Óbuda University

Generalization of Tensor Product Model Based Control Analysis and Synthesis

Ph.D. Theses

József Kuti

Supervisor:

Dr. Péter Galambos

Budapest
2018

Szigorlati bizottság:

Prof. Dr. Tar József Kázmér: Rendszer és irányításelmélet

Prof. Dr. Szalay Tibor: Robotika

Prof. Dr. Galántai Aurél: Mátrixanalízis

Házivédés oppenesek:

Prof. Dr. Huijun Gao (Harbin Institute of Technology)

Dr. Drexler Dániel (Óbudai Egyetem)

Nyilvános védés opponensek:

Dr. Takarics Béla (MTA SZTAKI)

Dr. Kovács Levente (Óbudai Egyetem)

Acknowledgments

I am grateful to my supervisor, Dr. Péter Galambos, who introduced me to the world of research during our collaboration since 2012. This thesis was motivated by his practical, straightforward attitude and by the inspiring atmosphere of scientific research and development he established.

I also wish to thank Prof. Péter Baranyi, who gave me the opportunity to follow my thoughts during the three years I spent in MTA SZTAKI.

I would like to thank my former colleagues, Prof. Zoltan Szabó, Prof. Gyula Sallai, Dr. Ádám Csapó and Dr. Béla Takarics for all the help I received from them.

I would also like to express my appreciation to Doctoral School of Applied Informatics and Applied Mathematics of Óbuda University and Research and Innovation Center of Óbuda University and teachers of Budapest University of Technology and Economics, who highly motivated me during that years: Prof. Gábor Stépán who showed the nature of dynamical systems and gave analytical tools to investigate it, Prof. Antal Huba who presented the role of engineers to influence their behaviour, and finally Prof. József Bokor and Prof. Béla Lantos who allowed me to learn about the essence of control theory.

Last, but not least, special thanks to my parents and my family, who have supported me during the whole period of my studies.

The research was supported by the Hungarian National Development Agency, (ERC-HU-09-1-2009-0004 MTASZTAK) (OMFB-01677/2009), by the Hungarian State and the European Union under the EFOP-3.6.1-16-2016-00010 project and by the UNKP-16-3 New National Excellence Program of the Ministry of Human Capacities.

Declaration

Undersigned, József Kuti, hereby state that this Ph.D. Thesis is my own work, wherein I only used the sources listed in the bibliography. All parts taken from other works, either as word for word citation or rewritten keeping the original meaning, have been unambiguously marked, and reference to the source was included.

Nyilatkozat

Alulírott Kuti József kijelentem, hogy ezt a doktori értekezést önállóan készítettem, és abban csak a listában szereplő forrásokat használtam fel. Minden olyan részt, amelyet szó szerint szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen a forrás megadásával megjelöltem.

Budapest, April 20, 2018

József Kuti

Contents

Preface **1**

Goals of the thesis **2**

Structure of the dissertation **3**

Nomenclature **4**

I Introduction **6**

1 Preliminaries and scientific background **7**

2 LMI based controller design for polytopic LPV/qLPV modells . . **10**

3 Tensor-Product Model Transformation based control **17**

II Theoretical Achievements **28**

4 Geometric interpretation of polytopic form derivation **31**

4.1 Basic concepts 31

4.2 Affine geometry of polytopic form derivation problem 32

4.2.1 Affine Singular Value Decomposition 33

4.2.2 Numerical reconstruction of Affine SVD 35

4.3 Polytopic form by determining enclosing polytope on the affine hull . 37

4.3.1 Simplex enclosing polytopes 38

4.3.2 Non-simplex enclosing polytopes 39

4.4 Summary 39

4.5 Proofs 40

5 New Polytopic TP form definition and Affine TP Model Transformation . . . 42

5.1 Tensor algebra extension . . . 43

5.2 Definition of relaxed Polytopic TP form . . . 45

5.2.1 Affine operators on Polytopic TP forms . . . 46

5.3 Derivation of Polytopic TP forms for multivariate functions . . . 48

5.3.1 Definition and properties of Affine TP form . . . 48

5.3.2 Derivation of Affine TP form . . . 49

5.3.3 Complexity reduction . . . 51

5.3.4 Complexity reduction preserving exactness . . . 52

5.3.5 Remarks on numerical reconstruction . . . 53

5.4 Summary . . . 59

5.5 Proofs . . . 60

6 Polytopic Tensor-Product Models for control purposes . . . 67

6.1 Problem formulation . . . 68

6.2 Polytopic TP Model Generation and Manipulation . . . 69

6.3 Minimal Volume Simplex (MVS) Generation and Manipulation Methodology . . . 71

6.3.1 Minimal Volume Enclosing Simplex Generation . . . 72

6.3.2 Minimal Volume Enclosing Simplex Manipulation . . . 74

6.4 Minimal Volume Non-Simplex Enclosing Polytope . . . 76

6.4.1 Duality of convex hull problem and the intersection of halfspaces 77

6.4.2 Local volume minimization on the normals . . . 78

6.4.3 Cut off regions by additional halfspaces . . . 79

6.5 Summary . . . 81

6.6 Proofs . . . 81

7 Generalization of Polytopic TP Model-based Controller Design . 84

7.1 Polytopic TP forms in control analysis and synthesis criteria . . . 84

7.2 Definite conditions in Polytopic TP form . . . 85

7.3 Application in state feedback controller design . . . 87

7.4 Summary . . . 89

III Practical merit 90

8 Inverted pendulum 91

8.1 Mechanical model and qLPV modeling 91

8.2 Control goals 92

8.3 Affine Tensor Product Model Transformation 93

8.4 MVS Polytopic TP Model-based controller design 97

8.5 Application of Polytopic TP Model Manipulation 103

8.6 Conclusion 104

9 Other published applications 106

9.1 Output feedback control of a dual-excenter vibration actuator 106

9.2 Translational Oscillator with a Rotational Actuator (TORA) system 109

IV Conclusion 112

10 Summary of the scientific results 113

Appendix 118

A Former methods of TP Model Transformation to generate simplex enclosing polytopes 119

A.1 SNNN enclosing simplex algorithm 119

A.2 IRNO enclosing simplex algorithm 120

A.3 CNO enclosing simplex algorithm 122

A.4 Proofs 124

B Extraction of multiple polytopic summation 126

B.1 Methods to extract double polytopic summation 126

B.2 Pólya-theorem based method to extract multiple polytopic summations 129

B.3 Proofs 130

List of Figures

3.1 Illustrations for tensor operations 19

3.2 TP model transformation (N=2) 24

3.3 The new control design workflow and the role of the chapters 30

4.1 Illustration of the used notations: the considered function $\mathfrak{c} : X \to H$, the affine hull $\mathfrak{A}$ of the image set, that is here $D = 2$ dimensional, orthogonal basis $(\mathfrak{a}_1, \mathfrak{a}_2)$, offset to it $\mathfrak{a}_3$, the convex hull of the image set $\mathfrak{C}$, enclosing polytope with vertices $(\mathfrak{s}_1, \mathfrak{s}_2, \mathfrak{s}_3)$ 32

4.2 Illustration of unique weighting functions of simplex polytopic forms . 39

5.1 Illustrations of extended tensor operations 44

5.2 Illustration of discretisation using piecewise linear interpolatory functions 51

5.3 Illustration of partitions if $K = 3$, $M_1 = 13$, $M_2 = 9$, $M_3 = 6$, $F = 3$, $F' = 1$, $G = 5$ 54

5.4 Illustration of Algorithm 5.20 considering the case depicted in Fig. 5.3 56

6.1 Enclosing polytope as intersection of halfspaces (given by $\mathbf{n}_i$ normals and α_i offsets) 76

8.1 Inverted pendulum model 91

8.2 The weighting functions of the Affine TP model 96

8.3 The weighting functions of the Affine TP model via discretisation . . 96

8.4 The weighting functions of the MVS polytopic TP model 98

8.5 Minimal volume enclosing simplex for $\mathbf{v}^{(1)}(p_1)$ function 99

8.6 The results of controller design (A) 100

8.7 The results of controller design (B) 101

8.8 The results of controller design (C) 102

8.9 The achievable γ_∞ disturbance rejection with different variable complexities 103

8.10 Minimal volume enclosing simplex for $\mathbf{v}^{(1)}(p_1)$ function 104

8.11 Minimal volume enclosing simplex for $\mathbf{v}^{(1)}(p_1)$ function 105

9.1 Mechanical model of the vibrating system 107

9.2 Simulation results: Three short vibration impulses show that the frequency and amplitude are adjustable independently. 108

