic region and n_α the number of vertices describing the region.

In the stabilization case, we work instead with a transformed state space (represented in this chapter by the variable $\mathbf{s}$). In the same manner as in the stability case, we restrict our conditions to a polytopic region around the origin given by

$$\mathbf{s} = \sum_{k=1}^{n_\alpha} \alpha_k \mathbf{s}_k, \qquad \sum_{k=1}^{n_\alpha} \alpha_k = 1, \qquad \alpha_k \geq 0. \tag{6.5}$$

with $\mathbf{s}_k$ the vertices of the polytopic region and n_α the number of vertices describing the region.

For the remainder of the chapter, the following short representations will be used for a single fuzzy sum and its inverse

$$P_h = \sum_{i=1}^{r} h_i(\mathbf{x}) P_i,$$
$$P_h^{-1} = \left(\sum_{i=1}^{r} h_i(\mathbf{x}) P_i \right)^{-1}.$$

Similarly, the following short representations will be used for a multiple fuzzy sum and its inverse

$$P_{h^q} = \sum_{i_1=1}^{r} \cdots \sum_{i_q=1}^{r} \left(\prod_{\ell=1}^{q} h_{i_\ell}(\mathbf{x}) \right) P_{i_1 \ldots i_q},$$
$$P_{h^q}^{-1} = \left(\sum_{i_1=1}^{r} \cdots \sum_{i_q=1}^{r} \left(\prod_{\ell=1}^{q} h_{i_\ell}(\mathbf{x}) \right) P_{i_1 \ldots i_q} \right)^{-1}.$$

In addition to h, we also consider that g and α can also be used as subscripts in this same manner to denote fuzzy summations. Therefore, if at some point we have a matrix $L_{h^q g\alpha}$, it will represent

$$L_{h^q g\alpha} = \sum_{i_1=1}^{r} \cdots \sum_{i_q=1}^{r} \sum_{j=1}^{n_g} \sum_{k=1}^{n_\alpha} \left(\prod_{\ell=1}^{q} h_{i_\ell} \right) g_j \alpha_k L_{i_1 \ldots i_q jk}.$$

The following Lemmas will prove useful in later developments

Lemma 6.1: Finsler's Lemma (Mozelli, Palhares, and Mendes 2010)

The following statements are equivalent:

1. $\mathbf{x}^T Q \mathbf{x} < 0, \quad \forall \mathbf{x} \mid B\mathbf{x} = 0,$
2. $B^{\perp T} Q B^{\perp} < 0, \quad BB^{\perp} = 0$ (i.e. $B^{\perp}$ *is a base for the null space of* B*),*
3. $\exists N \mid \mathbf{x}^T \left(Q + N^T B + B^T N \right) \mathbf{x} < 0,$
4. $\exists \mu > 0 \mid \mathbf{x}^T \left(Q - \mu B^T B \right) \mathbf{x} < 0.$

Lemma 6.2: Derivative of the matrix inverse

The derivate of the inverse of a matrix function is related to the derivative of the matrix function by means of the following equality

$$(\dot{Q^{-1}}(t)) = -Q^{-1}(t)\dot{Q}(t)Q^{-1}(t).$$

Proof. By taking the derivative of both sides of the equality

$$Q^{-1}(t)Q(t) = I,$$

we get that

$$\begin{aligned}(\dot{Q^{-1}}(t))Q(t) + Q^{-1}(t)\dot{Q}(t) &= 0,\\ (\dot{Q^{-1}}(t)) &= -Q^{-1}(t)\dot{Q}(t)Q^{-1}(t).\end{aligned}$$

□

Since in this chapter we work with multiple fuzzy summations, we make use of Lemma 2.4 to computationally implement them.

In order to illustrate the assumptions and notations presented in this section, we present an example system.

Example 6.1. Consider a system with four rules, with

$$A_1 = \begin{bmatrix} -3 & 2 \\ 0 & -0.9 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} -0.8 & 3 \\ 0 & -0.9 \end{bmatrix}$$
$$A_3 = \begin{bmatrix} -1.9 & 2 \\ -0.5 & 0.1 \end{bmatrix}, \qquad A_4 = \begin{bmatrix} 0.1 & 3 \\ -0.5 & -2 \end{bmatrix}$$

in a polytopic region given by vertices

$$\mathbf{x}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \qquad \mathbf{x}_2 = \begin{bmatrix} -2 \\ 2 \end{bmatrix}, \qquad \mathbf{x}_3 = \begin{bmatrix} 2 \\ -2 \end{bmatrix}, \qquad \mathbf{x}_4 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

and membership functions

$$h_1(\mathbf{x}) = \frac{\left(4 - x_1^2\right)\left(4 - x_2^2\right)}{16}, \qquad h_2(\mathbf{x}) = \frac{\left(4 - x_1^2\right)x_2^2}{16},$$
$$h_3(\mathbf{x}) = \frac{x_1^2\left(4 - x_2^2\right)}{16}, \qquad h_4(\mathbf{x}) = \frac{x_1^2 x_2^2}{16},$$

The Jacobian of the membership functions' is given by

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \frac{1}{8}\begin{bmatrix} -x_1(4 - x_2^2) & -x_2(4 - x_1^2) \\ -x_1 x_2^2 & x_2(4 - x_1^2) \\ x_1(4 - x_2^2) & -x_2 x_1^2 \\ x_1 x_2^2 & x_2 x_1^2 \end{bmatrix},$$

and we can write a 16 rule TS model for it with

$$J_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}, \quad J_2 = \begin{bmatrix} 0 & 2 \\ 1 & -2 \\ 0 & -1 \\ -1 & 1 \end{bmatrix}, \quad J_3 = \begin{bmatrix} 2 & 0 \\ -1 & 0 \\ -2 & 1 \\ 1 & -1 \end{bmatrix}, \quad J_4 = \begin{bmatrix} 2 & 2 \\ -1 & -2 \\ -2 & -1 \\ 1 & 1 \end{bmatrix},$$

$$J_5 = \begin{bmatrix} 0 & -2 \\ 1 & 2 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}, \quad J_6 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & -1 \\ -1 & 1 \end{bmatrix}, \quad J_7 = \begin{bmatrix} 2 & -2 \\ -1 & 2 \\ -2 & 1 \\ 1 & -1 \end{bmatrix}, \quad J_8 = \begin{bmatrix} 2 & 0 \\ -1 & 0 \\ -2 & -1 \\ 1 & 1 \end{bmatrix},$$

$$J_9 = \begin{bmatrix} -2 & 0 \\ 1 & 0 \\ 2 & 1 \\ -1 & -1 \end{bmatrix}, \quad J_{10} = \begin{bmatrix} -2 & 2 \\ 1 & -2 \\ 2 & -1 \\ -1 & 1 \end{bmatrix}, \quad J_{11} = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix}, \quad J_{12} = \begin{bmatrix} 0 & 2 \\ -1 & -2 \\ 0 & -1 \\ 1 & 1 \end{bmatrix},$$

$$J_{13} = \begin{bmatrix} -2 & -2 \\ 1 & 2 \\ 2 & 1 \\ -1 & -1 \end{bmatrix}, \quad J_{14} = \begin{bmatrix} -2 & 0 \\ 1 & 0 \\ 2 & -1 \\ -1 & 1 \end{bmatrix}, \quad J_{15} = \begin{bmatrix} 0 & -2 \\ -1 & 2 \\ 0 & 1 \\ 1 & -1 \end{bmatrix}, \quad J_{16} = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix}.$$

and

$$\begin{aligned}
g_1(\mathbf{x}) &= \omega_{11}\omega_{21}\omega_{31}\omega_{41}, & g_2(\mathbf{x}) &= \omega_{11}\omega_{21}\omega_{31}\omega_{42},\\
g_3(\mathbf{x}) &= \omega_{11}\omega_{21}\omega_{32}\omega_{41}, & g_4(\mathbf{x}) &= \omega_{11}\omega_{21}\omega_{32}\omega_{42},\\
g_5(\mathbf{x}) &= \omega_{11}\omega_{22}\omega_{31}\omega_{41}, & g_6(\mathbf{x}) &= \omega_{11}\omega_{22}\omega_{31}\omega_{42},\\
g_7(\mathbf{x}) &= \omega_{11}\omega_{22}\omega_{32}\omega_{41}, & g_8(\mathbf{x}) &= \omega_{11}\omega_{22}\omega_{32}\omega_{42},\\
g_9(\mathbf{x}) &= \omega_{12}\omega_{21}\omega_{31}\omega_{41}, & g_{10}(\mathbf{x}) &= \omega_{12}\omega_{21}\omega_{31}\omega_{42},\\
g_{11}(\mathbf{x}) &= \omega_{12}\omega_{21}\omega_{32}\omega_{41}, & g_{12}(\mathbf{x}) &= \omega_{12}\omega_{21}\omega_{32}\omega_{42},\\
g_{13}(\mathbf{x}) &= \omega_{12}\omega_{22}\omega_{31}\omega_{41}, & g_{14}(\mathbf{x}) &= \omega_{12}\omega_{22}\omega_{31}\omega_{42},\\
g_{15}(\mathbf{x}) &= \omega_{12}\omega_{22}\omega_{32}\omega_{41}, & g_{16}(\mathbf{x}) &= \omega_{12}\omega_{22}\omega_{32}\omega_{42},
\end{aligned}$$

with

$$\begin{aligned}
\omega_{11}(\mathbf{x}) &= \frac{2 - x_1}{4}, & \omega_{12}(\mathbf{x}) &= \frac{2 + x_1}{4},\\
\omega_{21}(\mathbf{x}) &= \frac{2 - x_2}{4}, & \omega_{22}(\mathbf{x}) &= \frac{2 + x_2}{4},\\
\omega_{31}(\mathbf{x}) &= \frac{8 - x_1 x_2^2}{16}, & \omega_{32}(\mathbf{x}) &= \frac{8 + x_1 x_2^2}{16},\\
\omega_{41}(\mathbf{x}) &= \frac{8 - x_2 x_1^2}{16}, & \omega_{42}(\mathbf{x}) &= \frac{8 + x_2 x_1^2}{16}.
\end{aligned}$$

6.3 Stability Conditions

The main idea of the stability conditions proposed in this section is to avoid using bounds over the membership functions time derivative by explicitly including the state vector into the stability conditions. By using a polytopic representation of a local region in state space, we are then able to get LMI conditions.

Consider a Lyapunov function of the form

$$V(\mathbf{x}) = \mathbf{x}^T P_h \mathbf{x},$$

its time derivative is given by

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T P_h \mathbf{x} + \mathbf{x}^T P_h \dot{\mathbf{x}} + \mathbf{x}^T \dot{P}_h \mathbf{x}.$$

However, note that the last term can be written as

$$\begin{aligned}
\mathbf{x}^T \dot{P}_h \mathbf{x} =& \sum_k^r \dot{h}_k \mathbf{x}^T P_k \mathbf{x}, \\
=& \mathbf{x}^T \begin{bmatrix} P_1 \mathbf{x} & \dots & P_r \mathbf{x} \end{bmatrix} \begin{bmatrix} \dot{h}_1 \\ \vdots \\ \dot{h}_r \end{bmatrix}, \\
=& \mathbf{x}^T \begin{bmatrix} P_1 \mathbf{x} & \dots & P_r \mathbf{x} \end{bmatrix} \dot{\mathbf{h}},
\end{aligned}$$

and, by considering a polytopic region in the state space given by (6.4), we get that

$$\begin{aligned}
\mathbf{x}^T \dot{P}_h \mathbf{x} &= \mathbf{x}^T \begin{bmatrix} P_1 \mathbf{x}_\alpha & \dots P_r \mathbf{x}_\alpha \end{bmatrix} \dot{\mathbf{h}}, \\
&= \mathbf{x}^T Q_\alpha \dot{\mathbf{h}}, \\
&= \frac{1}{2} \left(\mathbf{x}^T Q_\alpha \dot{\mathbf{h}} + \dot{\mathbf{h}}^T Q_\alpha^T \mathbf{x} \right),
\end{aligned}$$

in which we used $\mathbf{x}_\alpha$ to indicate that we replace the state vector by its polytopic representation.

From this point forward, we only need the relations

$$\begin{aligned}
\dot{\mathbf{x}} &= A_h \mathbf{x}, \\
\dot{\mathbf{h}} &= J_g \dot{\mathbf{x}}, \\
\sum_{i=1}^r \dot{h}_i &= 0,
\end{aligned}$$

in order to get suitable LMI conditions for the stability problem.

In the following, we present a generalization of this idea to the case in which the Lyapunov function may be described by multiple fuzzy summations. In addition, we apply the conditions to the system in Example 6.1 to illustrate its use.

Theorem 6.1: Nonquadratic stability conditions

Given a system described by (6.1) *in a polytopic region given by* (6.4) *whose membership functions' Jacobian matrix can be described by* (6.3)*, and a desired number q of fuzzy summations in the Lyapunov function*

$$V(\mathbf{x}) = \mathbf{x}^T P_{h^q} \mathbf{x}. \tag{6.6}$$

If there exists P_{h^q} *and* $N_{h^{q-1}\alpha}$ *such that*

$$\begin{aligned} &P_{h^q} > 0, \\ &\Lambda_{h^q\alpha} + N_{h^{q-1}\alpha}^T \tilde{A}_{hg} + \tilde{A}_{hg}^T N_{h^{q-1}\alpha} < 0, \end{aligned}$$

with

$$\Lambda_{h^q\alpha} = \begin{bmatrix} 0 & P_{h^q} & \frac{1}{2} Q_{h^{q-1}\alpha} \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix},$$

$$\tilde{A}_{hg} = \begin{bmatrix} A_h & -I & 0 \\ 0 & J_g & -I \\ 0 & 0 & \mathbf{1}^T \end{bmatrix},$$

and

$$Q_{h^{q-1}\alpha} = \begin{bmatrix} P_{h^{q-1}}^{(1)} \mathbf{x}_\alpha & \dots & P_{h^{q-1}}^{(r)} \mathbf{x}_\alpha \end{bmatrix}, \tag{6.7}$$

$$P_{h^{q-1}}^{(k)} = \sum_{i_1=1}^{r} \cdots \sum_{i_{q-1}=1}^{r} \left(\prod_{\ell=1}^{q-1} h_{i_\ell}(\mathbf{x}) \right) \left(P_{k i_1 \dots i_{q-1}} + P_{i_1 k \dots i_{q-1}} + \cdots + P_{i_1 \dots k i_{q-1}} + P_{i_1 \dots i_{q-1} k} \right), \tag{6.8}$$

then the system is locally asymptotically stable with the Lyapunov function (6.6) *and an estimate of its domain of attraction is given by the largest sublevel set of the Lyapunov function that is inside the polytopic region described by* (6.4).

Proof. Consider that we have a fuzzy Lyapunov function of the form (6.6).