9.3 The mechanical model of the TORA system 109

9.4 The results of optimisation LMI on manipulated models, the dash line shows the minimal feasible ν value for all $\mathbf{S}(\mathbf{p})$ system matrices for comparison 110

9.5 The MVS enclosing polytope and the cut one 111

9.6 Behaviour of the controlled system 111

10.1 The new control design workflow and role of the elaborated theses . . 113

10.2 Illustration of discretisation via multivariate interpolatory functions . 115

A.1 Illustration of SNNN enclosing polytope generation 120

A.2 Illustration of A, B, C, D steps for a given α 122

A.3 Phases of CNO enclosing polytope generation for a $D = 2$ problem and given $\varphi_1, \varphi_2, \varphi_3$ values 123

Preface

During my master's diploma research and period of the Young Research Award, I have applied the TP model transformation and basic LMI-based control design methods to different non-linear and time-delay feedback problems:

1. A haptic telemanipulator arm with time-delayed communication and unknown remote environment, where the effect of time-delay must be decreased because it influences the sensed properties of the remote environment and may cause unstable behavior.

2. The prototype of the so-called dual-excenter actuator. That is a special vibration actuator that is able to work at independently chosen frequency and amplitude in contrast to the widely used vibration devices. Its control is challenging because of its highly non-linear behavior and the unknown working environment.

The development and the corresponding theoretical and numerical supervisions dredged up more practical demand and sources of conservativeness. These experiences motivated my Ph.D. research to extend and renew the theoretical basis of TP model transformation based controller design.

The dissertation

– introduces new concepts and numerical methods to deal with practical challenges of modeling and control design,

– deeply renews the abstract, mathematical formalism of Tensor-Product Model-based control analysis, synthesis and the corresponding transformation method.

For the sake of generality, the elaborated definitions are characterized in an abstract mathematical way, and the corresponding proofs are provided as well. Similarly, the proposed numerical methods were described for general cases - that cover usually higher dimensional geometry, tensors with arbitrary number of dimensions, models that depend on arbitrary number of parameters, that are given in arbitrary number of parameter sets – that is necessary to see the nature of problems and the proposed methods to handle them.

Goals of the thesis

Tensor product models are multi-polytopic forms of Linear Parameter Varying (LPV) and quasi-LPV (qLPV) models separating their parameter dependencies. The motivation of their application is the existence of "automatic" numerical methods to obtain this kind of polytopic models and the numerous control analysis and synthesis methods that can be immediately applied to them.

The theses of this dissertation extend the practical opportunities and revise the methodology by taking into account the current problems:

1. In many cases, the infeasibility of controller synthesis was apparently caused by the actual structure of polytopic TP model, but there was no method that can efficiently, fast and repeatably generates and fine manipulate the polytopic structures. Furthermore, the existing methods are only capable of providing simplex polytopes.

2. Although the Higher Order Singular Value Decomposition (HOSVD) based Tensor Product form gives a good description about algebraic structure and complexity of the model, it is not really connected to the problem of generating a polytopic form. The nature of these problems is geometric, and the intermediate state is expected to represent the affine geometric structure of the model, which, however, is not provided by the former HOSVD based approach.

3. The separation of the parameter dependencies into a multi-polytopic structure may increase the complexity but was not exploited during controller design. There is a practical need either for lossless polytopic models for some cases and for a controller design approach that can exploit the parameter separations by handling them in different ways (according to their practical properties). For example, to apply controller candidates that depend only on the measurable parameters, and Lyapunov-candidates that depend only on the constant parameters, etc.

Structure of the dissertation

The dissertation is structured as follows:

The first part of the dissertation describes the related basic concepts.

- Chapter 1 gives a brief review of the history of control theory, clarifying the motivation of robust and optimal control design, and highlighting the relevance of Lyapunov's works.
- Chapter 2 details the LPV/qLPV modeling, its relationship to the LMI based controller design and the importance of convexity.
- Following that, Chapter 3 presents the concept of TP Model Transformation as the main objective of the theoretical research.

The second part of the dissertation presents the theoretical achievements.

- Chapter 4 shows the role of affine descriptions to derive polytopic forms and proposes the unique ASVD for this purpose.
- Chapter 5 generalizes the result to derive polytopic TP forms and proposes further extensions of the definitions.
- After that, Chapter 6 considers the geometric problem to generate and manipulate enclosing polytopes according to the control goals.
- Finally, Chapter 7 describes the extended concept of Polytopic TP model-based control analysis and synthesis, which generalizes the polytopic TP model-based controller design.

The third part shows the application of the proposed concepts and methods on practical problems.

- Chapter 8 provides a detailed numerical example by considering the simple 2D inverted pendulum problem.
- Chapter 9 summarizes the achieved results on the dual-excenter vibrotactor and the translational oscillator with rotational actuator (TORA) systems.

The fourth part concludes the scientific results in five theses.

Nomenclature

The following abbreviations and notations are used along the dissertation:

LTI	Linear Time Invariant
(q)LPV	(quasi)Linear Parameter Varying
LQR, LQG	Linear Quadratic Regulator, Linear Quadratic Gaussian
LMI	Linear Matrix Inequality
SDP	Semidefinite Programming
SQP	Sequential Quadratic Programming
SOS	Sum of Squares
SVD	Singular Value Decomposition
ASVD	Affine Singular Value Decomposition
HOSVD	Higher Order Singular Value Decomposition
HOOI	Higher Order Orthogonal Iteration
TP model	Tensor Product model
MVS, MVSA	Minimal Volume Simplex, Minimal Volume Simplex Analysis
PDC	Parallel Distributed Compensation

For scalars:

$a, b, \dots$	scalar values
δ_{ij}	Dirac-delta ($\delta_{ii} = 1$, $\delta_{ij} = 0$ if $i \neq j$)
$\underline{x}, \overline{x}$	lower and upper bounds for the x scalar

For vectors and matrices:

$\mathbf{a}, \mathbf{b}, \dots$	vectors
$\mathbf{A}, \mathbf{B}, \dots$	matrices
$\mathbf{0}^{a\times b}$,$\mathbf{1}^{a\times b}$	$a \times b$ size matrix of zeros/ones
$\mathbf{E}^{a\times b}$	$a \times b$ size identity matrix
$\mathbf{M}^T$	transposed matrix
$\mathrm{Sym}(\mathbf{M})$	sum of the matrix and its transposed
$\mathrm{Tr}(\mathbf{M})$	trace of the matrix
$\mathbf{M}^\dagger$	Moore-Penrose pseudoinverse of the matrix
$\succ$, $\succeq$	definite, semidefinite conditions of matrices
$[a_i]_{i=1,..,I}$	row vector as $\begin{bmatrix} a_1 & \dots & a_I \end{bmatrix}$

$[a_{ij}]_{i=1,..,I,j=1..J}$	matrix as $\begin{bmatrix} a_{11} & \dots & a_{1J} \\ \vdots & & \vdots \\ a_{I1} & \dots & a_{IJ} \end{bmatrix}$
$\mathbf{A} \otimes \mathbf{B}$	Kronecker-product of matrices

For tensors:

$\mathcal{A}, \mathcal{B}, \dots$	tensors
$\mathcal{A}_{j_k=j}$	the j-th k-mode subtensor of tensor $\mathcal{A}$
$\mathbf{A}_{(n)}$	n-mode unfold matrix of tensor $\mathcal{A}$
$\mathcal{A} \times_n \mathbf{U}$	n-mode tensor product
$\mathcal{A} \mathop{\boxtimes}\limits_{n=1}^{N} \mathbf{U}^{(n)}$	multiple tensor product as $\mathcal{A} \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)}$

For TP forms and TP models:

N	number of parameters
$\mathbf{p}$	vector of parameters
Ω	hyper-rectangle parameter domain as $\Omega = [\underline{p}_1, \overline{p}_1] \times \cdots \times [\underline{p}_N, \overline{p}_N]$
K	number of parameter sets
$\mathbf{p}^{(k)}$	the k-th parameter set $1 \leq k \leq K$
$\mathbf{v}(\mathbf{p})$	a weighting function of an Affine TP model (orthonormal and homogeneous: $\mathbf{v}(\mathbf{p}) = \begin{bmatrix}\mathbf{u}(\mathbf{p}) & 1\end{bmatrix}$)
$\mathbf{w}(\mathbf{p})$	a weighting function of a Polytopic TP model (denoting convex combinations for all $\mathbf{p} \in \Omega$)

For Hilbert spaces:

H	a Hilbert space, in general
$\mathfrak{a}, \mathfrak{b}, ...$	elements of a H Hilbert space, in general
$< \cdot, \cdot >$	inner product

For sets:

$\mathbb{R}$, $\mathbb{N}$	spaces of real numbers and non-negative integers
$\mathbb{R}^a$, $\mathbb{N}^a$	spaces of vectors on $\mathbb{R}$ and $\mathbb{N}$
$\mathfrak{A}, \mathfrak{B}, \dots$	sets on $\mathbb{R}^a, H, \dots$
Co(..)	Convex hull (set of all convex combinations)
$\{a_i\}_{i=1,..,I}$	set as $\{a_1, \dots, a_I\}$
V($\cdot$)	volume of a metric set
$\subseteq, \subset$	subset, strict subset

Furthermore:

$\mathbf{A}^{(n)}, \mathcal{B}^{(n)}$	indexing of different matrices, tensors
$\lVert \cdot \rVert_2$, $\lVert \cdot \rVert_\infty$	H_2, H_∞ norm for systems, L_2, L_∞ norm for signals

Part I

Introduction

Chapter 1

Preliminaries and scientific background

The robust and parameter varying controller design, which the dissertation is based on, is one of the most important advancements of control theory in the last decades. This chapter presents the key moments from their birth to the methods related to the TP model transformation.

Classical control of Linear time invariant (LTI) systems

Since Maxwell's work [122] it is known that the roots of the characteristic polynomial determine the stability of the system. The Routh-Hurwitz stability criteria [26, 144] allowed to check the stability for fourth- or higher-order systems.

Following that, the Nyquist's analysis method shows the phase margin as well [134]. It was extended by Bode in [28] by elaborating the relationship of gains and phase-shift and introducing the concepts of gain and phase margins. In this way, the mathematical background of feedback controllers was established, primarily for three-term PID control.

Later, Kalman's results on controllability and duality of controller, observer and filter design were crucial milestones, [79]. The poles of controllable systems can be arbitrarily placed via Ackermann's formula [1].

Linear systems show the general mathematical nature of controller design: state feedback and full order dynamic output feedback design are fundamentally convex optimization problems, but static and non-full order dynamic output feedback designs are not convex, and thus, cannot be designed systematically [109].

Optimal control

There emerged the need for control strategies that result in the "best" running. Based on the cost function defined in time domain, the best controller can be obtained by Bellman's dynamic programming [22] and Pontryagin's maximum principle that applies classical variational formulations of analytical mechanics given by Lagrange and Hamilton into optimal control.

For considering linear systems, Kalman introduced the integral quadratic cost allowing the synthesis for MIMO systems and proved that the optimal control can be achieved through linear state feedback [79]. This linear-quadratic (LQ), H_2 optimal controller can be obtained by solving the corresponding algebraic Ricatti-equations, and the Linear Quadratic Gaussian optimal filter can similarly be obtained as well [79].

Robust control

The practical recognition that optimal controllers work only under ideal circumstances [8] motivated the birth of robust control paradigm, where the primary purpose of controller synthesis is to reject the effects of disturbances, noises, unknown loads, etc. by taking into account the uncertainties of model parameters.

The H_∞ control design was born as a worst-case approach, and to decrease its conservativeness, the μ synthesis was formulated [56, 188]. By applying Singular Value Decomposition, it can be applied to MIMO systems as well [55, 145], which resulted in the widely used Robust Control Toolbox for Matlab [11, 146].

Stability based on Lyapunov's direct method

The central paradigm of the state-space model-based control is Lyapunov's second, so-called direct method introduced in his Ph.D. thesis [120, 137] for stability verification of autonomous and controlled systems via so-called positive semidefinite Lyapunov functions. The criteria for linear systems lead to definite conditions on matrices.

Lyapunov only considered simple mathematical models. Decades later, Lur'e et al. [119] applied the theory to nonlinear practical problems by solving the resulting inequalities analytically. Following that, Chetaev dealt with astronautical stability problems arising in the spin stabilization of rockets, constructing the Lyapunov function based on the mechanical energy terms [43].

Semi-Definite Programming (SDP)

The definite, semi-definite conditions had hardly been solved [184] until the convex nature of the Linear Matrix Inequalities (LMIs) was discovered [139, 140]. After the ellipsoid method, the interior point methods were the first, practically relevant polynomial solvers [80, 131, 133]. These methods were improved in the last decades, see [4, 158, 179] for more details.

Meanwhile, the research of their general application in control analysis and synthesis have been started by Yakubovich [186, 187]. The Ricatti-equations of LQ, H_2, H_∞ criteria can be easily written as LMIs, and furthermore, pole placement constraints can be formulated as LMI regions [45, 46, 71, 76, 111, 123]. The merit of the LMI based approach is the opportunity to achieve multiobjective controller design: The criteria (e.g., LMI region constraints, disturbance rejection H_∞ constraints and H_2 optimal constraint) can be applied together [60].

There are two directions of taking into account the uncertainties of the linear model: the so-called norm bounded description and polytopic envelopes [32, 66]. It is important to denote that although

- the robust/optimal state feedback,
- the full-order dynamic output feedback methods without model uncertainties

lead to LMI conditions so a convex optimization problem, whereas

- the – practically absolutely relevant – robust output feedback design
- not full-order output feedback design

are not convex problems, but they can be convexified with additional conservativeness.

Linear Parameter Varying (LPV) modelling

Linear parameter varying models are a special class of systems that can be modeled as a parametrized linear system. Their control includes robust and adaptive methods, where the controller depends also on the parameters that are measured, observed or estimated online [10, 30, 41, 121, 126].

Nonlinear systems can also be considered as "quasi-LPV (qLPV)" systems [115, 116, 117], which is the clear extension of local linearization of the state space, but the conservativeness and non-uniqueness of the resulting qLPV model must be considered.

There are more approaches beside polytopic model based approach as affine LPV description based [106, 62], norm-bounded descriptions and Linear Fractional Transformation-based descriptions [153, 190], interval analysis [78, 83].

Polytopic model based control

The polytopic model-based control was first systematically mentioned by Boyd in [32], who showed that the uncertainties or parameter dependencies described by possible convex combinations of so-called vertex systems allow performing certain control analysis and synthesis methods via convex optimization on Linear Matrix Inequalities. The polytopic model-based control design is syntactically equivalent to control design for the well known Takagi-Sugeno systems [159]. In the past two decades, plenty of methods were published, which consider a wide range of control goals and possible relaxations of conservativeness [66, 112].