In order to assure that it is positive definite, it suffices that

$$P_{h^q} > 0.$$

Its time derivative is given by

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T P_{h^q} \mathbf{x} + \mathbf{x}^T P_{h^q} \dot{\mathbf{x}} + \mathbf{x}^T \dot{P}_{h^q} \mathbf{x}.$$

However, note that we can write

$$\begin{aligned}
\dot{P}_{h^q} &= \sum_{i_1=1}^{r} \cdots \sum_{i_q=1}^{r} \left(\dot{h}_{i_1} \prod_{\substack{\ell=1 \\ \ell \neq 1}}^{q} h_{i_\ell}(\mathbf{x}) + \cdots + \dot{h}_{i_q} \prod_{\substack{\ell=1 \\ \ell \neq q}}^{q} h_{i_\ell}(\mathbf{x}) \right) P_{i_1 \ldots i_q}, \\
&= \sum_{i_1=1}^{r} \cdots \sum_{i_q=1}^{r} \left(\sum_{\kappa=1}^{q} \dot{h}_{i_\kappa} \prod_{\substack{\ell=1 \\ \ell \neq \kappa}}^{q} h_{i_\ell}(\mathbf{x}) \right) P_{i_1 \ldots i_q}, \\
&= \sum_{k}^{r} \dot{h}_k \sum_{i_1=1}^{r} \cdots \sum_{i_{q-1}=1}^{r} \left(\prod_{\ell=1}^{q-1} h_{i_\ell}(\mathbf{x}) \right) \\
&\quad \left(P_{k i_1 \ldots i_{q-1}} + P_{i_1 k \ldots i_{q-1}} + \cdots + P_{i_1 \ldots k i_{q-1}} + P_{i_1 \ldots i_{q-1} k} \right).
\end{aligned}$$

By using $P_{h^{q-1}}^{(k)}$ defined in (6.8) we get that

$$\begin{aligned}
\mathbf{x}^T \dot{P}_{h^q} \mathbf{x} &= \sum_{k}^{r} \dot{h}_k \mathbf{x}^T P_{h^{q-1}}^{(k)} \mathbf{x}, \\
&= \mathbf{x}^T \begin{bmatrix} P_{h^{q-1}}^{(1)} \mathbf{x} & \ldots & P_{h^{q-1}}^{(r)} \mathbf{x} \end{bmatrix} \begin{bmatrix} \dot{h}_1 \\ \vdots \\ \dot{h}_r \end{bmatrix}, \\
&= \mathbf{x}^T \begin{bmatrix} P_{h^{q-1}}^{(1)} \mathbf{x} & \ldots & P_{h^{q-1}}^{(r)} \mathbf{x} \end{bmatrix} \dot{\mathbf{h}}.
\end{aligned}$$

Inside the region described by (6.4), we can write

$$\begin{aligned}
\mathbf{x}^T \dot{P}_{h^q} \mathbf{x} &= \mathbf{x}^T \begin{bmatrix} P_{h^{q-1}}^{(1)} \mathbf{x}_\alpha & \ldots & P_{h^{q-1}}^{(r)} \mathbf{x}_\alpha \end{bmatrix} \dot{\mathbf{h}}, \\
&= \mathbf{x}^T Q_{h^{q-1}\alpha} \dot{\mathbf{h}}, \\
&= \frac{1}{2} \mathbf{x}^T Q_{h^{q-1}\alpha} \dot{\mathbf{h}} + \frac{1}{2} \dot{\mathbf{h}}^T Q_{h^{q-1}\alpha}^T \mathbf{x},
\end{aligned}$$

with $Q_{h^{q-1}\alpha}$ given by (6.7).

Therefore, the time derivative of the Lyapunov function can be written as

$$\begin{aligned}
\dot{V}(\mathbf{x}) &= \dot{\mathbf{x}}^T P_{h^q} \mathbf{x} + \mathbf{x}^T P_{h^q} \dot{\mathbf{x}} + \frac{1}{2} \mathbf{x}^T Q_{h^{q-1}\alpha} \dot{\mathbf{h}} + \frac{1}{2} \dot{\mathbf{h}}^T Q_{h^{q-1}\alpha}^T \mathbf{x}, \\
&= \begin{bmatrix} \mathbf{x}^T \dot{\mathbf{x}}^T \dot{\mathbf{h}}^T \end{bmatrix} \begin{bmatrix} 0 & P_{h^q} & \frac{1}{2} Q_{h^{q-1}\alpha} \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \\ \dot{\mathbf{h}} \end{bmatrix}, \\
&= \boldsymbol{\zeta}^T \Lambda_{h^q \alpha} \boldsymbol{\zeta}.
\end{aligned}$$

So to check if $\dot{V}(\mathbf{x}) < 0$, we need only to check that $\boldsymbol{\zeta}^T \Lambda_{h^q\alpha} \boldsymbol{\zeta} < 0$ under the equality constraints

$$\begin{bmatrix} A_h & -I & 0 \\ 0 & J_g & -I \\ 0 & 0 & \mathbf{1}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \\ \dot{\mathbf{h}} \end{bmatrix} = 0,$$

$$\tilde{A}_{hg} \boldsymbol{\zeta} = 0,$$

in which $\mathbf{1}^T$ represents a row vector of ones.

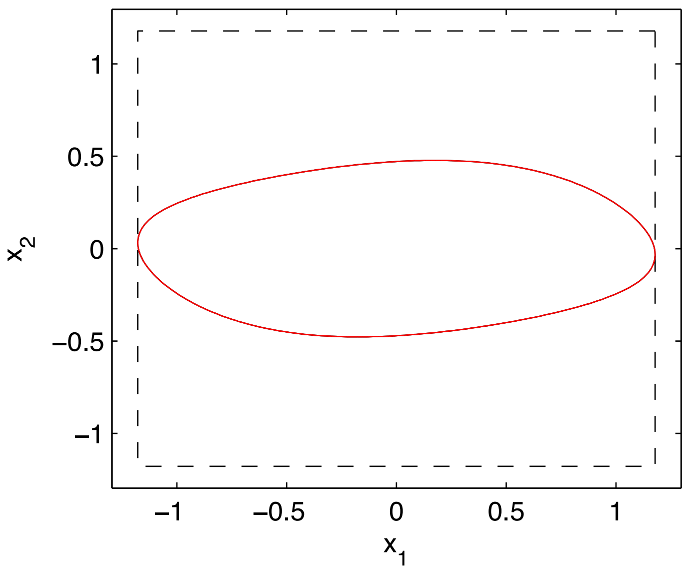

Figure 6.1: Largest domain of attraction found in Example 6.2 using Theorem 6.1 with $q = 1$.

So applying Finsler's Lemma (3 implies 1), and choosing the slack matrices in a way that we do not increase the number of fuzzy summations, we get that sufficient conditions for $\dot{V}(\mathbf{x}) < 0$ are given by

$$\Lambda_{h^q\alpha} + N^T_{h^{q-1}\alpha}\tilde{A}_{hg} + \tilde{A}^T_{hg}N_{h^{q-1}\alpha} < 0.$$

□

Remark 6.1

By making use of Lemma 2.4 to implement Theorem 6.1, we have $\frac{(r+q-1)!}{r!(q-1)!}(n+(2n+r)n_g n_\alpha)$ rows of LMIs and $\frac{r^q(n^2+n)}{2} + n_\alpha r^{q-1}(n+r+1)(2n+r)$ scalar decision variables, with n the number of states, r the number of rules of the TS model of the system, n_α the number of vertices used to described the local region, q the number of fuzzy summations in the Lyapunov function, and n_g the number of vertices used to describe the Jacobian of the membership functions vector.

Example 6.2. Consider the system in Example 6.1. If we apply the conditions from Theorem 6.1 with $q = 1$, they are not feasible for the region presented in Example 6.1. However, by doing a bisection search, we find that the conditions are feasible in the region $\|\mathbf{x}\|_\infty \leq 1.178$. Figure 6.1 presents the largest Lyapunov sublevel set that fits inside of the polytopic region which is the largest domain of attraction we are able to estimate using $q = 1$ in Theorem 6.1.

Using Theorem 6.1 with $q = 2$, the conditions work inside of the desired region, *i.e.* $\|\mathbf{x}\|_\infty \leq 2$. Figure 6.2 presents the largest Lyapunov sublevel set that fits inside of the polytopic region which is the largest domain of attraction we are able to estimate using $q = 2$ in Theorem 6.1. Figure 6.3 compares this domain of attraction with the ones found using (Lee, Joo, and Tak 2014, Theorem 1) and (Lee and Kim 2014, Theorem 1).

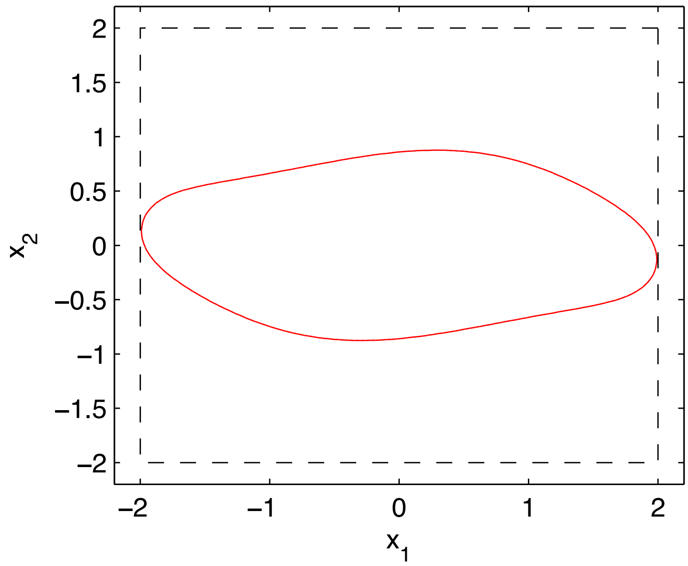

Figure 6.2: Largest domain of attraction found in Example 6.2 using Theorem 6.1 with $q = 2$.

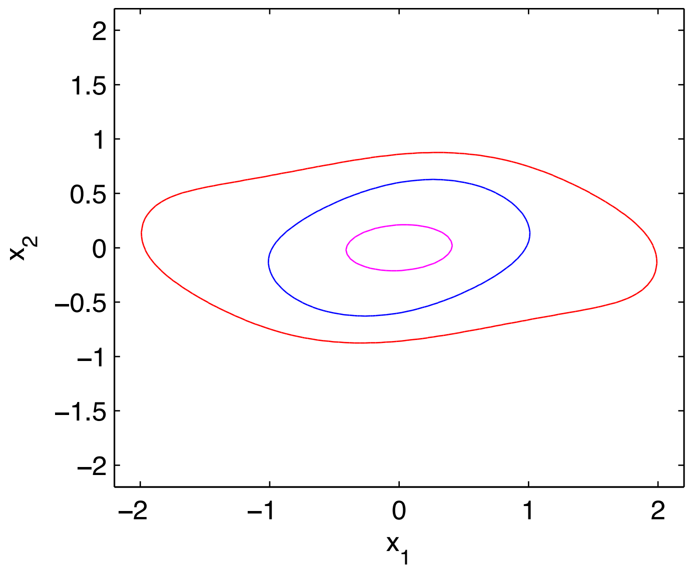

Figure 6.3: Different domains of attraction found in Example 6.2. The outermost region is the one found by Theorem 6.1 with $q = 2$ and $\|\mathbf{x}\|_\infty \leq 2$. The middle one is the one found by using the V-s iteration algorithm of (Lee, Joo, and Tak 2014) with $q = 2$ and 20 iterations. The innermost region is the one found by Theorem 1 of (Lee and Kim 2014) with $q = 2$ and $\phi_1 = \phi_2 = \phi_3 = \phi_4 = 2.59$.

6.4 Stabilization Conditions

The main idea of the stabilization conditions proposed in this section is to adapt the idea of the stability conditions to the closed loop scenario. If we were to directly substitute the closed loop system in the conditions from the previous section, it is not possible to get a set of LMI conditions. Therefore we present a set of conservative sufficient conditions to solve the problem, but that are guaranteed to give better results than a quadratic Lyapunov function coupled with a PDC control law would.

Theorem 6.2: Nonquadratic stabilization conditions

Given a system described by (6.2) *whose membership functions' Jacobian matrix can be described by* (6.3)*, a desired number q of fuzzy summations in the Lyapunov function and control gains, a scalar value μ, and a polytopic design region given by* (6.5)*, if there exists matrices P_{h^q}, K_{h^q}, F_{h^q} and $X_{h^{q-1}}$, and a scalar β solving the optimization problem*

$$\begin{aligned}
&\max \beta,\\
&\beta > 0,\\
&P_{h^q} > \beta I,\\
&\begin{bmatrix} A_h P_{h^q} + P_{h^q} A_h^T + B_h\left(K_{h^q} + F_{h^q}\right) + \left(K_{h^q}^T + F_{h^q}^T\right) B_h^T & * \\ J_g\left(P_{h^q} A_h^T + K_{h^q}^T B_h^T\right) + \mu \bar{Q}_{h^{q-1}\alpha} & -\mu\left(4I - \bar{Q}_{h^{q-1}\alpha}^T J_g^T - J_g \bar{Q}_{h^{q-1}\alpha}\right) \end{bmatrix} < 0,
\end{aligned}$$

with

$$\bar{Q}_{h^{q-1}\alpha} = \left[\left(P_{h^{q-1}}^{(1)} + X_{h^{q-1}}\right)\mathbf{s}_\alpha \quad \ldots \quad \left(P_{h^{q-1}}^{(r)} + X_{h^{q-1}}\right)\mathbf{s}_\alpha\right], \tag{6.9}$$

and $P_{h^{q-1}}^{(k)}$ in (6.8)*, then the system is locally asymptotically stable in closed loop with control law*

$$\mathbf{u} = \left(K_{h^q} + F_{h^q}\left(I - \frac{1}{2} J_g^T \bar{Q}_{h^{q-1}\alpha}^T\right)\right) P_{h^q}^{-1}\mathbf{x},$$

and Lyapunov function

$$V(\mathbf{x}) = \mathbf{x}^T P_{h^q}^{-1}\mathbf{x} \tag{6.10}$$

inside of the region $P_{h^q}\mathbf{s}$, where $\mathbf{s}$ is given by (6.5)*. In addition, an estimate of its domain of attraction is given by the largest sublevel set of the Lyapunov function that is inside this region.*

Proof. Consider we have a Lyapunov function as in (6.10), its time derivative would be given by

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T P_{h^q}^{-1}\mathbf{x} + \mathbf{x}^T P_{h^q}^{-1}\dot{\mathbf{x}} + \mathbf{x}^T \left(\dot{P_{h^q}^{-1}}\right)\mathbf{x}.$$

By remembering the fact the time derivative of the inverse of a matrix can be written as

$$\left(\dot{P_{h^q}^{-1}}\right) = -P_{h^q}^{-1}\dot{P}_{h^q}P_{h^q}^{-1},$$

we get that

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T P_{h^q}^{-1}\mathbf{x} + \mathbf{x}^T P_{h^q}^{-1}\dot{\mathbf{x}} - \mathbf{x}^T P_{h^q}^{-1}\dot{P}_{h^q}P_{h^q}^{-1}\mathbf{x}.$$

Similarly to the stability conditions, note that we can write

$$\begin{aligned}
\mathbf{x}^T P_{h^q}^{-1}\dot{P}_{h^q}P_{h^q}^{-1}\mathbf{x} =& \sum_k^r \dot{h}_k \mathbf{x}^T P_{h^q}^{-1} P_{h^{q-1}}^{(k)} P_{h^q}^{-1}\mathbf{x},\\
=& \mathbf{x}^T P_{h^q}^{-1}\left[P_{h^{q-1}}^{(1)} P_{h^q}^{-1}\mathbf{x} \quad \ldots \quad P_{h^{q-1}}^{(r)} P_{h^q}^{-1}\mathbf{x}\right]\dot{\mathbf{h}},
\end{aligned}$$

with $P^{(k)}_{h^{q-1}}$ given by (6.8).