TP model transformation

Later in [13], Baranyi proposed a polytopic form, the Tensor-Product model, for parametric uncertain systems, where the scalar parameter dependencies are represented separately, and all the parameters have their polytopic structure. For more details, see [15] and [18] for surveys of the latest applications.

Chapter 2

LMI based controller design for polytopic LPV/qLPV modells

This chapter discusses the basics of LMI based controller design for polytopic models. First, it briefly summarizes the basic concepts of optimization on Linear Matrix Inequalities, Linear Parameter Varying (LPV) modeling and its extension, the quasi-Linear Parameter Varying (qLPV) modeling for nonlinear systems. A few definite condition-based control criteria for LPV/qLPV models will be shown, and it will be pointed out, how they can be transformed to LMIs by applying polytopic modeling.

Linear Matrix Inequality (LMI) optimisation

Linear Matrix Inequalities are definite or semi-definite conditions of matrices with affine variable dependencies as

$$\mathbf{M}(\mathbf{x}) \equiv \mathbf{M}_{(0)} + \sum_{k=1}^{K} x_k \mathbf{M}_{(k)} \succ 0, \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^K$ contains the variables, the $\mathbf{M}_{(k)}$ $k = 0, ..., K$ matrices are given and symmetric, and the symbol $\succ$ denotes positive definiteness, so the eigenvalues of $\mathbf{M}(\mathbf{x})$ must be positive. The positive or negative semi-definiteness are denoted by "$\succeq$", "$\preceq$", which also allow zero eigenvalues and are called non-strict LMIs.

The notations, like $\mathbf{A} \succ \mathbf{B}$, are often used for conditions $\mathbf{A} - \mathbf{B} \succ 0$.

An LMI condition (or a set of conditions) is called feasible if there exists $\mathbf{x}$ that satisfies the conditions. Then the set of solutions is called feasible set. The feasible set of an LMI condition (or set of LMI conditions) is convex.

The minimisation of a convex scalar function $g(\mathbf{x})$, subject to LMI conditions as

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\, g(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{M}(\mathbf{x}) \succ 0, \quad \mathbf{N}(\mathbf{x}) \succeq 0 \end{aligned} \tag{2.2}$$

is called a Semi-Definite Program.

It is a convex optimization problem in general because the constraints restrict $\mathbf{x}$ to a convex set and a convex function is optimized on it.

It is a special type of Cone Programming (beside the Linear Programming, Quadratic Programming, and a few other types). As such the interior point methods can be applied, and their computational time depends on the complexity polynomially [4, 5, 6, 81, 132, 158, 179].

Application in Stability Analysis of Linear Systems. To illustrate the role of the upper concepts in control theory, consider an autonomous linear system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$$

From Lyapunov's direct method [120] it is stable if there exists a quadratic Lyapunov function $V(t) = \mathbf{x}^T(t)\mathbf{P}\mathbf{x}(t)$ that is positive definite, and its derivative

$$\dot{V}(t) = \mathbf{x}^T(t)\left(\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A}\right)\mathbf{x}(t)$$

is negative definite. These conditions provides the simplest LMI system as

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} \prec 0, \qquad \mathbf{P} \succ 0. \tag{2.3}$$

This way, the stability test is to check feasibility of LMIs, that can be done numerically by interior point solvers. For more details, see the works of Boyd [32, 33].

LPV modelling for trajectory tracking control

Consider a typical continuous-time, linear parameter varying system as

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \\ \mathbf{z}(t) \end{bmatrix} = \mathbf{S}(\mathbf{p}(t)) \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \\ \mathbf{u}(t) \end{bmatrix}, \quad \mathbf{S}(\mathbf{p}(t)) = \begin{bmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}_v(\mathbf{p}(t)) & \mathbf{B}_u(\mathbf{p}(t)) \\ \mathbf{C}_y(\mathbf{p}(t)) & \mathbf{D}_{y,v}(\mathbf{p}(t)) & \mathbf{D}_{y,u}(\mathbf{p}(t)) \\ \mathbf{C}_z(\mathbf{p}(t)) & \mathbf{D}_{z,v}(\mathbf{p}(t)) & \mathbf{D}_{z,u}(\mathbf{p}(t)) \end{bmatrix}, \tag{2.4}$$

where the state variables are denoted by $\mathbf{x} \in \mathbb{R}^n$, the input signals by $\mathbf{u} \in \mathbb{R}^{m_u}$, the noise or disturbances by $\mathbf{w} \in \mathbb{R}^{m_w}$, the measured output $\mathbf{y} \in \mathbb{R}^{p_y}$, the performance output $\mathbf{z} \in \mathbb{R}^{p_z}$, and is defined over a set $\mathfrak{P}$ of allowed functions $\mathbf{p} : \mathbb{R}^+ \to \mathbb{R}^N$. Denote $\Omega \subset \mathbb{R}^N$ the set of possible $\mathbf{p}(t)$ values.

Consider a control along a desired trajectory $\mathbf{x}_d(t)$ and its derivative $\dot{\mathbf{x}}_d(t)$. Assume that there exists a desired input $\mathbf{u}_d(\mathbf{x}_d, \dot{\mathbf{x}}_d, \mathbf{p}_{meas})$ generating the desired state trajectories

$$\dot{\mathbf{x}}_d = \mathbf{A}(\mathbf{p}(t))\mathbf{x}_d + \mathbf{B}_u(\mathbf{p}(t))\mathbf{u}_d(\mathbf{x}_d, \dot{\mathbf{x}}_d, \mathbf{p}_{meas}) \tag{2.5}$$

that depend only on parameters that are measured during the process or a-priori known (denoted as $\mathbf{p}_{meas}$), and that the $\mathbf{D}$ matrices are zero. Then the LPV system that describes the difference from the desired state which is to be controlled to zero, can be written as

$$\begin{bmatrix} \Delta\dot{\mathbf{x}}(t) \\ \Delta\mathbf{y}(t) \\ \Delta\mathbf{z}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}_u(\mathbf{p}(t)) & \mathbf{B}_w(\mathbf{p}(t)) \\ \mathbf{C}_y(\mathbf{p}(t)) & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_z(\mathbf{p}(t)) & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}(t) \\ \Delta\mathbf{u}(t) \\ \mathbf{w}(t) \end{bmatrix}, \tag{2.6}$$

where $\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_d(t)$, $\Delta\mathbf{y}(t) = \mathbf{y}(t) - \mathbf{C}_y(\mathbf{p})\mathbf{x}_d(t)$, $\Delta\mathbf{z}(t) = \mathbf{z}(t) - \mathbf{C}_z(\mathbf{p})\mathbf{x}_d(t)$ and $\Delta\mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_d(\mathbf{x}_d, \dot{\mathbf{x}}_d, \mathbf{p})$.

In this way, the control signal $\Delta\mathbf{u}(t) = \mathbf{h}(\Delta\mathbf{x}(t), \mathbf{p}(t))$ designed to control the $\Delta\mathbf{x}$ state to zero with appropriate performance for description (2.4), can be obtained as

$$\mathbf{u}(t) = \mathbf{h}(\mathbf{x}(t) - \mathbf{x}_d(t), \mathbf{p}(t)) + \mathbf{u}_d(\mathbf{x}_d(t), \dot{\mathbf{x}}_d(t), \mathbf{p}(t)). \tag{2.7}$$

Similar, so-called quasi-Linear Parameter Varying system descriptions can be obtained for nonlinear systems as well.

qLPV realizations of parameter-varying nonlinear system

Consider the following input-affine nonlinear system in general,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{q}(t)) + g_u(\mathbf{x}(t), \mathbf{q}(t))\mathbf{u}(t) + g_w(\mathbf{x}(t), \mathbf{q}(t))\mathbf{w}(t), \tag{2.8}$$
$$\mathbf{y}(t) = f_y(\mathbf{x}(t), \mathbf{q}(t)), \tag{2.9}$$
$$\mathbf{z}(t) = f_z(\mathbf{x}(t), \mathbf{q}(t)), \tag{2.10}$$

where $\mathbf{q}$ denotes the external parameters that are independent of the state variables, and Ω_q denotes the corresponding parameter domain.