Since

$$\sum_k^r \dot{h}_k \mathbf{x}^T P^{-1}_{h^q} X_{h^{q-1}} P^{-1}_{h^q} \mathbf{x} = 0,$$

we can add the extra matrix $X_{h^{q-1}}$ into this equality to get

$$\mathbf{x}^T P^{-1}_{h^q} \dot{P}_{h^q} P^{-1}_{h^q} \mathbf{x} = \mathbf{x}^T P^{-1}_{h^q} \left[\left(P^{(1)}_{h^{q-1}} + X_{h^{q-1}} \right) P^{-1}_{h^q} \mathbf{x} \quad \dots \quad \left(P^{(r)}_{h^{q-1}} + X_{h^{q-1}} \right) P^{-1}_{h^q} \mathbf{x} \right] \dot{\mathbf{h}}.$$

By defining $\mathbf{s} = P^{-1}_{h^q}\mathbf{x}$ and choosing a design region given by (6.5), we get that

$$\begin{aligned}
\mathbf{x}^T P^{-1}_{h^q} \dot{P}_h P^{-1}_{h^q} \mathbf{x} =& \mathbf{x}^T P^{-1}_{h^q} \left[\left(P^{(1)}_{h^{q-1}} + X_{h^{q-1}} \right) \mathbf{s}_\alpha \quad \dots \quad \left(P^{(r)}_{h^{q-1}} + X_{h^{q-1}} \right) \mathbf{s}_\alpha \right] \dot{\mathbf{h}}, \\
=& \mathbf{x}^T P^{-1}_{h^q} \bar{Q}_{h^{q-1}\alpha} \dot{\mathbf{h}}, \\
=& \frac{1}{2} \left(\mathbf{x}^T P^{-1}_{h^q} \bar{Q}_{h^{q-1}\alpha} \dot{\mathbf{h}} + \dot{\mathbf{h}}^T \bar{Q}^T_{h^{q-1}\alpha} P^{-1}_{h^q} \mathbf{x} \right),
\end{aligned}$$

with $\bar{Q}_{h^{q-1}\alpha}$ given by equation (6.9).

By using these last developments in the Lyapunov function's time derivative equation leads to

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T P^{-1}_{h^q} \mathbf{x} + \mathbf{x}^T P^{-1}_{h^q} \dot{\mathbf{x}} - \frac{1}{2} \left(\mathbf{x}^T P^{-1}_{h^q} \bar{Q}_{h^{q-1}\alpha} \dot{\mathbf{h}} + \dot{\mathbf{h}}^T \bar{Q}^T_{h^{q-1}\alpha} P^{-1}_{h^q} \mathbf{x} \right).$$

By using the relations that

$$\begin{aligned}
\dot{\mathbf{x}} =& A_h \mathbf{x} + B_h \mathbf{u}, \\
\mathbf{u} =& \left(K_{h^q} + F_{h^q} \left(I - \frac{1}{2} J_g^T \bar{Q}^T_{h^{q-1}\alpha} \right) \right) P^{-1}_{h^q} \mathbf{x}, \\
\dot{\mathbf{h}} =& J_g \dot{\mathbf{x}},
\end{aligned}$$

we get that

$$\begin{aligned}
\dot{\mathbf{x}} =& \left(A_h + B_h \left(K_{h^q} + F_{h^q} \left(I - \frac{1}{2} J_g^T \bar{Q}^T_{h^{q-1}\alpha} \right) \right) P^{-1}_{h^q} \right) \mathbf{x}, \\
\dot{\mathbf{h}} =& J_g \left(A_h + B_h \left(K_{h^q} + F_{h^q} \left(I - \frac{1}{2} J_g^T \bar{Q}^T_{h^{q-1}\alpha} \right) \right) P^{-1}_{h^q} \right) \mathbf{x},
\end{aligned}$$

such that sufficient conditions for the time derivative of the Lyapunov function to be negative definite are given by

$$\begin{aligned}
& P^{-1}_{h^q} A_h + (*) + P^{-1}_{h^q} \left(B_h \left(K_{h^q} + F_{h^q} \left(I - \frac{1}{2} J_g^T \bar{Q}^T_{h^{q-1}\alpha} \right) \right) + (*) \right) P^{-1}_{h^q} \\
& - \frac{1}{2} P^{-1}_{h^q} \bar{Q}_{h^{q-1}\alpha} J_g \left(A_h + B_h \left(K_{h^q} + F_{h^q} \left(I - \frac{1}{2} J_g^T \bar{Q}^T_{h^{q-1}\alpha} \right) \right) P^{-1}_{h^q} \right) - (*) < 0,
\end{aligned}$$

in which $(*)$ represents the transpose of the term just before it.

By multiplying it left and right by P_{h^q}, we get

$$\begin{aligned}
& A_h P_{h^q} + (*) + B_h \left(K_{h^q} + F_{h^q} \left(I - \frac{1}{2} J_g^T \bar{Q}^T_{h^{q-1}\alpha} \right) \right) + (*) \\
& - \frac{1}{2} \bar{Q}_{h^{q-1}\alpha} J_g \left(A_h P_{h^q} + B_h \left(K_{h^q} + F_{h^q} \left(I - \frac{1}{2} J_g^T \bar{Q}^T_{h^{q-1}\alpha} \right) \right) \right) - (*) < 0, \\
& \left(I - \frac{1}{2} \bar{Q}_{h^{q-1}\alpha} J_g \right) \left(A_h P_{h^q} + B_h K_{h^q} \right) + (*) \\
& + \left(I - \frac{1}{2} \bar{Q}_{h^{q-1}\alpha} J_g \right) \left(B_h F_{h^q} + F^T_{h^q} B^T_h \right) \left(I - \frac{1}{2} J_g^T \bar{Q}^T_{h^{q-1}\alpha} \right) < 0.
\end{aligned}$$

From the matrix inversion lemma, we get that

$$\left(I - \frac{1}{2} J_g^T \bar{Q}_{h^{q-1}\alpha}^T\right)^{-1} = I + J_g^T \left(2I - \bar{Q}_{h^{q-1}\alpha}^T J_g^T\right)^{-1} \bar{Q}_{h^{q-1}\alpha}^T$$

and by multiplying the last inequality we had on the left by $\left(I - \frac{1}{2}\bar{Q}_{h^{q-1}\alpha} J_g\right)^{-1}$ and on the right by $\left(I - \frac{1}{2} J_g^T \bar{Q}_{h^{q-1}\alpha}^T\right)^{-1}$, we get

$$(A_h P_{h^q} + B_h K_{h^q} + B_h F_{h^q}) + (*) + (A_h P_{h^q} + B_h K_{h^q}) J_g^T \left(2I - \bar{Q}_{h^{q-1}\alpha}^T J_g^T\right)^{-1} \bar{Q}_{h^{q-1}\alpha}^T + (*) < 0,$$

which can be recast in the form ${L^{\perp}}^T \Delta L^{\perp} < 0$, with

$$\Delta = \begin{bmatrix} A_h P_{h^q} + B_h (K_{h^q} + F_{h^q}) + (*) & * \\ J_g \left(P_{h^q} A_h^T + K_{h^q}^T B_h^T\right) & 0 \end{bmatrix},$$

$$L^{\perp} = \begin{bmatrix} I \\ \left(2I - \bar{Q}_{h^{q-1}\alpha}^T J_g^T\right)^{-1} \bar{Q}_{\alpha h^{q-1}}^T \end{bmatrix} < 0.$$

By noting that $L^{\perp}$ is a base to the null space of

$$L = \begin{bmatrix} \bar{Q}_{h^{q-1}\alpha}^T & -\left(2I - \bar{Q}_{h^{q-1}\alpha}^T J_g^T\right) \end{bmatrix},$$

and by making use of Finsler's Lemma statements (2) and (3), another equivalent condition would be

$$\Delta + M^T L + L^T M < 0. \tag{6.11}$$

By choosing a specific M with the structure,

$$M = \begin{bmatrix} 0 & \mu I \end{bmatrix},$$

we get that a sufficient condition for the time derivative of the Lyapunov function to be negative definite is given by

$$\begin{bmatrix} A_h P_{h^q} + P_{h^q} A_h^T + B_h (K_{h^q} + F_{h^q}) + \left(K_{h^q}^T + F_{h^q}^T\right) B_h^T & * \\ J_g \left(P_{h^q} A_h^T + K_{h^q}^T B_h^T\right) + \mu \bar{Q}_{h^{q-1}\alpha} & -\mu \left(4I - \bar{Q}_{h^{q-1}\alpha}^T J_g^T - J_g \bar{Q}_{h^{q-1}\alpha}\right) \end{bmatrix} < 0.$$

Note, however, that, when we defined the design variable $\mathbf{s}$ and chose a design region, our conditions became only valid inside of the region $\mathbf{x} = P_{h^q}\mathbf{s}$, with $\mathbf{s}$ given by (6.5). In that regard, we will search for the "largest" P_{h^q} matrices such that our conditions are valid in the largest region possible. In order to do that, instead of solving a feasiblilty problem, we will solve a maximization problem over β and add the conditions

$$\beta > 0,$$
$$P_{h^q} > \beta I.$$

In doing so, not only do we search for the "largest" P_{h^q} matrices, but also assure that the Lyapunov function is positive definite. □

Remark 6.2

We have $\frac{(r+q-1)!}{(r-1)!(q-1)!}\left(\frac{n}{r}+\frac{(n+r)n_\alpha n_g(r+q)}{q(q+1)}\right)+1$ rows of LMIs in Theorem 6.2 when we employ Lemma 2.4 to implement it, as well as $1+r^{q-1}n\left(\frac{r(n+1)}{2}+n+2rm\right)$ scalar decision variables . In these relations, n is the number of states, m the number of controlled inputs, r the number of rules of the TS model of the system, n_α the number of vertices used to describe the design region, q the number of fuzzy summations in the Lyapunov function and control gains, and n_g the number of vertices used to describe the Jacobian of the membership functions vector.

Remark 6.3

Note that if we take $P_{h^q} = P$, and $X_{h^{q-1}} = -P^{(k)}_{h^{q-1}}$ (note that $P^{(k)}_{h^{q-1}}$ does not depend on k when $P_{h^q} = P$), we will have $\bar{Q}_{h^{q-1}\alpha} = 0$ in Theorem 6.2. In this case, if we take a large enough value of μ, we will recover the quadratic stabilization conditions.

In order to avoid having to find the largest Lyapunov sublevel set by inspection, we make use of a set of LMI conditions that find an estimate of the largest sublevel set that is guaranteed to be inside of the region under which the conditions presented are valid. These conditions are presented in the following lemma.

Lemma 6.3

Given a Lyapunov function of the form $\mathbf{s}^T P_{h^q}\mathbf{s} = \mathbf{x}^T P_{h^q}^{-1}\mathbf{x}$, and a polytopic region described by $|\mathbf{a}_i^T\mathbf{s}| \leq b_i,\ i \in \{1,2,\ldots,n_{ineq}\}$, the solution of the LMI optimization problem

$$\max \gamma,$$
$$\begin{bmatrix} -b_i^2 & \sqrt{\gamma}\mathbf{a}_i^T \\ * & -P_{h^q} \end{bmatrix} < 0, \quad \forall i \in \{1,2,\ldots,n_{ineq}\},$$

gives a maximum guaranteed sublevel set of the Lyapunov function ($\mathbf{s}^T P_{h^q}\mathbf{s} \leq \gamma$) that fits inside of the polytopic region.

Proof. Consider that the region in $\mathbf{s}$ can be written, in an inequality representation, as $|\mathbf{a}_i^T\mathbf{s}| \leq b_i$, and we want to find the maximum γ such that $\mathbf{s}^T P_{h^q}\mathbf{s} \leq \gamma$ is guaranteed to be inside the limiting region.

Since $P_{h^q} > 0$, there exists a matrix $R(\mathbf{x})$ such that $P_{h^q} = R(\mathbf{x})^T R(\mathbf{x})$, therefore, an equivalent representation of the Lyapunov sublevel set is $\mathbf{s} = R^{-1}(\mathbf{x})\mathbf{y}$, $\|\mathbf{y}\|_2 \leq \sqrt{\gamma}$.

By using the Cauchy-Schwarz inequality on the limiting region we get that

$$\begin{aligned} |\mathbf{a}_i^T\mathbf{s}| &\leq \|\mathbf{a}_i^T R^{-1}(\mathbf{x})\|_2\|\mathbf{y}\|_2, \\ &\leq \sqrt{\gamma}\|\mathbf{a}_i^T R^{-1}(\mathbf{x})\|_2. \end{aligned}$$

Then, a sufficient condition for the Lyapunov sublevel set to fit inside the desired region is that

$$\begin{aligned} \sqrt{\gamma}\|\mathbf{a}_i^T R^{-1}(\mathbf{x})\|_2 &\leq b_i, \\ \gamma\mathbf{a}_i^T R^{-1}(\mathbf{x})R^{-T}(\mathbf{x})\mathbf{a}_i &\leq b_i^2, \\ \gamma\mathbf{a}_i^T P_{h^q}^{-1}\mathbf{a}_i &\leq b_i^2, \end{aligned}$$

and, using Schur's complement,

$$\begin{bmatrix} -b_i^2 & \sqrt{\gamma}\mathbf{a}_i^T \\ * & -P_{h^q} \end{bmatrix} < 0.$$

□

Remark 6.4

By making use of Lemma 2.4 to implement Lemma 6.3 we have $\frac{(r+q-1)!}{r!(q-1)!}n_{\text{ineq}}(n+1)$ rows of LMIs and one scalar decision variable, with n the number of states, r the number of rules of the Lyapunov function, n_{ineq} the number of inequalities used to described the region, and q the number of fuzzy summations in the Lyapunov function.

Remark 6.5

It is important to observe that, these exact same conditions could also be applied to the stability case to find an estimate of the largest Lyapunov sublevel set that fits inside of the region. Even though the results obtained may be a little more conservative than the ones obtained by means of inspection, the estimate found is guaranteed to be contained within the polytopic region and usually yield good results (with optimal results if we have a quadratic Lyapunov function). In addition, it is a lot simpler to employ Lemma 6.3 than to try and find the largest Lyapunov sublevel set by inspection for systems with more than 3 states (since we can't draw the state space in this case).

Remark 6.6

Note that conditions from Lemma 6.3 could easily be incorporated directly into either Theorems 6.1 or 6.2. However, by doing them separately, we reduce the computational cost (as solving these optimization problems probably does not scale linearly with the number of conditions).

Example 6.3. Consider the two rule system given by

$$A_1 = \begin{bmatrix} 4 & -4 \\ -1 & -2 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 1 \\ 10 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -2 & -4 \\ 20 & -2 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 1 \\ -8 \end{bmatrix}.$$

with membership functions

$$h_1(\mathbf{x}) = \frac{1+\sin(x_1)}{2}, \qquad h_2(\mathbf{x}) = 1 - h_1(\mathbf{x}).$$

The Jacobian of the membership functions' is then given by

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\cos(x_1)}{2} & 0 \\ -\frac{\cos(x_1)}{2} & 0 \end{bmatrix}$$

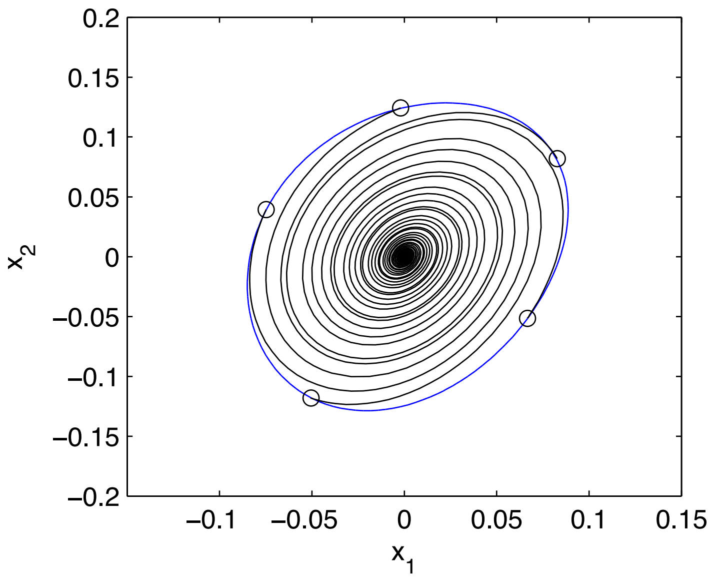

Figure 6.4: Largest domain of attraction found for the closed loop in Example 6.3 using Theorem 6.2 with $q = 2$, $\mu = 1.4$ and Lemma 6.3, as well as the system's trajectory from different initial conditions.

which can be globally described by the vertex matrices

$$J_1 = \begin{bmatrix} 0.5 & 0 \\ -0.5 & 0 \end{bmatrix}, \qquad J_2 = \begin{bmatrix} -0.5 & 0 \\ 0.5 & 0 \end{bmatrix},$$

with membership functions

$$g_1(\mathbf{x}) = \frac{1 + \cos x_1}{2}, \qquad g_2(\mathbf{x}) = \frac{1 - \cos x_1}{2}.$$

We consider a region in $\mathbf{s}$ given by $\|\mathbf{s}\|_1 \leq 1$, which can be described by the vertices

$$\mathbf{s}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \mathbf{s}_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \qquad \mathbf{s}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad \mathbf{s}_4 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

and also as $|\mathbf{a}_i^T \mathbf{s}| \leq b_i$ with

$$\mathbf{a}_1 = \begin{bmatrix} -1 & 1 \end{bmatrix}^T, \qquad b_1 = 1,$$
$$\mathbf{a}_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}^T, \qquad b_2 = 1.$$

Using Theorem 6.2 with the design region in $\mathbf{s}$ above, $q = 2$ and $\mu = 1.4$, we are able to find a control law that locally stabilizes the system. Figure 6.4 presents the estimation of the domain of attraction of the Lyapunov function found by using Lemma 6.3 as well as the system's trajectory in closed loop for different initial conditions. Figure 6.5 presents the domains of attraction found by using Theorem 6.2 with $q = 2$, $q = 4$ and $q = 6$. It is worth noting that the conditions in (Pan et al. 2012, Theorem 2) and (Lee and Kim 2014, Theorem 2) (with $q \leq 6$) were found to be unfeasible for this example.

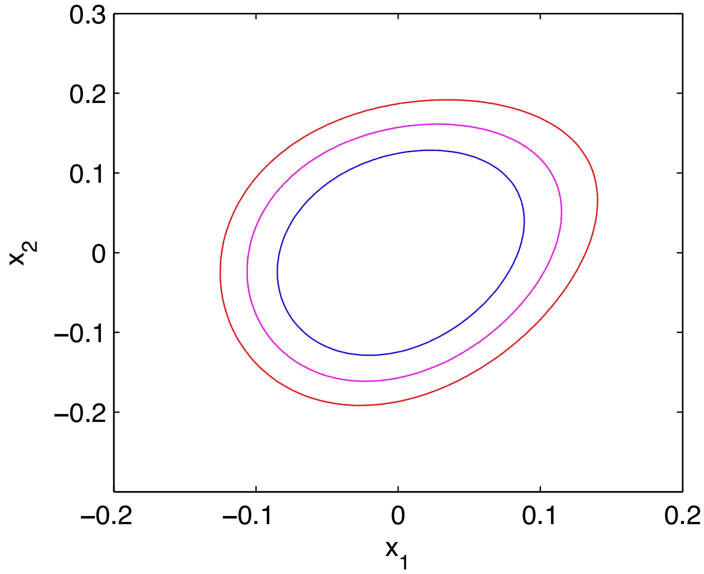

Figure 6.5: Different domains of attraction found in Example 6.3 with Theorem 6.2. The outermost region is the one found with $q = 6$. The middle one is the one found with $q = 4$. The innermost region is the one found with $q = 2$.

7 LMI synthesis conditions using piecewise-like Lyapunov functions

Even though we showed that, by making use of local transformation matrices, piecewise Lyapunov functions can yield very nice stability analysis conditions (as shown in chapter 5), they are usually too cumbersome to be used in synthesis conditions (we usually end up with Bilinear Matrix Inequalities (BMIs) (Feng et al. 2005), or we need to augment the system with virtual stable dynamics that allow us to invert the parameter matrices (Qiu, Feng, and Gao 2013)).

The work presented in this chapter was done under the supervision of Prof. João Pedro Hespanha.

7.1 Problem definition

Given a nonlinear system that can be described, in an hyperrectangle Ω of the state space including the origin, by a qLPV model

$$\dot{\mathbf{x}} = A(\mathbf{x})\mathbf{x} + B(\mathbf{x})\mathbf{u}, \tag{7.1}$$

with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $A(\mathbf{x})$ a continuous and bounded function from $\mathbb{R}^n$ to $\mathbb{R}^{n\times n}$, and $B(\mathbf{x})$ a continuous and bounded function from $\mathbb{R}^n$ to $\mathbb{R}^{n\times m}$, we would like to find a set of sufficient LMI conditions for the synthesis of a control law that renders the origin locally asymptotically stable.

Remark 7.1

Note that every input affine nonlinear system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + B(\mathbf{x})\mathbf{u},$$

with the origin as an equilibrium point, $\mathbf{f}(\mathbf{x})$ a continuous and bounded function, and $B(\mathbf{x})$ a continuous and bounded function, can be cast as a qLPV model as in (7.1). However, this representation is generally not unique.

7.2 Solution proposed

Given

$$\Omega = [\underline{x}_1, \bar{x}_1] \times [\underline{x}_2, \bar{x}_2] \times \cdots \times [\underline{x}_n, \bar{x}_n],$$

define a partition of each interval $[\underline{x}_i, \bar{x}_i]$ as

$$[\underline{x}_i, \bar{x}_i] = \bigcup_{k=1}^{r_i} \omega_k^{(i)},$$
$$\omega_k^{(i)} = \begin{cases} [\underline{x}_i + (k-1)\delta_i, \underline{x}_i + k\delta_i), & \text{if } k < r_i, \\ [\underline{x}_i + (r_i - 1)\delta_i, \bar{x}_i], & \text{if } k = r_i, \end{cases}$$

with r_i the number of desired subintervals and $\delta_i = \dfrac{(\bar{x}_i - \underline{x}_i)}{r_i}$, so that

$$\Omega = \bigcup \Omega_{i_1 i_2 \ldots i_n},$$
$$\Omega_{i_1 i_2 \ldots i_n} = \omega_{i_1}^{(1)} \times \omega_{i_2}^{(2)} \times \cdots \times \omega_{i_n}^{(n)}. \tag{7.2}$$

This amounts to dividing Ω into $r = \prod_{i=1}^{n} r_i$ hyperrectangles.

Remark 7.2

By defining a specific ordering of the indices $(i_1, i_2, \ldots, i_n)$ (or a one-to-one relation between them and the set of positive integers less than or equal to r) a single index can be used to represent each region. Therefore, whenever it will not create confusion, a single index will be used to represent the region.

Inside each Ω_i region, the system in (7.1) may be exactly represented by a TS fuzzy model

$$\dot{\mathbf{x}} = \sum_{j=1}^{q_i} h_{ij}(\mathbf{x}) \left[A_{ij}\mathbf{x} + B_{ij}\mathbf{u} \right],$$
$$= A_{h_i}\mathbf{x} + B_{h_i}\mathbf{u},$$

with q_i the number of linear models composing the TS model, $h_{ij}(\mathbf{x})$ the system's membership functions with

$$\sum_{j=1}^{q_i} h_{ij}(\mathbf{x}) = 1, \qquad h_{ij}(\mathbf{x}) \geq 0, \qquad \sum_{j=1}^{q_i} \dot{h}_{ij}(\mathbf{x}) = 0,$$

and A_{h_i} and B_{h_i} short notations for the fuzzy summations over the matrices composing the model.

By making use of the Ω_i regions' indicator functions, the TS models can be combined to represent the system in the Ω region as

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} \theta_i(\mathbf{x}) \left[A_{h_i}\mathbf{x} + B_{h_i}\mathbf{u} \right] = A_{\theta h_i}\mathbf{x} + B_{\theta h_i}\mathbf{u}, \tag{7.3}$$

with

$$\theta_i(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \Omega_i, \\ 0, & \text{otherwise.} \end{cases}$$

Remark 7.3

Note that the representation in (7.3) is simply a TS representation of the piecewise TS model found. Therefore a straightforward way of deriving less conservative LMI conditions for the system would be to make use of a piecewise Lyapunov function. However, when deriving synthesis conditions, the special structure imposed over the parameter matrices hinders their use in continuous time systems (they imply in either getting BMI conditions (Bernal, Guerra, and Kruszewski 2009), or augmenting the system with artificial dynamics (Qiu, Feng, and Gao 2013)). In that regard, this chapter proposes the use of a piecewise-like Lyapunov function (which is a fuzzy Lyapunov function whose premise variables are different from the system's, but have some special properties) to deal with these piecewise systems. In order to deal with the time derivative of the Lyapunov function's membership functions, it adapts the ideas from (Lee and Kim 2014; Lee, Park, and Joo 2012) to derive sufficient conditions that guarantee that a bound over the time derivatives will be respected in closed loop.

Remark 7.4

A piecewise TS model could also be obtained from finding a single TS model for the system and applying the local transformation proposed in chapter 5 inside each of the Ω_i regions, or applying the transformation proposed in (Bernal, Guerra, and Kruszewski 2009) in the regions induced by the ordering relations of the membership functions.

Due to the nature of the partitions proposed in this chapter, the Lyapunov function's membership functions can be defined with a tensor product structure, *i.e.* each partition's membership function is given by the product of single variable membership functions relating to each of the intervals that compose the partition. This particular structure eases the proposition of the membership functions since it is simply a combination of the membership functions devised for each of the state variables separately and endows the region's function with some interesting properties. Namely, if its written as

$$g_i(\mathbf{x}) = \prod_{k=1}^{n} \alpha_{i_k}^{(k)}(x_k),$$

and

$$\begin{aligned} \sum_{j=1}^{r_k} \alpha_j^{(k)}(x_k) &= 1, \\ \alpha_j^{(k)}(x_k) &\geq 0, \\ \alpha_j^{(k)}(x_k) &\neq 0 \text{ only if } x_k \in \omega_j^{(k)} \text{ or neighbours}(\omega_j^{(k)}), \end{aligned} \tag{7.4}$$

then

$$\begin{aligned} \sum_{i=1}^{r} g_i(\mathbf{x}) &= 1, \\ g_i(\mathbf{x}) &\geq 0, \\ g_i(\mathbf{x}) &\neq 0 \text{ only if } \mathbf{x} \in \Omega_i \text{ or neighbours}(\Omega_i), \end{aligned}$$

in which two partitions are said to be neighbours if they share at least one common vertex.

Considering membership functions with these properties, we can define a piecewise-like Lyapunov function as

$$V(\mathbf{x}) = \mathbf{x}^T \left(\sum_{i=1}^{r} g_i(\mathbf{x}) P_i \right)^{-1} \mathbf{x} = \mathbf{x}^T P_g^{-1} \mathbf{x} \tag{7.5}$$

and a *continuous* piecewise-like control law as

$$\mathbf{u} = \left(\sum_{i=1}^{r} g_i(\mathbf{x}) K_i \right) \left(\sum_{i=1}^{r} g_i(\mathbf{x}) P_i \right)^{-1} \mathbf{x} = K_g P_g^{-1} \mathbf{x},$$

with P_g^{-1} a short notation for $\left(\sum_{i=1}^{r} g_i(\mathbf{x}) P_i \right)^{-1}$ and K_g a short notation for $\left(\sum_{i=1}^{r} g_i(\mathbf{x}) K_i \right)$.

Remark 7.5

Defining this fuzzy Lyapunov function and control law leads to conditions one would expect to get from using a piecewise quadratic Lyapunov function. The main difference, however is that, instead of enforcing a special structure upon the P_i matrices guaranteeing the function's continuity (that, as stated before, leads to either BMI conditions or augmenting the system with virtual dynamics), one needs to deal with the membership functions' time derivative and the fact that they are also "active" on the neighbouring regions.

7.3 LMI conditions - generic membership functions

The Lyapunov function's time derivative is given by

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \dot{\mathbf{x}}^T P_g^{-1} \mathbf{x} + \mathbf{x}^T P_g^{-1} \dot{\mathbf{x}} + \mathbf{x}^T \dot{\overline{P_g^{-1}}} \mathbf{x}, \\ &= \dot{\mathbf{x}}^T P_g^{-1} \mathbf{x} + \mathbf{x}^T P_g^{-1} \dot{\mathbf{x}} - \mathbf{x}^T P_g^{-1} \dot{P}_g P_g^{-1} \mathbf{x}, \\ &= \mathbf{x}^T \left(A_{\theta h_i}^T P_g^{-1} + P_g^{-1} A_{\theta h_i} + P_g^{-1} \left(K_g^T B_{\theta h_i}^T + B_{\theta h_i} K_g - \dot{P}_g \right) P_g^{-1} \right) \mathbf{x}. \end{aligned}$$

Sufficient conditions to ensure the system's stability are then given by

$$\begin{aligned} &P_g^{-1} > 0, \\ &A_{\theta h_i}^T P_g^{-1} + P_g^{-1} A_{\theta h_i} + P_g^{-1} \left(K_g^T B_{\theta h_i}^T + B_{\theta h_i} K_g - \dot{P}_g \right) P_g^{-1} < 0. \end{aligned}$$

By performing a similarity transformation with P_g (*i.e* by multiplying it on the right by this value and on the left by its transpose),

$$\begin{aligned} &P_g > 0, \\ &P_g A_{\theta h_i}^T + A_{\theta h_i} P_g + K_g^T B_{\theta h_i}^T + B_{\theta h_i} K_g - \dot{P}_g < 0. \end{aligned}$$

which actually represent

$$\begin{aligned} &\sum_{i=1}^{r} g_i(\mathbf{x}) P_i > 0, \\ &\sum_{i=1}^{r} \sum_{j=1}^{r} \sum_{k=1}^{q_j} g_i(\mathbf{x}) \theta_j(\mathbf{x}) h_{jk}(\mathbf{x}) \left[P_i A_{jk}^T + A_{jk} P_i + K_i^T B_{jk}^T + B_{jk} K_i - \left(\sum_{\ell=1}^{r} \dot{g}_\ell(\mathbf{x}) P_\ell \right) \right] < 0. \end{aligned} \tag{7.6}$$

By noting that

$$
\begin{aligned}
&g_i(\mathbf{x})\theta_j(\mathbf{x}) \neq 0, && \text{only if } i = j \text{ or } i \text{ is a neighbor of } j,\\
&\dot{g}_i(\mathbf{x})\theta_j(\mathbf{x}) \neq 0, && \text{only if } i = j \text{ or } i \text{ is a neighbor of } j,
\end{aligned}
$$

the following sufficient conditions are sufficient to ensure the system's stability

$$
\begin{aligned}
&P_i > 0, \quad \forall i \in \{1,\dots,r\}\\
&P_i A_{jk}^T + A_{jk}P_i + K_i^T B_{jk}^T + B_{jk}K_i - \left(\sum_{\ell \in \mathbb{L}(j)} \dot{g}_\ell(\mathbf{x}) P_\ell\right) < 0, \quad \forall \begin{array}{l} j \in \{1,\dots,r\},\\ i \in \mathbb{L}(j),\\ k \in \{1,\dots,q_j\} \end{array}
\end{aligned}
$$

in which $\mathbb{L}(j)$ denotes the set composed by j and its neighbors.