Suppose there exists an input $\mathbf{u}_d(\mathbf{x}_d(t), \dot{\mathbf{x}}_d(t), \mathbf{q}(t))$ such that a desired state trajectory $\mathbf{x}_d(t)$ is generated from

$$\dot{\mathbf{x}}_d(t) = f(\mathbf{x}_d(t), \mathbf{q}(t)) + g_u(\mathbf{x}_d(t), \mathbf{q}(t))\mathbf{u}_d(\mathbf{x}_d(t), \dot{\mathbf{x}}_d(t), \mathbf{q}(t)) \quad \forall \mathbf{q}(t) \in \Omega_q. \tag{2.11}$$

The corresponding outputs without disturbance are

$$\mathbf{y}_d(t) = f_y(\mathbf{x}_d(t), \mathbf{q}(t)), \tag{2.12}$$
$$\mathbf{z}_d(t) = f_z(\mathbf{x}_d(t), \mathbf{q}(t)). \tag{2.13}$$

Then the error system, where the desired value of states $\Delta\mathbf{x}(t)$ and outputs $\Delta\mathbf{y}(t)$ are zero, is given by

$$\begin{aligned} \Delta\dot{\mathbf{x}}(t) = {} & f(\mathbf{x}(t), \mathbf{q}(t)) - f(\mathbf{x}_d(t), \mathbf{q}(t)) + g_u(\mathbf{x}(t), \mathbf{q}(t))\mathbf{u}(t) - \\ & - g_u(\mathbf{x}_d(t), \mathbf{q}(t))\mathbf{u}_d(\mathbf{x}_d(t), \mathbf{q}(t)) + g_w(\mathbf{x}(t), \mathbf{q}(t))\mathbf{w}(t), \end{aligned} \tag{2.14}$$
$$\Delta\mathbf{y}(t) = f_y(\mathbf{x}(t), \mathbf{q}(t)) - f_y(\mathbf{x}_d(t), \mathbf{q}(t)), \tag{2.15}$$
$$\Delta\mathbf{z}(t) = f_z(\mathbf{x}(t), \mathbf{q}(t)) - f_y(\mathbf{x}_d(t), \mathbf{q}(t)), \tag{2.16}$$

which can be written into quasi-LPV form:

$$\begin{bmatrix} \Delta\dot{\mathbf{x}}(t) \\ \Delta\mathbf{y}(t) \\ \Delta\mathbf{z}(t) \end{bmatrix} = \mathbf{S}(\mathbf{p}(t)) \begin{bmatrix} \Delta\mathbf{x}(t) \\ \Delta\mathbf{u}(t) \\ \Delta\mathbf{w}(t) \end{bmatrix} \tag{2.17}$$

for all $\mathbf{x}(t)$, $\mathbf{x}_d(t)$, $\mathbf{q}(t)$, where $\mathbf{p}(t) = \begin{bmatrix} \mathbf{x}(t) & \mathbf{x}_d(t) & \mathbf{q}(t) \end{bmatrix}$.

It is important to note, that – as the name "realization" suggests – this description is not unique at all except at $\mathbf{x}(t) = \mathbf{x}_d(t)$ state.

If a controller is designed to stabilize it with given performance as $\Delta\mathbf{u}(t) = h(\mathbf{x}(t) - \mathbf{x}_d(t), \mathbf{p}(t))$, the control input of the plant can be obtained as $\mathbf{u}(t) = \Delta\mathbf{u}(t) + \mathbf{u}_d(\mathbf{x}_d(t), \mathbf{q}(t))$.

Jacobian linearization. The Jacobian linearization is a special case of the description above, where the $\mathbf{x}(t) \approx \mathbf{x}_d(t)$ case is modelled for control analysis or synthesis. In this case, the parameter vector can be written as $\mathbf{p} = \begin{bmatrix} \mathbf{x}_d(t) & \mathbf{q}(t) \end{bmatrix}$ and the elements of $\mathbf{S}(\mathbf{p})$:

$$\begin{aligned} \mathbf{A}(\mathbf{p}) &= \frac{\partial f(\mathbf{x},\mathbf{q})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_d}, & \mathbf{B}_u(\mathbf{p}) &= g_u(\mathbf{x}_d, \mathbf{q}), \\ & & \mathbf{B}_w(\mathbf{p}) &= g_w(\mathbf{x}_d, \mathbf{q}), \\ \mathbf{C}_y(\mathbf{p}) &= \frac{\partial f_y(\mathbf{x},\mathbf{q})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_d}, & \mathbf{C}_z(\mathbf{p}) &= \frac{\partial f_z(\mathbf{x},\mathbf{q})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_d}. \end{aligned}$$

But in practice, the LPV systems obtained based on linearization does not ensure the control performance or even stability [105].

For more details about LPV/qLPV modell based control, see the works [30, 41, 121, 153].

Definite condition-based criteria for state feedback controller design

To illustrate the relevance of definite conditions considering systems in forms (2.6) or (2.17), this subsection recites the definite conditions that must be fulfilled to obtain a stable system, and the criteria to obtain given H_2 or H_∞ norms or pole placements.

For sake of brevity, only state feedback design

$$\mathbf{u} = \mathbf{M}(\mathbf{p})\mathbf{X}(\mathbf{p})^{-1}\mathbf{x}, \tag{2.18}$$

is considered, where

$$\mathbf{M} : \Omega \to \mathbb{R}^{m_u \times n}, \tag{2.19}$$
$$\mathbf{X} : \Omega \to \mathbb{R}^{n \times n}, \tag{2.20}$$

but the functions cannot depend on the parameters, that are not known, measured or estimated during the control process.

The following lemmas show definite conditions for different control criteria and depend on affinely from the functions (2.19) and (2.20) as LMIs from the variables. The following notations are used for sake of brevity: $\mathrm{AX}(\mathbf{p}) = \mathbf{A}(\mathbf{p})\mathbf{X}(\mathbf{p}) + \mathbf{B}_u(\mathbf{p})\mathbf{M}(\mathbf{p})$, and $\mathrm{CX}(\mathbf{p}) = \mathbf{C}_z(\mathbf{p})\mathbf{X}(\mathbf{p}) + \mathbf{D}_{z,u}(\mathbf{p})\mathbf{M}(\mathbf{p})$.

Lemma 2.1 (Stability)**.** *The LPV system with state feedback* (2.18) *is stable for all allowed parameter trajectories, if there exists functions* (2.19)-(2.20) *such that*

$$\mathbf{X}(\mathbf{p}) \succ 0, \qquad \mathrm{Sym}(\mathrm{AX}(\mathbf{p})) - \dot{\mathbf{X}}(\mathbf{p}) \prec 0 \tag{2.21}$$

for all allowed parameter trajectories.

Lemma 2.2 (H_2 criteria)**.** *The LPV system with state feedback* (2.18) *is stable and H_2 norm of the system is less than γ_2 for all allowed parameter trajectories, if there exists functions* (2.19)-(2.20) *and $\mathbf{R} : \Omega \to \mathbb{R}^{p_z \times p_z}$ such that*

$$\mathbf{X}(\mathbf{p}) \succ 0, \qquad \mathrm{Tr}(\mathbf{R}(\mathbf{p})) < \gamma_2^2, \tag{2.22}$$

$$\begin{bmatrix} \mathbf{R}(\mathbf{p}) & \mathrm{CX}(\mathbf{p}) \\ * & \mathbf{X}(\mathbf{p}) \end{bmatrix} \succ 0, \tag{2.23}$$

$$\begin{bmatrix} \dot{\mathbf{X}}(\mathbf{p}) - \mathrm{Sym}(\mathrm{AX}(\mathbf{p})) & \mathbf{B}_v(\mathbf{p}) \\ * & \mathbf{I} \end{bmatrix} \succ 0, \qquad \mathbf{D}_{zv}(\mathbf{p}(t)) = \mathbf{0} \tag{2.24}$$

for all allowed parameter trajectories.