Because of the membership functions g_i properties, their time derivatives have the interesting property that

$$
\sum_{\ell \in \mathbb{L}(j)} \dot{g}_\ell(\mathbf{x}) P_\ell = \sum_{\ell \in \mathbb{L}(j)} \dot{g}_\ell(\mathbf{x}) \left(P_\ell + X_j\right).
$$

Considering this property, one way to deal with the $\dot{g}_\ell$ terms is to consider that $|\dot{g}_\ell| \leq \phi_\ell$ and that $P_\ell + X_j > 0, \forall \ell \in \mathbb{L}(j)$, in that way

$$
-\sum_{\ell \in \mathbb{L}(j)} \dot{g}_\ell(\mathbf{x}) P_\ell \leq \sum_{\ell \in \mathbb{L}(j)} \phi_\ell \left(P_\ell + X_j\right) = P_{\phi_j}
$$

and the summation term can be replaced by P_{ϕ_j} to get LMI conditions.

In order to enforce the bound $|\dot{g}_\ell| \leq \phi_\ell$ in closed loop, the ideas from (Lee and Kim 2014; Lee, Park, and Joo 2012) are adapted to a piecewise scenario. The main idea in those papers was to make the square of the membership functions' time derivative less than ϕ_k^2 times the Lyapunov function and assuring that the 1-sublevel set of the Lyapunov function is inside the desired region. By doing that, the LMI conditions assures the system's stability only inside of the 1-sublevel set of the Lyapunov function (which also serves as the best estimate of the domain of attraction their conditions can find). In that regard, the authors search for the "largest" 1-sublevel set trying to find the largest domain of attraction possible with their conditions.

However, those ideas cannot be directly applied in this context because of the several different regions. In that regard, this work uses an auxiliary function in every region and forces it to be less than 1 for every value inside of the region (in that regard, one could think of the approach here as an outer approximation of the regions instead of the inner approximation used in (Lee and Kim 2014; Lee, Park, and Joo 2012)). The main drawback, however, is that we do not get an explicit approximation of the domain of attraction.

In that regard, we want that

$$
\frac{\dot{g}_\ell^2}{\phi_\ell^2} \leq \mathbf{x}^T P_g^{-1} Q_{g\theta} P_g^{-1} \mathbf{x} \leq 1, \quad \mathbf{x} \in \Omega_j, \ell \in \mathbb{L}(j). \tag{7.7}
$$

The first inequality can be written as

$$
\begin{aligned}
&\mathbf{x}^T \left(P_g^{-1} K_g^T B_{\theta h_i}^T + A_{\theta h_i}^T\right) \nabla g_\ell^T \phi_\ell^{-2} \nabla g_\ell \left(A_{\theta h_i} + B_{\theta h_i} K_g P_g^{-1}\right) \mathbf{x} \leq \mathbf{x}^T P_g^{-1} Q_{g\theta} P_g^{-1} \mathbf{x},\\
&\sum_{i=1}^{r} \sum_{j=1}^{r} \sum_{k=1}^{q_j} g_i(\mathbf{x})\theta_j(\mathbf{x}) h_{jk}(\mathbf{x}) \begin{bmatrix} -Q_{ij} & \left(P_i A_{jk}^T + K_i^T B_{jk}^T\right) \nabla g_\ell^T \\ * & -\phi_\ell^2 \end{bmatrix} < 0,
\end{aligned}
$$

for which a sufficient condition is that

$$\begin{bmatrix} -Q_{ij} & \left(P_i A_{jk}^T + K_i^T B_{jk}^T\right) \nabla g_\ell^T \\ * & -\phi_\ell^2 \end{bmatrix} < 0, \forall \begin{array}{l} j \in \{1,\dots,r\}, \\ i,\ell \in \mathbb{L}(j), \\ k \in \{1,\dots,q_j\} \end{array}$$

The second inequality can be written as

$$\begin{gathered} \mathbf{x}^T P_g^{-1} Q_{g\theta} P_g^{-1} \mathbf{x} \leq 1, \\ \begin{bmatrix} -1 & \mathbf{x}^T \\ * & -P_g Q_{g\theta}^{-1} P_g \end{bmatrix} < 0, \end{gathered}$$

and by using the fact that, for $Q_{g\theta} > 0$, $-P_g Q_{g\theta}^{-1} P_g < -2P_g + Q_{g\theta}$, sufficient conditions for it are

$$\begin{gathered} \begin{bmatrix} -1 & \mathbf{x}^T \\ * & -2P_g + Q_{g\theta} \end{bmatrix} < 0, \forall \mathbf{x} \in \Omega_j, \\ \sum_{i=1}^{r} \sum_{j=1}^{r} g_i(\mathbf{x}) \theta_j(\mathbf{x}) \begin{bmatrix} -1 & \mathbf{x}^T \\ * & -2P_i + Q_{ij} \end{bmatrix} < 0, \forall \mathbf{x} \in \Omega_j, \end{gathered}$$

Since Ω_j is a hyper-rectangle (and thus a polytope), a necessary and sufficient condition is that the condition above is fulfilled at the vertices of Ω_j. By remembering the remark about the membership functions only being different than zero in neighbouring regions, we then get that sufficient conditions to find a stabilizing controller are

$$\begin{aligned} & P_i > 0, \quad \forall i \in \{1,\dots,r\} \\ & P_\ell + X_j > 0, \forall j \in \{1,\dots,r\}, \ell \in \mathbb{L}(j) \\ & P_{\phi_j} = \sum_{\ell \in \mathbb{L}(j)} \phi_\ell \left(P_\ell + X_j\right) \\ & P_i A_{jk}^T + A_{jk} P_i + K_i^T B_{jk}^T + B_{jk} K_i + P_{\phi_j} < 0, \quad \forall \begin{array}{l} j \in \{1,\dots,r\}, \\ i \in \mathbb{L}(j), \\ k \in \{1,\dots,q_j\} \end{array} \\ & \begin{bmatrix} -Q_{ij} & \left(P_i A_{jk}^T + K_i^T B_{jk}^T\right) \nabla g_\ell^T \\ * & -\phi_\ell^2 \end{bmatrix} < 0, \forall \begin{array}{l} j \in \{1,\dots,r\}, \\ i,\ell \in \mathbb{L}(j), \\ k \in \{1,\dots,q_j\} \end{array} \\ & \begin{bmatrix} -1 & \mathbf{x}_v^T \\ * & -2P_i + Q_{ij} \end{bmatrix} < 0, \forall \begin{array}{l} j \in \{1,\dots,r\} \\ \mathbf{x}_v \in \mathbb{V}(j) \\ i \in \mathbb{L}(j) \end{array} \end{aligned}$$

with ϕ_ℓ given, $\mathbb{L}(j)$ the set composed by j and its neighbors, and $\mathbb{V}(j)$ the set of vertices composing Ω_j.

Remark 7.6

It is important to notice two things about the above conditions. First is that, since they are local conditions, they only ensure local stability inside of the largest Lyapunov sublevel-set that fits inside of Ω. Second is that, whenever we would like to apply these conditions, we need to choose the values of ϕ_ℓ that are going to upper bound the time derivative of our membership functions. A smaller value usually lead to less restrictive LMI conditions regarding the time derivative of the Lyapunov function (because they lead to a smaller P_{ϕ_j} term), however they also lead to harder restrictions upon the control gains (since we would have tighter bounds on the membership functions time derivative). A higher value of ϕ_ℓ would lead to the opposite situation. Notice also that as ϕ_ℓ tends to infinity, our conditions tend to the quadratic stability conditions (since we would be forced to choose all of the P_i with the same value and $X_j = -P_i$ and we would not impose any condition over the time derivative of the membership functions).

Remark 7.7

Note that the LMI conditions proposed in this work do not make explicit use of the system's membership function (so they give them, locally, the same treatment as polytopic uncertainties). A way to further relax the conditions could be employ the system's membership functions in the control law and in the Q matrices (at the cost of an increased number of LMIs and decision variables.

Remark 7.8

The advantages of the proposed solution in this work over others available on the literature is that, unlike piecewise Lyapunov function, we have LMI synthesis conditions, and, unlike the regular fuzzy Lyapunov function approach (using the system's membership function in the Lyapunov function), the system's membership function need not be continuously differentiable.

7.4 Choice of the membership functions

Notice that the choice of the membership functions used for the Lyapunov function and the control law directly affects the LMI stabilization conditions proposed above. Notice as well that, in order to truly have LMI conditions, we need to find a polytopic representation for them, similar to what we did to use $\mathbf{x}$ in the conditions.

Because of the tensor product structure, we get that

$$\begin{aligned} g_i(\mathbf{x}) &= \prod_{k=1}^{n} \alpha_{i_k}^{(k)}(x_k) \\ \nabla g_i(\mathbf{x}) &= \begin{bmatrix} \frac{d}{dx_1}\alpha_{i_1}^{(1)}(x_1) \prod_{k=2}^{n} \alpha_{i_k}^{(k)}(x_k) & \dots & \frac{d}{dx_n}\alpha_{i_n}^{(n)}(x_n) \prod_{k=1}^{n-1} \alpha_{i_k}^{(k)}(x_k) \end{bmatrix}. \end{aligned} \tag{7.8}$$

Therefore, we would like to choose a set of $\alpha_{i_k}^{(k)}(x_k)$ whose derivative $\dfrac{d}{dx_k}\alpha_{i_k}^{(k)}(x_k)$ is small (so that we get less conservative LMI conditions).

In that regard, we can write an optimization problem to search for an optimal "worst case" set of membership functions. That is, we can write an optimization program to search for a set of $\alpha_j^{(k)}(x_k)$ membership functions respecting (7.4) and minimizing the maximum norm of their derivative. In other words,

$$
\begin{aligned}
\min\, & t, \\
& \text{s.t.} \\
& \quad \sum_{j=1}^{r_k} \alpha_j^{(k)}(x_k) = 1, \\
& \quad \alpha_j^{(k)}(x_k) \geq 0, \\
& \quad \alpha_j^{(k)}(x_k) \neq 0 \text{ only if } x_k \in \omega_j^{(k)} \text{ or neighbours}(\omega_j^{(k)}), \\
& \quad \left| \frac{d}{dx_k} \alpha_j^{(k)}(x_k) \right| < t, \quad \forall x_k \in [\underline{x}_k, \bar{x}_k], \forall j.
\end{aligned}
\tag{7.9}
$$

By approximating the membership functions as piecewise linear functions, the search for the membership functions can be replaced with the search for points along $[\underline{x}_k, \bar{x}_k]$, and the inequality involving the norm of the derivative (which now will only be valid almost everywhere), can be replaced by the norm of the difference between a point and its neighbour. With this approximation, the optimization problem presented above becomes a linear optimization problem and can be efficiently solved for a large number of points in $[\underline{x}_k, \bar{x}_k]$.

An important step when doing this approximation is ensuring that the points on the borders of two ω_j regions is always chosen by the optimization problem, in order to guarantee that the neighbouring constraint will be respected. Notice also that, due to the neighbouring constraint, we only need to seek the values of the membership functions when they are active (*i.e.* when they are in their own regions, or in a neighbouring region).

Figure 7.1 presents the result obtained for $x_k \in [-5, 5]$ with 2, 3, 4, 5, 6, and 7 regions. Note that, if we have less than 4 regions in x_k, it is better to assume a single region in that variable (since, just like in the LMI conditions presented, it does not make a difference). For that reason, in this work, we suppose that either $r_k = 1$ or $r_k > 3$.

Note also that, from 4 regions upwards, the shape of the membership functions is fixed. The membership functions that are not active in the limit regions (*i.e.* the ones that are zero in -5 and 5 in the figure) all have a trapezoidal shape, whereas the others have a different shape (a little strange, probably due to a "border effect"). However, the derivative in these regions are all less than the worst case, and we can impose a trapezoidal structure to these regions as well without changing the "worst case" derivative.

Figure 7.2 presents the alternative optimal membership functions obtained for $x_k \in [-5, 5]$ with 4, 5, 6, and 7 regions. Note that these membership functions are simpler to implement and grant our LMI synthesis conditions with some interesting properties (that will be presented in the following section).

The optimization problem used to find these membership functions does not depend on the range used for x_k, but rather it depends only on the number of regions used. In that case, we can say, considering that $j \in 1, 2, \ldots, r_k$ and $r_k > 3$, that an almost everywhere differentiable solution to the optimization problem in (7.9) is given by

$$\alpha_j^{(k)}(x_k) = \begin{cases} 0.5, & \text{if } \begin{cases} x_k \in \omega_j^{(k)}, \\ x_k \in \omega_1^{(k)},\ j = 2, \\ x_k \in \omega_{r_k}^{(k)},\ j = r_k - 1, \end{cases} \\ \dfrac{x_k - x_{k(j-1)}}{x_{k(j-1)} - x_{k(j-2)}}, & \text{if } x_k \in \omega_{j-1}^{(k)},\ j > 2, \\ \dfrac{-x_k + x_{k(j+2)}}{x_{k(j+2)} - x_{k(j+1)}}, & \text{if } x_k \in \omega_{j+1}^{(k)},\ j < r_k - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{7.10}$$

with $x_{kj} = \underline{x_k} + (j-1)\delta_k$, $\delta_k = \dfrac{(\bar{x}_k - \underline{x}_k)}{r_k}$, and its derivative, for $x_k \neq x_{km}$ with $m \in 1, 2, \ldots, r_k$, is given by

$$\frac{d\alpha_j^{(k)}}{dx_k}(x_k) = \begin{cases} \dfrac{1}{\delta_k}, & \text{if } x_k \in \omega_{j-1}^{(k)},\ j > 2, \\ \dfrac{-1}{\delta_k}, & \text{if } x_k \in \omega_{j+1}^{(k)},\ j < r_k - 1, \\ 0, & \text{otherwise.} \end{cases}$$

For the simple case, in which $r_k = 1$, we have

$$\alpha_1^{(k)}(x_k) = 1, \qquad \frac{d\alpha_1^{(k)}}{dx_k}(x_k) = 0. \tag{7.11}$$

Remembering from (7.8) that

$$\frac{\partial g_i}{\partial x_k}(\mathbf{x}) = \frac{d}{dx_k}\alpha_{i_k}^{(k)}(x_k) \prod_{j=1, j\neq k}^{n} \alpha_{i_j}^{(j)}(x_j)$$

and noting that, since the maximum of $\alpha_{i_j}^{(j)}(x_j)$ is 1 if $r_j = 1$, or 0.5 if $r_j > 3$,

$$\prod_{j=1, j\neq k}^{n} \alpha_{i_j}^{(j)}(x_j) \in [0, (0.5)^{\kappa_k}]$$

with κ_k being the number of regions, aside from k, that have more than 3 regions. We have that

$$\frac{\partial g_i}{\partial x_k}(\mathbf{x}) \in \begin{cases} \mathbf{co}\left\{ 0,\ \dfrac{(0.5)^{\kappa_k}}{\delta_k} \right\}, & \text{if } x_k \in \omega_{i_k-1}^{(k)},\ i_k > 2, \\ \mathbf{co}\left\{ \dfrac{-(0.5)^{\kappa_k}}{\delta_k},\ 0 \right\}, & \text{if } x_k \in \omega_{i_k+1}^{(k)},\ i_k < r_k - 1, \\ 0, & \text{otherwise,} \end{cases} \tag{7.12}$$

which, as desired in the beginning of this section, allow us to write a polytopic representation for ∇g_i. Note that this expression for the gradient will not be valid for $x_k = x_{km}$ with $m \in 1, 2, \ldots, r_k$ for every k, which is a zero Lebesgue measure set, and thus our expression in only valid almost everywhere.

It is important to note that, even though $\alpha_j^{(k)}(x_k)$ is not continuously differentiable, it is bounded and Lipschitz continuous (with Lipschitz constant $1/\delta_k$). This ensures that the $g_i(\mathbf{x})$ membership functions are also Lipschitz continuous and that the Lyapunov in (7.5) is locally Lipschitz continuous.

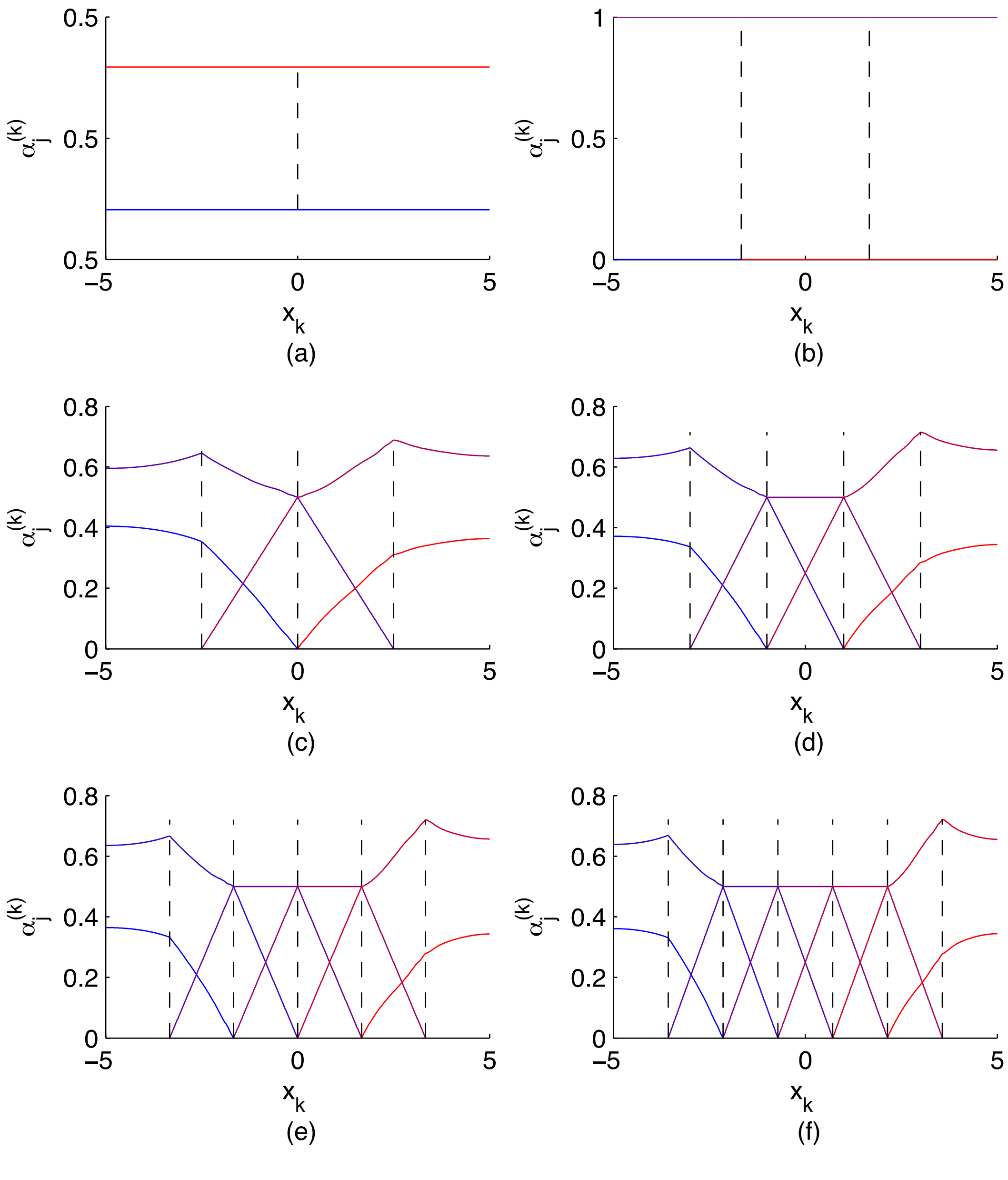

Figure 7.1: Optimal membership functions for $x_k \in [-5, 5]$ with 2, 3, 4, 5, 6, and 7 regions in (a), (b), (c), (d), (e), and (f) respectively.

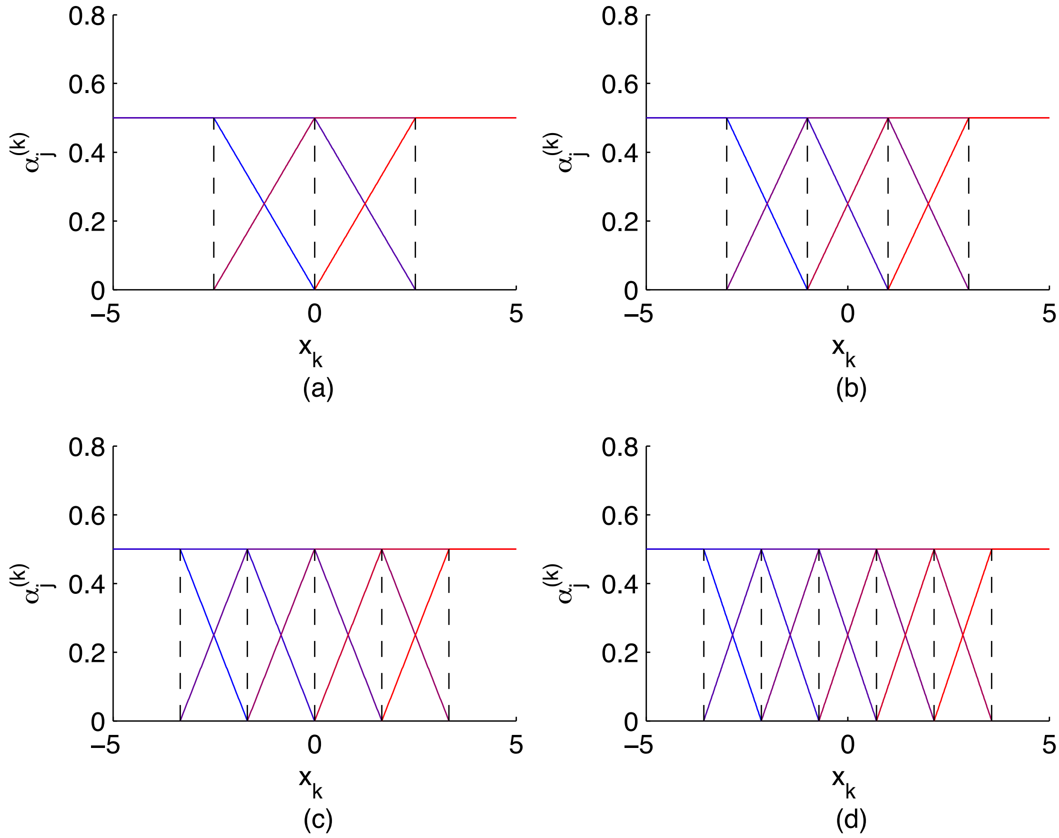

Figure 7.2: Alternative optimal membership functions for $x_k \in [-5, 5]$ with 4, 5, 6, and 7 regions in (a), (b), (c), and (d) respectively.

7.5 LMI conditions - optimal membership functions

In this section, we present how to take further advantage from the structure on the optimal membership functions found in the previous section to further relax the LMI conditions proposed in section 7.3.

First, note that, due to the trapezoidal shape of the membership functions, we have that

$$\dot{g}_i(\mathbf{x})\theta_i(\mathbf{x}) = 0,$$

so that the summation $\sum_{j=1}^{r}\sum_{\ell=1}^{r}\theta_j((x))\dot{g}_\ell(\mathbf{x})P_\ell$ in (7.6) can now be replaced by $\sum_{j=1}^{r}\sum_{\ell\in\mathbb{S}(j)}\theta_j((x))\dot{g}_\ell(\mathbf{x})P_\ell$, with $\mathbb{S}(j)$ the set of neighbours of j. That means that now we use the set $\mathbb{S}(j)$, instead of $\mathbb{L}(j)$, in the definition of P_{ϕ_j} and in $P_\ell + X_j > 0$.

Second, note that, from (7.8) we can write (7.12) as

$$\frac{\partial g_i}{\partial x_k}(\mathbf{x}) = \lambda_{ik}(\mathbf{x})(0.5)^{\kappa_k}\frac{d}{dx_k}\alpha_{i_k}^{(k)}(x_k)$$

with $\lambda_{ik}(\mathbf{x}) \in [0, 1]$. But this implies that

$$\begin{aligned}\dot{g}_\ell^2 &= \sum_{k=1}^{n}\left(\dot{x}_k\frac{\partial g_\ell}{\partial x_k}\right)^2 = \sum_{k=1}^{n}\left(\dot{x}_k\lambda_{\ell k}(0.5)^{\kappa_k}\frac{d}{dx_k}\alpha_{\ell_k}^{(k)}\right)^2 = \sum_{k=1}^{n}\dot{x}_k^2\lambda_{\ell k}^2(0.5)^{2\kappa_k}\left(\frac{d}{dx_k}\alpha_{\ell_k}^{(k)}\right)^2\\ &\leq \sum_{k=1}^{n}\dot{x}_k^2(0.5)^{2\kappa_k}\left(\frac{d}{dx_k}\alpha_{\ell_k}^{(k)}\right)^2 = \sum_{j=1}^{r}\theta_j(\mathbf{x})\dot{\mathbf{x}}^T\boldsymbol{\nu}_{\ell j}\boldsymbol{\nu}_{\ell j}^T\dot{\mathbf{x}}.\end{aligned}$$

with $\boldsymbol{\nu}_{\ell j}$ a vector whose k-th element, $\nu_{\ell jk}$, given by

$$\nu_{\ell jk} = \begin{cases} \dfrac{(0.5)^{\kappa_k}}{\delta_k}, & \text{if } \ell_k > 2,\ \ell_k = j_k - 1, \\ \dfrac{-(0.5)^{\kappa_k}}{\delta_k}, & \text{if } \ell_k < r_k - 1,\ \ell_k = j_k + 1, \\ 0, & \text{otherwise,} \end{cases} \tag{7.13}$$

and this means that we can replace ∇g_ℓ^T by $\boldsymbol{\nu}_{\ell j}$ in the LMIs in order to get sufficient conditions for (7.7) at the points where g_ℓ is differentiable.

Note that, with the optimal membership functions, our LMI conditions only assure that $\nabla V \dot{\mathbf{x}} < 0$ almost everywhere in the state space. But, since $\dot{\mathbf{x}}$ is continuous (because of our assumptions and the fact that we are using a continuous control law) and $V(\mathbf{x})$ is locally Lipschitz continuous, by applying (Hespanha, Liberzon, and Teel 2008, Lemma 1) we get that $V(t)$ is absolutely continuous and $\dot{V}(\mathbf{x}(t)) < 0$ almost everywhere on t (which implies that the system is locally asymptotically stable).

Having discussed all of this, we can enunciate the following theorem.

Theorem 7.1: LMI synthesis conditions for piecewise-like Lyapunov functions

Given a system described by (7.3) *in a region given by* Ω *in* (7.2)*, and scalars* $\phi_i > 0$*, if there exists matrices* P_i*,* K_i*, and* Q_{ij} *solving the LMI feasibility problem*

$$P_i > 0, \quad \forall i \in \{1, \dots, r\}$$
$$P_\ell + X_j > 0, \forall j \in \{1, \dots, r\}, \ell \in \mathbb{S}(j)$$
$$P_{\phi_j} = \sum_{\ell \in \mathbb{S}(j)} \phi_\ell \left(P_\ell + X_j\right)$$

$$P_i A_{jk}^T + A_{jk} P_i + K_i^T B_{jk}^T + B_{jk} K_i + P_{\phi_j} < 0, \quad \forall \begin{array}{l} j \in \{1, \dots, r\}, \\ i \in \mathbb{L}(j), \\ k \in \{1, \dots, q_j\} \end{array}$$

$$\begin{bmatrix} -Q_{ij} & \left(P_i A_{jk}^T + K_i^T B_{jk}^T\right) \boldsymbol{\nu}_{\ell j} \\ * & -\phi_\ell^2 \end{bmatrix} < 0, \forall \begin{array}{l} j \in \{1, \dots, r\}, \\ i \in \mathbb{L}(j), \ell \in \mathbb{S}(j), \\ k \in \{1, \dots, q_j\} \end{array}$$

$$\begin{bmatrix} -1 & \mathbf{x}_v^T \\ * & -2P_i + Q_{ij} \end{bmatrix} < 0, \forall \begin{array}{l} j \in \{1, \dots, r\} \\ \mathbf{x}_v \in \mathbb{V}(j) \\ i \in \mathbb{L}(j) \end{array}$$

with $\boldsymbol{\nu}_{\ell j}$ *given by* (7.13)*, then the system is locally asymptotically stable with control law*

$$\mathbf{u} = \left(\sum_{i=1}^r g_i(\mathbf{x}) K_i\right) \left(\sum_{i=1}^r g_i(\mathbf{x}) P_i\right)^{-1} \mathbf{x}$$

and Lyapunov function

$$V(\mathbf{x}) = \mathbf{x}^T \left(\sum_{i=1}^r g_i(\mathbf{x}) P_i\right)^{-1} \mathbf{x} = \mathbf{x}^T P_g^{-1} \mathbf{x}$$

with $g_i(\mathbf{x}) = \prod_{k=1}^n \alpha_{i_k}^{(k)}(x_k)$ *and* $\alpha_{i_k}^{(k)}(x_k)$ *given by either* (7.10) *or* (7.11).

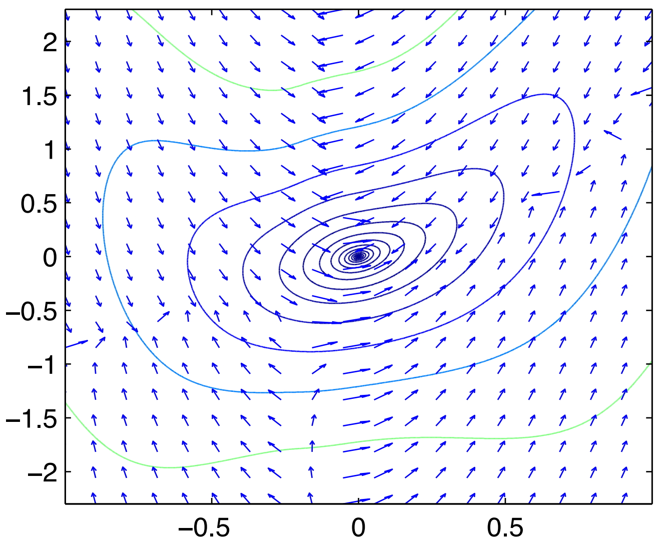

Figure 7.3: Phase plane of the system in example 7.1 with the control law found. The curves presented in the figure are Lyapunov level curves.

7.6 Example

In this section, a numerical example is presented to illustrate the proposed stabilization conditions.