Lemma 2.3 (H_∞ criteria based on the Bounded Real Lemma [54])**.** *The LPV system with state feedback* (2.18) *is stable and H_2 norm of the system is less than γ_2 for all allowed parameter trajectories, if there exists functions* (2.19)-(2.20) *such that*

$$\mathbf{X}(\mathbf{p}) \succ 0, \qquad \begin{bmatrix} \dot{\mathbf{X}}(\mathbf{p}) - \mathrm{Sym}(\mathrm{AX}(\mathbf{p})) & \mathbf{B}_v(\mathbf{p}) & (\mathrm{CX}(\mathbf{p}))^T \\ * & \gamma_\infty \mathbf{I} & \mathbf{D}_{z,v}^T(\mathbf{p}) \\ * & * & \gamma_\infty \mathbf{I} \end{bmatrix} \succ 0 \tag{2.25}$$

for all allowed parameter trajectories.

The poles s of the closed-loop system characterizes the settling time and overshoot the system. For complex poles $s = -\zeta\omega_n \pm j\omega_d$, ω_n is the natural frequency, ζ denotes the relative damping and ω_d is the damped natural frequency as $\omega_d = \omega_n\sqrt{1-\zeta^2}$ and for real poles $s = -\alpha$, α is the decay rate. By assigning a domain $\mathfrak{D}$ of the complex plane via LMI

$$\mathfrak{D} = \{ s \in \mathbb{C} \mid \mathbf{L}_\mathfrak{D} + \mathrm{Sym}(s\mathbf{M}_\mathfrak{D}) \prec 0 \} \tag{2.26}$$

the poles can be constrained to be within, which is called $\mathfrak{D}$-stability. For more details about matrices $\mathbf{L}_\mathfrak{D}$ and $\mathbf{M}_\mathfrak{D}$, see [45, 46, 71, 123].

Lemma 2.4 (LMI region)**.** *The LPV system with state feedback* (2.18) *is $\mathfrak{D}$-stable considering region* (2.26) *for all allowed parameter trajectories if there exists functions* (2.19)-(2.20) *such that*

$$\mathbf{X}(\mathbf{p}) \succ 0, \qquad \mathbf{L}_{\mathfrak{D}} \otimes \mathbf{X}(\mathbf{p}) + \mathrm{Sym}(\mathbf{M}_{\mathfrak{D}} \otimes \mathrm{A}\mathrm{X}(\mathbf{p})) - \dot{\mathbf{X}}(\mathbf{p}) \prec 0 \tag{2.27}$$

for all allowed parameter trajectories.

Methods for descriptor models were introduced in [31, 148, 152] and line-integral type Lyapunov–function candidates, where the derivative condition does not appear, in [143].

Some approaches are specially developed for discrete-time systems (as in [52, 107]), where the variables of the controller and the Lyapunov function are independent. In these cases, the measure/estimation opportunities do not constrain the multiplicities in the Lyapunov function candidate. The discrete-time case is relaxed by applying a delayed Lyapunov-function in [107]. The uncertainty of the model can be taken into account by combining the method with the norm-bound uncertainty description [31, 168].

Lyapunov-Krasovski functional or Razumikhin theory can be applied to handle systems with time delay [48, 68, 114, 170, 185] and very complex criteria systems are constituted as well: for example, predictive output feedback control [51].

Polytopic LPV/qLPV models based controller design

The parameter dependency of the matrix $\mathbf{S}(\mathbf{p})$ can be handled by constructing a polytopic description defined by vertices $\mathbf{S}_1$, $\mathbf{S}_2$, ..., $\mathbf{S}_J$ as

$$\mathbf{S}(\mathbf{p}) = \mathrm{Co}(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_J). \tag{2.28}$$

Then it can be described as

$$\mathbf{S}(\mathbf{p}) = \sum_{j=1}^{J} h_j(\mathbf{p})\mathbf{S}_j, \tag{2.29}$$

where the $h_j(\mathbf{p})$ weighting functions denote convex combinations.

Considering the controller candidate with constant $\mathbf{M}(\mathbf{p}) = \mathbf{M}$, $\mathbf{X}(\mathbf{p}) = \mathbf{X}$ matrices, the definite conditions in Lemma 2.1, 2.2, 2.3 and 2.4 can be written as

$$\sum_{j=1}^{J} h_j(\mathbf{p})\Gamma_j \prec 0, \tag{2.30}$$

where the Γ_j values come from $\mathbf{S}_j$ matrices, and depend on the $\mathbf{M}$, $\mathbf{X}$ variables. It is negative definite for all convex combinations of the vertices if and only if the vertices are negative definite: $\Gamma_j \prec 0$ for $j = 1, ..., J$. In this way, the problem is transformed to LMIs and can be solved via convex optimization.

Considering the function $\mathbf{M}(\mathbf{p})$ on single polytopic summation as

$$\mathbf{M}(\mathbf{p}) = \sum_{j=1}^{J} h_j(\mathbf{p})\mathbf{M}_j, \tag{2.31}$$

we get the so called Parallel Distributed Compensation (PDC) [171, 182], the definite conditions in Lemma 2.1, 2.2, 2.3 and 2.4 can be written as a double summation

$$\sum_{i=1}^{J}\sum_{j=1}^{J} h_i(\mathbf{p})h_j(\mathbf{p})\Gamma_{ij} \prec 0. \tag{2.32}$$

However, sufficient and necessary conditions can be given only for simple summation. For double ones, there exist various sufficient conditions allowing to apply trade-off between conservatism and computational cost in [61, 84, 118, 150, 178, 183]. For example, the LMIs

$$\mathbf{X} \succ 0, \quad \mathrm{Sym}\left(\mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_i\right) \prec 0 \qquad \forall i, \tag{2.33}$$

$$\mathrm{Sym}\left((\mathbf{A}_i + \mathbf{A}_j)\mathbf{X} - (\mathbf{B}_i\mathbf{M}_j + \mathbf{B}_j\mathbf{M}_i)\right) \preceq 0 \qquad \forall i, j < i. \tag{2.34}$$

guarantee condition (2.21) considering (2.31), (2.29) where matrices $\mathbf{A}_j$, $\mathbf{B}_j$ are appropriate partitions of $\mathbf{S}_j$ for $j = 1, ..., J$ and $\mathbf{X}(\mathbf{p}) = \mathbf{X}$ by applying method in [183].

The functions can be defined on multiple summations e.g.,

$$\mathbf{M}(\mathbf{p}) = \sum_{i=1}^{J}\sum_{j=1}^{J}\sum_{k=1}^{J} h_i(\mathbf{p})h_j(\mathbf{p})h_k(\mathbf{p})\mathbf{M}_{ijk}. \tag{2.35}$$

In this case, the definite conditions are on multiple summations and LMIs can be obtained via the method based on the Pólya-theorem, see [150].

Considering parameter dependent $\mathbf{X}(\mathbf{p})$, its derivative must be handled. If $\dot{\mathbf{X}}(\mathbf{p}) \neq 0$, the $|\dot{w}_j(\mathbf{p})|$ values can be bounded as in [103, 127, 143]. By taking into account the weighting functions, the approaches can be relaxed as well [147, 149, 151].

The non-convex nature of robust output-feedback design appears in this case as well. It results in Bilinear Matrix Inequalities motivating research on approaches to consider convex subsets of the solutions [47, 110, 141] and local optimisation methods on BMIs [23, 70, 173].

Chapter 3

Tensor-Product Model Transformation based control

This chapter briefly discusses the basics of Tensor Model Transformation highlighting its application in control analysis and synthesis. The methodology consists of three main parts:

1. Extension of tensor algebra for multivariate functions and its key concept: the HOSVD-based TP form, which is a compact, canonical form, where the parameter dependencies are separated.
2. Algorithms to derive Polytopic Tensor Product form from the HOSVD-based one.
3. Polytopic model-based controller design methods, which can be immediately applied to the Polytopic TP form. (The TP structure can be exploited to relax the extraction of multiple polytopic summations [7].)