Example 7.1 (Same example system used in Example 6.3)**.** Consider the nonlinear system given by equation (7.1) with

$$A(\mathbf{x}) = \begin{bmatrix} 1+3\sin(x_1) & -4 \\ \frac{19-21\sin(x_1)}{2} & -2 \end{bmatrix}, \qquad B(\mathbf{x}) = \begin{bmatrix} 1 \\ \frac{2+18\sin(x_1)}{2} \end{bmatrix},$$

and $x_1 \in [-1.2, 1.2]$, $x_2 \in [-2, 2]$.

Using the conditions from Theorem 7.1 with 401 partitions in x_1 and considering $[-2, 2]$ as the only region in x_2 (since there are only nonlinear terms in x_1, we have decided to only partition the state space in this variable) and choosing $\phi_i = \phi = 6000$, we are able to find a control law that locally stabilizes the system. Figure 7.3 presents the phase plane of the closed loop system with some of the Lyapunov level curves. Figure 7.4 presents an estimate of the closed loop domain of attraction (given by the largest Lyapunov sublevel set that fits inside of the given region) as well as several different trajectories for the system whose initial conditions are all on the border of the domain of attraction estimated.

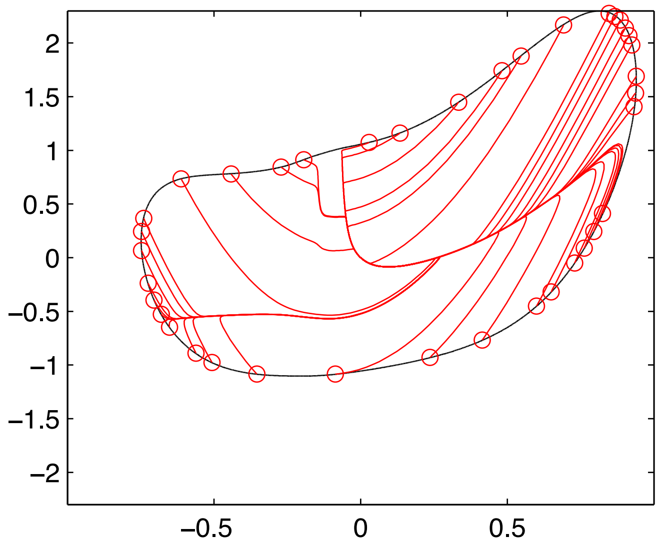

Figure 7.4: Estimated domain of attraction for the closed loop system in example 7.1. The red curves are the system's trajectories in closed loop, whereas the red balls are their initial conditions.

Part III

Conclusions

8 Discussion and Future Directions

8.1 Discussion

Exploiting the membership functions

In chapter 3 we presented two different partition strategies for the membership functions' image space in order to generate sufficient and asymptotically necessary conditions by means of Theorems 3.1 and 3.2.

Two examples were presented in order to illustrate the proposed methods. The first example dealt with the stabilization and comparing the proposed methods with others from the literature, whereas the second one focused in guaranteed $\mathcal{H}_\infty$ control and comparing the proposed methods among themselves.

It is interesting to note that the proposed partition strategies performed well when compared to the other strategies (despite having been outperformed by the combination of (Sala and Ariño 2007, Theorem 5) and (Sala and Ariño 2008, Proposition 3.1)) and that, overall, the second partition strategy performed better than the first one. However, that increase in performance comes in exchange for the lost of determinism of the partition strategy and conditions that are not "monotically less conservative" (*i.e.* conditions that are guaranteed to be always less conservative as the number of partitions increase).

Some possible enhancements for the partition strategy are:

- Defining a formal stop criterion to be used when the partitions are being used to generate sufficient and asymptotically necessary conditions (*i.e.* how to we check if the conditions have reached necessity). So far the criterion used is the number of iterations in the method;
- The use of heuristics, similar to (Kruszewski et al. 2009, Algorithm 2), in order to direct the division of the simplices (or groupings) so that more divisions are made in regions that affect/violate more the LMI conditions and the performance/iteration (or performance/number of simplices) ratio is increased;
- Defining a systematic and deterministic manner to initialize the center of the clusters used in the second strategy, making it deterministic without compromising performance;
- The conditions from the *Grouping the membership points* partition strategy could be made "monotically less conservative" if we were able to impose that the new simplices had to be contained inside of the previous ones;
- The conditions from the *Dividing and Discarding* partition strategy could converge faster if a different partition strategy were chosen for the simplicial partition step. An interesting alternative could be given by (Gonçalves et al. 2006).

We briefly touched the topic that, even though the conditions presented are sufficient and asymptotically necessary for the parametric LMIs, they might not be for the original quadratic conditions if any of the premise variables are functions of the state variables (due to the fact that in this case the parametric LMIs are only sufficient for the quadratic form).

In addition, we also showed that these partitions correspond to an exact switched representation of the TS model, and this representation inspired the use of the partitions to look for control laws that switch according to the membership functions. We also showed how these control law conditions could be modified so that a continuous fuzzy controller could be reconstructed by means of the Tensor Product Model Transformation.

Rule reduction

Chapter 4 presented an alternative rule reduction technique for TS models based on tensor approximation by using the HOSVD, as well as how to model the uncertainty generated by this rule reduction.

The proposed approach differs from the usual approaches in the literature since it deals with approximating a tensor composed by the "vertex systems" instead of dealing with approximating the nonlinear function directly. This approach advantages and disadvantages were discussed throughout the chapter.

The model uncertainty generated by the rule reduction proposed is given in a polytopic form. However, since the polytopic uncertainty representation is not very common in the TS fuzzy control literature, we presented LMI conditions that allow converting this uncertainty into either a norm bounded representation or a structured linear fractional form.

New stability analysis conditions

Chapter 5 proposed the use of local transformation matrices that, unlike the matrices generated by the membership functions ordering relations (Bernal, Guerra, and Kruszewski 2009), does not increase the number of LMI conditions.

New conditions were proposed to check the stability of continuous time TS systems. They made use of two piecewise fuzzy Lyapunov functions and the local transformation matrices.

The first of these conditions, based on the Lyapunov function proposed in (Rhee and Won 2006), lead to a set of conditions that do not depend on the knowledge of upper bounds on the time derivatives of the membership functions. However, as in (Rhee and Won 2006), they demand a particular structure on the matrices parametrizing the Lyapunov functions.

The second, based on the Lyapunov function proposed in (Tanaka, Hori, and Wang 2003), lead to a set of conditions that impose a less complex structure over the parameter matrices, but that demand the knowledge of upper bounds on the membership functions time derivatives and that, in some cases, only check for local stability.

Both conditions made use of the null-term approach (Mozelli, Palhares, and Mendes 2010) in order to relax the results. Theorem 5.1 can be seen as a generalization of (Bernal, Guerra, and Kruszewski 2009, Theorem 2). If global bounds on the time derivatives of the membership functions are known, then Theorem 5.2 can also be seen as a generalization of (Bernal, Guerra, and Kruszewski 2009, Theorem 2).

An example was presented to illustrate the relaxation introduced by the proposed conditions and the proposed local transformation matrices. As can be seen in the example, the proposed conditions where valid for a larger region than other conditions available in the literature. Its interesting to note that, *for this particular example*, the proposed local transformation matrices outperformed those using the matrices defined by the ordering relations from the membership functions in (Bernal, Guerra, and Kruszewski 2009).

A problem regarding the proposed candidate Lyapunov functions is that, in a direct way, the controller synthesis conditions will be BMIs. However, it might be possible to get LMI conditions by using a derivation similar to the one in (Qiu, Feng, and Gao 2013).

A parallel between chapters 3 and 5 can be traced, and in a similar way as the one discussed in chapter 3, we may say that the use of the local transformation matrices is equivalent to finding a switched representation, in the state space, for the original TS system.

Nonquadratic Lyapunov functions

Chapter 6 proposed new conditions for the local stability and local stabilization of continuous time TS fuzzy systems by using a nonquadratic Lyapunov function. These conditions avoid the use of upper bounds on the time derivative of the membership functions by making use of a novel manipulation that instead employs the membership functions Jacobian matrix and a polytopic description of the local region.

From the example presented in Section 6.3, we can see that the proposed stability conditions are an interesting alternative to other conditions available in the literature, with its main advantage being the fact that there is no need to choose a value for the upper bound of the membership functions' time derivative.

It is important to note that, even though the examples presented performed better than the ones from (Lee and Kim 2014) for the stability conditions, that was not always the case. It is also noteworthy that, whenever the stabilization conditions from (Lee and Kim 2014) were feasible, they yielded the largest domain of attraction out of all conditions we tested against (which in turn led us to present an example in which it does not work, in order to motivate the use of the proposed conditions).

By choosing a particular structure for M in (6.11), we were able to find a set of conditions for the stabilization problem which are guaranteed to find better results than those found by a quadratic Lyapunov function (given a suitable choice of μ).

LMI synthesis conditions

Chapter 7 proposed a set of new LMI synthesis conditions for continuous-time TS fuzzy systems by making use of piecewise-like Lyapunov functions. It adapted the bounding conditions on the membership functions time derivative from (Lee and Kim 2014) to work in this switched scenario and therefore made use of a chosen bound on the time derivative of the Lyapunov function's memberhsip functions in closed loop.

In order to diminish the effects of this time-derivative, we presented an optimal choice of the membership functions for these Lyapunov functions that guarantees the best worst-case derivative of the membership functions (and some other interesting properties that allows us to further relax the LMIs).

Finally, the same stabilization example from chapter 6 was used, and, in this case, the stabilization conditions proposed in chapter 7 were able to find a larger domain of attraction than those from chapter 6.

It is important to note that, for every example we tested in which the conditions from (Lee and Kim 2014) were also feasible, they were always able to find a larger domain of attraction (by correctly choosing the bounds on the membership functions' time derivative). Which motivates us to keep saerching for different ways to relax the results from this chapter.

8.2 Future Directions

Many different possible future directions are possible for the research presented in this thesis starting from the results presented so far.

A first, more theoretical, direction aims to try and understand under what conditions the feasibility of a parametric LMI is necessary for the feasibility of the quadratic form (a possible numerical direction could be to rewrite the quadratic form using a polytopic representation and trying to employ Polya's Theorem for homogenous polynomials to try and get asymptotically necessary conditions for the quadratic form).

Still on the topic of Polya's Theorem for homogenous polynomials, we would like to study if it could be used to generate a set of sufficient and asymptotically necessary conditions for solving general copositive programming problems like it has been used to solve fuzzy summation conditions.

Another interesting research direction would be continuing the research on the conditions of chapter 7. More specifically, we would like to determine a more systematic way of choosing the values of the upper bounds used for $\dot{g}_i$.

A different research direction, which would have direct impact over LMI synthesis conditions, would be to study under what conditions does the stability of the system

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{x}) A_i^T \mathbf{x}$$

is related to the stability of the system

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{x}) A_i \mathbf{x}.$$

Another research direction would be on the use of sufficient and asymptotically necessary conditions to define a SDP feasibility solver based on approximating LMIs by scalar linear inequalities constraints. We are specially interested in using something similar to (Kruszewski et al. 2009, Algorithm 2) as that would allow for the scalar linear inequalities approximation to be refined only where it mattered.

Bibliography

Ariño, C. V. (2007). "Relaxed LMI conditions for control of nonlinear Takagi-Sugeno models". PhD thesis. Universidad Politécnica de Valencia - Departamento de Ingeniería de Sistemas y Automática.

Ariño, C. and A. Sala (2007). "Relaxed LMI conditions for closed-loop fuzzy systems with tensor-product structure". In: *Engineering Applications of Artificial Intelligence* 20.8, pp. 1036 –1046. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2007.02.011.

Balas, G. J., A. K. Packard, and P. J. Seiler (2009). "Uncertain Model Set Calculation from Frequency Domain Data". In: *Model-Based Control*. Ed. by P. M. Hof, C. Scherer, and P. S. Heuberger. Springer US, pp. 89–105. ISBN: 978-1-4419-0895-7. DOI: 10.1007/978-1-4419-0895-7_6.

Baranyi, P. (1999). "Reduction of fuzzy rule base via singular value decomposition". In: *IEEE Transactions on Fuzzy Systems* 7.2, pp. 120–132. ISSN: 10636706. DOI: 10.1109/91.755394.

Baranyi, P. (2004). "TP model transformation as a way to LMI-based controller design". In: *IEEE Transactions on Industrial Electronics* 51.2, pp. 387–400. ISSN: 0278-0046.

Baranyi, P., D. Tikk, Y. Yam, and R. J. Patton (2003). "From differential equations to {PDC} controller design via numerical transformation". In: *Computers in Industry* 51.3, pp. 281 –297. DOI: http://dx.doi.org/10.1016/S0166-3615(03)00058-7.

Baranyi, P., Z. Petres, P. Varlaki, and P. Michelberger (2006). "Observer and control law design to the TORA system via TPDC framework." In: *WSEAS Transactions on Systems* 5.1, pp. 156–163.

Baranyi, P., Z. Petres, P. Korondi, Y. Yam, and H. Hashimoto (2007). "Complexity relaxation of the tensor product model transformation for higher dimensional problems". In: *Asian Journal of Control* 9.2, pp. 195–200.

Bernal, M. and T. M. Guerra (2010). "Generalized Nonquadratic Stability of Continuous-Time Takagi-Sugeno Models". In: *IEEE Transactions on Fuzzy Systems* 18.4, pp. 815 –822.

Bernal, M., T. M. Guerra, and A. Kruszewski (2009). "A membership-function-dependent approach for stability analysis and controller synthesis of Takagi–Sugeno models". In: *Fuzzy Sets and Systems* 160 (19), pp. 2776–2795.

Bernstein, D. S. (2009). *Matrix Mathematics: Theory, Facts, and Formulas*. Second. Princeton, New Jersey: Princeton University Press.

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press. ISBN: 978-0-521-83378-3.

Boyd, S., L. E. Ghaoui, E. Feron, and V. Balakrishnan (1994). *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics (SIAM). ISBN: 0-89871-334-X.

Bundfuss, S. (2009). "Copositive Matrices, Copositive Programming, and Applications". PhD thesis. Technischen Universität Darmstadt.

Bupp, R. T., D. S. Bernstein, and V. T. Coppola (1998). "A benchmark problem for nonlinear control design". In: *International Journal of Robust and Nonlinear Control* 8, pp. 307 –310.

Campos, V. C. S. (2011). "Projeto de Controladores Takagi-Sugeno baseados em Modelos de Referência". MA thesis. Universidade Federal de Minas Gerais.

Campos, V. C. S., L. A. B. Tôrres, and R. M. Palhares (2012). "Using information on membership function shapes in Asymptotically exact triangulation approaches". In: *51st IEEE Conference on Decision and Control*. IEEE, pp. 6205–6210.

Campos, V. C. S., L. A. B. Tôrres, and R. M. Palhares (2015). "Revisiting the TP Model Transformation: Interpolation and Rule Reduction". In: *Asian Journal of Control* 17.2, pp. 392–401. ISSN: 1934-6093. DOI: 10.1002/asjc.866.

Campos, V. C. S., F. O. Souza, L. A. B. Tôrres, and R. M. Palhares (2013). "New Stability Conditions Based on Piecewise Fuzzy Lyapunov Functions and Tensor Product Transformations". In: *IEEE Transactions on Fuzzy Systems* 21.4, pp. 748–760. DOI: 10.1109/TFUZZ.2012.2230178.

Chen, S.-H., W.-H. Ho, and J.-H. Chou (2009). "Robust Controlability of TS Fuzzy-Model-Based Control Systems With Parametric Uncertainties". In: *IEEE Transactions on Fuzzy Systems* 17.6, pp. 1324–1335.