The chapter is structured as follows: First basic definitions and properties of the used tensor algebra are shown. Then the TP model transformation is introduced for scalar functions, and it is extended for parameter dependent system matrices of LPV/qLPV models.

Related concepts of tensor algebra

The tensor algebra proposed by Hitchcock [72] defines the properties and the operations based on the n-mode unfold of the tensor. For its derivation (and restoration of the tensor) see the following definition, which is also illustrated in Figure 3.1a.

Definition 3.1 (Unfold tensor)**.** *Assume an Nth order tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ *where* $I_n \geq 1$ *for all* $n = 1, ..., N$*. The n-mode matrix unfolding*

$$\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1} \dots I_N I_1 \dots I_{n-1})} \tag{3.1}$$

contains the element $a_{i_1,\ldots,i_N}$ *at the position* (i_n, j_n)*, where*

$$j_n = \sum_{l=n+1}^{N} (i_l - 1) \left(\prod_{m=l+1}^{N} I_m \right) \left(\prod_{m=1}^{n-1} I_m \right) + \sum_{l=1}^{n-1} (i_l - 1) \prod_{m=l+1}^{n-1} I_m + 1. \tag{3.2}$$

Remark 3.2. *Eq.* (3.2) *is an ordering for the elements of* i_n*-th* n*-mode subtensor. Any other ordering rule can be chosen, but its inverse must be applied for restoration.*

It implicates the following n-mode rank definition.

Definition 3.3 (n-mode rank)**.** *The* n*-mode rank of tensor* $\mathcal{A}$*, denoted by* $\operatorname{rank}_n(\mathcal{A})$ *is the number of independent rows in* n*-mode unfold of tensor* $\mathcal{A}$*.*

The tensor can be multiplied n-mode with a matrix as follows, which is illustrated in Figure 3.1b.

Definition 3.4 (n-mode tensor product)**.** *The* n*-mode product of a tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *and the matrix* $\mathbf{U} \in \mathbb{R}^{L \times I_n}$*, denoted by* $\mathcal{A} \times_n \mathbf{U}$*, is a tensor with size* $I_1 \times \cdots \times I_{n-1} \times L \times I_{n+1} \times \cdots \times I_N$*, and its elements are given by*

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

The tensor product definition has the following properties:

Lemma 3.5 (Tensor product commutativity)**.** *Given the tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *and the matrices* $\mathbf{U} \in \mathbb{R}^{J \times I_n}$*,* $\mathbf{V} \in \mathbb{R}^{N \times I_l}$ *($n \neq l$), one has*

$$(\mathcal{A} \times_n \mathbf{U}) \times_l \mathbf{V} = (\mathcal{A} \times_l \mathbf{V}) \times_n \mathbf{U}.$$

Lemma 3.6 (Combination of tensor products)**.** *Given the tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *and the matrices* $\mathbf{U} \in \mathbb{R}^{J \times I_n}$*,* $\mathbf{V} \in \mathbb{R}^{M \times J}$*, one has*

$$(\mathcal{A} \times_n \mathbf{U}) \times_n \mathbf{V} = \mathcal{A} \times_n (\mathbf{V}\mathbf{U}).$$

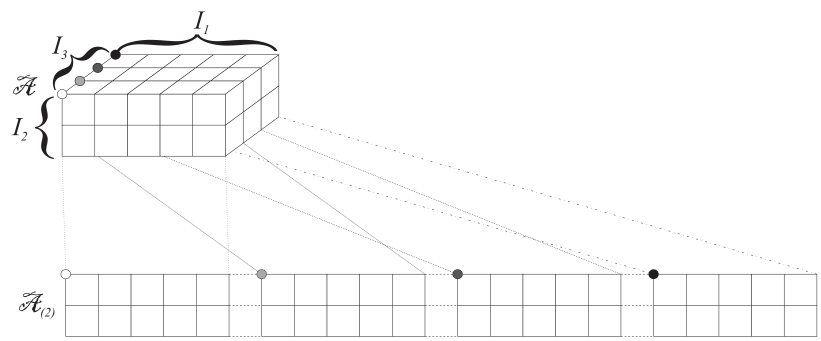
Furthermore, the space of tensors becomes a Hilbert-space with the following inner product and norm definitions.

Definition 3.7 (Scalar product)**.** *The scalar product* $< \mathcal{A}, \mathcal{B} >$ *of two tensors* $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *is defined as*

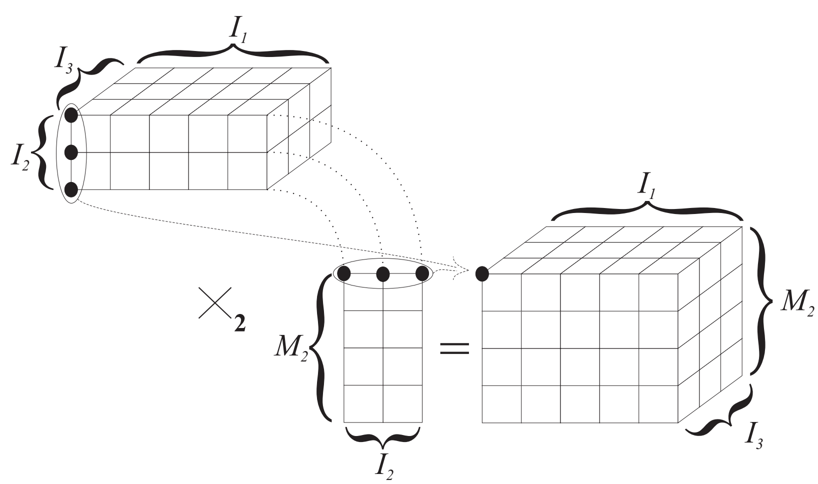
$$< \mathcal{A}, \mathcal{B} > = \sum_{i_1} \cdots \sum_{i_N} a_{i_1,\ldots,i_N} b_{i_1,\ldots,i_N},$$

and tensors with zero scalar product are called orthogonal.

Then, the Frobenius-norm:



(a) Example for 2-mode tensor unfold ($I_1 = 6$, $I_2 = 3$, $I_3 = 4$)



(b) Example for 2-mode tensor product ($I_1 = 6$, $I_2 = 3$, $I_3 = 4$, $M_2 = 5$)

Figure 3.1: Illustrations for tensor operations

Definition 3.8 (Frobenius-norm)**.** *The Frobenius-norm of a tensor* $\mathcal{A}$ *is given by* $||\mathcal{A}|| = \sqrt{<\mathcal{A}, \mathcal{A}>}$.

Notation 3.9 (Subtensor)**.**

Lathauwer in [49] defined the Higher Order Singular Value Decomposition (HOSVD).

Definition 3.10 (HOSVD)**.** *The decomposition of the tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *as*

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \cdots \times_N \mathbf{U}^{(N)} = \mathcal{S} \underset{n=1}{\overset{N}{\boxtimes}} \mathbf{U}^{(n)} \tag{3.3}$$

is called an HOSVD if

- *the matrices* $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ *are orthogonal for all* $n = 1, ..., N$,

- *the n-mode subtensors of the $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ core tensor hold the following properties for all $n = 1, ..., N$*

  1. *all-orthogonality:* $< \mathcal{S}_{i_n=a}, \mathcal{S}_{i_n=b} >= \delta_{ab}\sigma_a^{(n)2}$ *for all* $1 \leq a, b \leq I_n$*, where the* $\sigma_a^{(n)}$ *norm of subtensor* $\mathcal{S}_{i_n=a}$ *is called the a-th n-mode singular value,*
  2. *ordering:* $\sigma_1^{(n)} \geq \sigma_2^{(n)} \geq \cdots \geq \sigma_{I_n}^{(n)} \geq 0$.