Chen, Y.-J., H. Ohtake, W.-J. Wang, K. Tanaka, and H. O. Wang (2009). "Guaranteed cost control of T-S fuzzy systems using piecewise Lyapunov function based switching fuzzy controller". In: *Control Applications, (CCA) Intelligent Control, (ISIC), 2009 IEEE*, pp. 440 –445.

Choi, H. H. (2008). "Robust Stabilization of Uncertain Fuzzy Systems Using Variable Structure System Approach". In: *IEEE Transactions on Fuzzy Systems* 16.3, pp. 715–724.

Choi, H. H. (2010). “Robust Stabilization of Uncertain Fuzzy-Time-Delay Systems Using Sliding-Mode-Control Approach”. In: *IEEE Transactions on Fuzzy Systems* 18.5, pp. 979–984.

de Boor, C. (2001). *A Pratical Guide to Splines: Revised Edition*. New York: Springer.

De Lathauwer, L., B. De Moor, and J. Vandewalle (2000a). “A Multilinear Singular Value Decomposition”. In: *SIAM Journal on Matrix Analysis and Applications* 21.4, p. 1253. ISSN: 08954798. DOI: 10.1137/S0895479896305696.

De Lathauwer, L., B. De Moor, and J. Vandewalle (2000b). “On the Best Rank-1 and Rank-(R1 ,R2 ,. . .,RN) Approximation of Higher-Order Tensors”. In: *SIAM Journal on Matrix Analysis and Applications* 21.4, pp. 1324–1342. DOI: 10.1137/S0895479898346995.

Faria, F. A., G. N. Silva, and V. A. Oliveira (2012). “Reducing the conservatism of LMI-based stabilisation conditions for TS fuzzy systems using fuzzy Lyapunov functions”. In: *International Journal of Systems Science* 0.0, pp. 1–14. DOI: 10.1080/00207721.2012.670307.

Feng, G. (2011). *Analysis and Synthesis of Fuzzy Control Systems*. CRC Press. ISBN: 9781420092653.

Feng, G., C.-L. Chen, D. Sun, and Y. Zhu (2005). “H_∞ controller synthesis of fuzzy dynamic systems based on piecewise Lyapunov functions and bilinear matrix inequalities”. In: *IEEE Transactions on Fuzzy Systems* 13.1, pp. 94 –103.

Gonçalves, E. N., R. M. Palhares, R. H. C. Takahashi, and R. C. Mesquita (2006). “Algorithm 860: SimpleS - an Extension of Freudenthal’s Simplex Subdivision”. In: *ACM Trans. Math. Softw.* 32.4, pp. 609–621. ISSN: 0098-3500. DOI: 10.1145/1186785.1186792.

Guedes, J. A., M. C. M. Teixeira, R. Cardim, and E. Assunção (2013). “Stability of Nonlinear System Using Takagi-Sugeno Fuzzy Models and Hyper-Rectangle of LMIs”. In: *Journal of Control, Automation and Electrical Systems* 24.1-2, pp. 46–53. ISSN: 2195-3880. DOI: 10.1007/s40313-013-0015-4.

Guerra, T. M., M. Bernal, K. Guelton, and S. Labiod (2012). “Non-quadratic local stabilization for continuous-time Takagi-Sugeno models”. In: *Fuzzy Sets and Systems* 201.0, pp. 40 –54.

Hespanha, J. P., D. Liberzon, and A. R. Teel (2008). “Lyapunov conditions for input-to-state stability of impulsive systems”. In: *Automatica* 44.11, pp. 2735 –2744. DOI: 10.1016/j.automatica.2008.03.021.

Ho, W.-H., J.-T. Tsai, and J.-H. Chou (2007). “Robust-Stable and quadratic-optimal control for TS-fuzzy-model-based control systems with elemental parametric uncertainties”. In: *IET Control Theory and Appl.* 3.1, pp. 731–742.

Johansson, M., A. Rantzer, and K.-E. Arzen (1999). “Piecewise quadratic stability of fuzzy systems”. In: *IEEE Transactions on Fuzzy Systems* 7.6, pp. 713–722.

Ko, J. W. and P. G. Park (2012). "Further enhancement of stability and stabilisability margin for Takagi-Sugeno fuzzy systems". In: *IET Control Theory Applications* 6.2, pp. 313–318.

Ko, J. W., W. I. Lee, and P. Park (2012). "Stabilization for Takagi-Sugeno fuzzy systems based on partitioning the range of fuzzy weights". In: *Automatica* 48.5, pp. 970 –973.

Kruszewski, A., A. Sala, T. Guerra, and C. Ariño (2009). "A Triangulation Approach to Asymptotically Exact Conditions for Fuzzy Summations". In: *Fuzzy Systems, IEEE Transactions on* 17.5, pp. 985 –994.

Lam, H.-K. and F. H.-F. Leung (2011). *Stability Analysis of Fuzzy-Model-Based Control Systems*. Springer. ISBN: 9783642178443.

Lam, H. and M. Narimani (2010). "Quadratic-Stability Analysis of Fuzzy-Model-Based Control Systems Using Staircase Membership Functions". In: *IEEE Transactions on Fuzzy Systems* 18.1, pp. 125–137. DOI: 10.1109/TFUZZ.2009.2037744.

Lee, D. H., Y. H. Joo, and M. H. Tak (2014). "Local stability analysis of continuous-time Takagi-Sugeno fuzzy systems: A fuzzy Lyapunov function approach". In: *Information Sciences* 257.0, pp. 163 –175. DOI: 10.1016/j.ins.2013.08.036.

Lee, D. H. and D. W. Kim (2014). "Relaxed LMI Conditions for Local Stability and Local Stabilization of Continuous-Time Takagi-Sugeno Fuzzy Systems". In: *IEEE Transactions on Cybernetics* 44.3, pp. 394–405. DOI: 10.1109/TCYB.2013.2256781.

Lee, D. H., J. B. Park, and Y. H. Joo (2012). "A fuzzy Lyapunov function approach to estimating the domain of attraction for continuous-time Takagi-Sugeno fuzzy systems". In: *Information Sciences* 185.1, pp. 230 –248.

Liberzon, D. (2003). *Switching in systems and control*. Birkhäuser.

Lin, C., Y. He, Q.-G. Wang, and T. H. Lee (2007). *LMI Approach to Analysis and Control of Takagi-Sugeno Fuzzy Systems with Time Delay*. Springer. ISBN: 3540495525.

Lo, J.-C. and M.-Z. Liu (2010). "Exact solutions to fuzzy PD-LMIs via SOS". In: *Proc. 2010 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010)*, pp. 1 –6.

Löfberg, J. (2004). "YALMIP : A Toolbox for Modeling and Optimization in MATLAB". In: *Proceedings of the CACSD Conference*. Taipei, Taiwan.

MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations". In: *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281 –297.

Mansouri, B., N. Manamanni, K. Guelton, A. Kruszewski, and T. Guerra (2009). "Output feedback LMI tracking control conditions with H_∞ criterion for uncertain and disturbed T-S models". In: *Information Sciences* 179.4, pp. 446–457. ISSN: 00200255.

Mean Value Theorem (2012). URL: http://en.wikipedia.org/wiki/Mean_value_theorem.

Montagner, V., R. Oliveira, and P. Peres (2009). "Convergent LMI Relaxations for Quadratic Stabilizability and $\mathcal{H}_\infty$ Control of Takagi-Sugeno Fuzzy Systems". In: *Fuzzy Systems, IEEE Transactions on* 17.4, pp. 863 –873.

Mozelli, L. A., R. M. Palhares, and G. S. C. Avellar (2009). "A systematic approach to improve multiple Lyapunov function stability and stabilization conditions for fuzzy systems". In: *Information Sciences* 179.8, pp. 1149–1162.

Mozelli, L. A., R. M. Palhares, and E. M. A. M. Mendes (2010). "Equivalent techniques, extra comparisons and less conservative control design for Takagi-Sugeno (TS) fuzzy systems". In: *IET Control Theory and Applications* 4.12, pp. 2813–2822.

Mozelli, L. A., R. M. Palhares, F. O. Souza, and E. M. A. M. Mendes (2009). "Reducing conservativeness in recent stability conditions of TS fuzzy systems". In: *Automatica* 45.6, pp. 1580–1583.

Nagy, S., Z. Petres, P. Baranyi, and H. Hashimoto (2009). "Computational relaxed TP model transformation: restricting the computation to subspaces of the dynamic model". In: *Asian Journal of Control* 11.5, pp. 461–475.

Pan, J.-T., T.-M. Guerra, S. min Fei, and A. Jaadari (2012). "Nonquadratic Stabilization of Continuous T-S Fuzzy Models: LMI Solution for a Local Approach". In: *IEEE Transactions on Fuzzy Systems* 20.3, pp. 594–602. DOI: 10.1109/TFUZZ.2011.2179660.

Petres, Z., P. Baranyi, and H. Hashimoto (2010). "Approximation and complexity trade-off by TP model transformation in controller design: A case study of the TORA system". In: *Asian Journal of Control* 12.5, pp. 575–585. ISSN: 1934-6093. DOI: 10.1002/asjc.222.

Petres, Z., P. Baranyi, F. Kolonic, and A. Poljugan (2005). "Approximation Trade-off by TP Model Transformation". In: *6th International Symposium of Hungarian Researchers on Computational Intelligence*.

Petres, Z., P. Baranyi, P. Korondi, and H. Hashimoto (2007). "Trajectory Tracking by TP Model Transformation: Case Study of a Benchmark Problem". In: *IEEE Transactions on Industrial Electronics* 54.3, pp. 1654–1663. ISSN: 0278-0046. DOI: 10.1109/TIE.2007.894697.

Qhull (2011). URL: http://www.qhull.org/.

Qiu, J., G. Feng, and H. Gao (2013). "Static-Output-Feedback $\mathcal{H}_\infty$ Control of Continuous-Time T-S Fuzzy Affine Systems Via Piecewise Lyapunov Functions". In: *IEEE Transactions on Fuzzy Systems* 21.2, pp. 245–261. DOI: 10.1109/TFUZZ.2012.2210555.

Rhee, B. and S. Won (2006). "A new fuzzy Lyapunov function approach for a Takagi-Sugeno fuzzy control system design". In: *Fuzzy Sets and Systems* 157.9, pp. 1211–1228.

Sala, A. (2007). "Is bridging the gap between fuzzy and nonlinear control possible (partially)?" In: *Proc. European Control Conference*. Kos, Greece.

Sala, A. (2009). “On the conservativeness of fuzzy and fuzzy-polynomial control of nonlinear systems”. In: *Annual Reviews in Control* 33.1, pp. 48 –58.

Sala, A. and C. Ariño (2007). “Asymptotically necessary and sufficient conditions for stability and performance in fuzzy control: Applications os Polya’s theorem”. In: *Fuzzy Sets and Systems* 158 (24), pp. 2671 –2686.

Sala, A. and C. Ariño (2008). “Relaxed Stability and Performance LMI Conditions for Takagi–Sugeno Fuzzy Systems With Polynomial Constraints on Membership Function Shapes”. In: *Fuzzy Systems, IEEE Transactions on* 16.5, pp. 1328 –1336.

Sofianos, N. A. and O. I. Kosmidou (2009). “Robust Fuzzy Control of Nonlinear Discrete-Time Systems with Polytopic Uncertainties”. In: *2009 IEEE Mediterranean Conference on Control and Automation*. IEEE, pp. 25–30.

Stein, P. (1966). “A Note on the Volume of a Simplex”. In: *The American Mathematical Monthly* 73.3, pp. 299–301.

Takagi, T. and M. Sugeno (1985). “Fuzzy identification of systems and its application to modeling and control”. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-15.1, pp. 116–132.

Tanaka, K., T. Hori, and H. Wang (2003). “A multiple Lyapunov function approach to stabilization of fuzzy control systems”. In: *IEEE Transactions on Fuzzy Systems* 11.4, pp. 582 –589.

Tanaka, K. and H. Wang (2001). *Fuzzy control systems design and analysis: a linear matrix inequality approach*. Wiley-Interscience. ISBN: 0471323241.

Taniguchi, T., K. Tanaka, H. Ohtake, and H. Wang (2001). “Model construction, rule reduction, and robust compensation for generalized form of Takagi-Sugeno fuzzy systems”. In: *IEEE Transactions on Fuzzy Systems* 9.4, pp. 525–538. DOI: 10.1109/91.940966.

Teixeira, M. C. M. and E. Assunção (2007). “Enciclopédia de Automática - Controle e Automação”. In: ed. by L. A. Aguirre. Vol. 1. Editora Blucher. Chap. 9 - Extensões para sistemas não-lineares, pp. 218–231.

Tognetti, E. S., R. C. L. F. Oliveira, and P. L. D. Peres (2011). “Selective H_2 and H_∞ Stabilization of Takagi-Sugeno Fuzzy Systems”. In: *IEEE Transactions on Fuzzy Systems* 19.5, pp. 890–900.

Tognetti, E. S., R. C. Oliveira, and P. L. Peres (2012). “Reduced-order dynamic output feedback control of continuous-time T-S fuzzy systems”. In: *Fuzzy Sets and Systems* 207.0, pp. 27 –44. DOI: http://dx.doi.org/10.1016/j.fss.2012.04.013.

Tuan, H., P. Apkarian, T. Narikiyo, and Y. Yamamoto (2001). “Parameterized linear matrix inequality techniques in fuzzy control system design”. In: *IEEE Transactions on Fuzzy Systems* 9.2, pp. 324–332. ISSN: 1063-6706. DOI: 10.1109/91.919253.

Varkonyi, P., D. Tikk, P. Korondi, and P. Baranyi (2005). “A new algorithm for RNO-INO type tensor product model representation”. In: *2005 IEEE International Confer-*

ence on Intelligent Engineering Systems, INES '05. IEEE, pp. 263–266. ISBN: 0-7803-9474-7. DOI: 10.1109/INES.2005.1555170.

Wang, H. and G.-H. Yang (2013). "Controller design for affine fuzzy systems via characterization of dilated linear matrix inequalities". In: *Fuzzy Sets and Systems* 217.0, pp. 96 –109. ISSN: 0165-0114. DOI: 10.1016/j.fss.2012.10.006.

Xiaodong, L. and Z. Qingling (2003). "New approaches to H_∞ controller designs based on fuzzy observers for T-S fuzzy systems via LMI". In: *Automatica* 39.9, pp. 1571–1582. ISSN: 0005-1098. DOI: 10.1016/S0005-1098(03)00172-9.

Zhang, B., S. Zhou, and T. Li (2007). "A new approach to robust and non-fragile H_∞ control for uncertain fuzzy systems". In: *Information Sciences* 177, pp. 5118–5133.

Zhang, H., C. Li, and X. Liao (2005). "Stability analysis and H_∞; controller design of fuzzy large-scale systems based on piecewise Lyapunov functions". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 36.3, pp. 685 –698.

Zhang, H. and X. Xie (2011). "Relaxed Stability Conditions for Continuous-Time T-S Fuzzy-Control Systems Via Augmented Multi-Indexed Matrix Approach". In: *IEEE Transactions on Fuzzy Systems* 19.3, pp. 478 –492.

Zhou, S., G. Feng, J. Lam, and S. Xu (2005). "Robust H_∞ control for discrete-time fuzzy systems via basis-dependent Lyapunov functions". In: *Information Sciences* 174.3–4, pp. 197–217. ISSN: 0020-0255. DOI: 10.1016/j.ins.200407.015.