Definition 3.11. *(CHOSVD) Consider a HOSVD of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ with the same notations as in* (3.3)*.*

Disregarding the last subtensors with zero elements of tensor $\mathcal{S}$ and the corresponding columns of matrices $\mathbf{U}^{(n)}$, a Compact HOSVD (CHOSVD) can be obtained, where the singular values are positive, the core tensor has the size $R_1 \times \cdots \times R_N$ and the n-mode singular matrix has the size $I_n \times R_n$ for all $n = 1, ..., N$, where $R_n = \mathrm{rank}_n\mathcal{A}$.

Definition 3.12. *(RHOSVD) Consider a CHOSVD of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ with the same notations as in* (3.3)*.*

Disregarding its last subtensors of tensor $\mathcal{S}$ and the corresponding columns of matrices $\mathbf{U}^{(n)}$ in one or more mode, a Reduced HOSVD (RHOSVD) can be obtained, where the singular values are positive, the core tensor has the size $L_1 \times \cdots \times L_N$ and the n-mode singular matrix has the size $I_n \times L_n$ for all $n = 1, ..., N$, where $L_n \leq \mathrm{rank}_n\mathcal{A}$.

Lemma 3.13. *(Approximation error of RHOSVD) The error of approximation based on n-mode rank reduction via RHOSVD can be bounded as*

$$||\mathcal{A} - \mathcal{A}^{RHOSVD}||^2 \leq \sum_{i_1=L_1+1}^{R_1} \sigma_{i_1}^{(1)2} + \sum_{i_2=L_2+1}^{R_2} \sigma_{i_2}^{(2)2} + \cdots + \sum_{i_N=L_N+1}^{R_N} \sigma_{i_N}^{(N)2} \tag{3.4}$$

The inequality is sharp if the reduction is applied in only one n-mode, as in the classical Eckhart-Young theorem [57].

It is important to denote that the core tensor of RHOSVD forms does not hold the properties of HOSVD, only after performing HOSVD again on it. Furthermore, the error is minimal if only one n-mode rank is reduced.

Lemma 3.14 (Best $(r_1, r_2, \ldots, r_N)$ rank approximation)**.** *Consider an Nth order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$. The $\hat{\mathcal{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ tensor with ranks $rank_n(\hat{\mathcal{A}}) = R_n \leq I_n$, that minimizes the error function*

$$f(\hat{\mathcal{A}}) = ||\mathcal{A} - \hat{\mathcal{A}}|| \tag{3.5}$$

can be written as

$$\hat{\mathcal{A}} = \mathcal{B} \mathop{\boxtimes}_{n=1}^{N} \mathbf{U}^{(n)} \quad where \;\; \mathcal{B} = \mathcal{A} \mathop{\boxtimes}_{n=1}^{N} \mathbf{U}^{(n)T}, \tag{3.6}$$

and it is equivalent to the following maximization

$$\max_{\{\mathbf{U}^{(n)}|\mathbf{U}^{(n)T}\mathbf{U}^{(n)}=\mathbf{E}_{R_n}\}_{n=1..N}} \mathcal{A} \underset{n=1}{\overset{N}{\boxtimes}} \mathbf{U}^{(n)T}. \tag{3.7}$$

For more details about the HOOI algorithms, see [50, 75, 154].

Scalar Tensor Product functions

The goal of Tensor Model transformation is to generalize the concept of tensor algebra for multivariate functions as tensor product functions, see the next definition similarly to quasimatrix concept in [21, 175, 176, 177].

Definition 3.15 (Tensor Product (TP) function)**.** *The following form*

$$f(\mathbf{p}) = \mathcal{B} \underset{n=1}{\overset{N}{\boxtimes}} \mathbf{w}^{(n)}(p_n) \tag{3.8}$$

of a real $f : \Omega \to \mathbb{R}$ function is called TP function, where

- *it is defined on the $\Omega = [\underline{p}_1, \overline{p}_1] \times [\underline{p}_2, \overline{p}_2] \times \cdots \times [\underline{p}_N, \overline{p}_N] \subset \mathbb{R}^N$ hyperrectangular domain,*
- *the real $\mathcal{B}$ tensor has sizes $I_1 \times \cdots \times I_N$,*
- *the n-mode weighting functions are $\mathbf{w}_n : [\underline{p}_n, \overline{p}_n] \to \mathbb{R}^{I_n}$, respectively.*

Remark 3.16. *In general, functions cannot be written into TP form with finite I_n sizes. See function $f(\mathbf{p}) = 1/(p_1 + p_2)$.*

The HOSVD-based form is canonical because of its uniqueness properties.

Definition 3.17 (HOSVD based TP form)**.** *The following TP form*

$$f(\mathbf{p}) = \mathcal{S} \underset{n=1}{\overset{N}{\boxtimes}} \mathbf{u}^{(n)}(p_n), \tag{3.9}$$

where $\mathcal{S} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\mathbf{u}^{(n)} : [\underline{p}_n, \overline{p}_n] \to \mathbb{R}^{I_n}$ is called HOSVD based, if for all $n = 1, ..., N$,

- *the weighting functions in $\mathbf{u}^{(n)}(\cdot)$ are orthonormal as*

$$\int_{p_n=\underline{p}_n}^{\overline{p}_n} u_i^{(n)}(p_n) u_j^{(n)}(p_n) dp_n = \delta_{ij} \quad \text{for all } i, j = 1, ..., I_n. \tag{3.10}$$

- *the n-mode subtensors of $\mathcal{S}$ have the following properties*

- *all-orthogonality:* $< \mathcal{S}_{i_n=a}, \mathcal{S}_{i_n=b} >= \delta_{ab}\sigma_a^{(n)2}$ *for all* $1 \leq a, b \leq I_n$*, where the* $\sigma_a^{(n)}$ *norm of subtensor* $\mathcal{S}_{i_n=a}$ *is called the a-th n-mode singular value,*
- *ordering:* $\sigma_1^{(n)} \geq \sigma_2^{(n)} \geq \cdots \geq \sigma_{I_n}^{(n)} > 0.$

It can generally be computed from an arbitrary TP form. First, its weighting functions must be orthonormalized as

$$\mathbf{w}^{(n)}(p_n) = \mathbf{T}^{(n)}\tilde{\mathbf{u}}^{(n)}(p_n), \tag{3.11}$$

where the $\tilde{\mathbf{u}}^{(n)}(p_n)$ functions are orthonormal. Then the TP form can be written as

$$\mathcal{C} \mathop{\boxtimes}_{n=1}^{N} \tilde{\mathbf{u}}^{(n)}(p_n), \quad \text{where} \quad \mathcal{C} = \mathcal{B} \mathop{\boxtimes}_{n=1}^{N} \mathbf{T}^{(n)}. \tag{3.12}$$

Then by computing CHOSVD of tensor $\mathcal{C}$ as

$$\mathcal{C} = \mathcal{S} \mathop{\boxtimes}_{n=1}^{N} \mathbf{U}^{(n)}, \tag{3.13}$$

the core tensor of an HOSVD based form is $\mathcal{S}$ with weighting function $\mathbf{u}^{(n)}(p_n) = \tilde{\mathbf{u}}^{(n)}(p_n)\mathbf{U}^{(n)}$.

Property 3.18 (Uniqueness)**.** *Consider the HOSVD-based TP form in* (3.9)*. Then the* $\sigma_d^{(n)}$ *singular values are unique. Further denote their multiplicities by* $(m_1^{(n)}, m_2^{(n)}, \dots)$ *such that*

$$\underbrace{\sigma_1^{(n)} = \cdots = \sigma_{m_1^{(n)}}^{(n)}}_{m_1^{(n)}} > \underbrace{\sigma_{m_1^{(n)}+1}^{(n)} = \cdots = \sigma_{m_1^{(n)}+m_2^{(n)}}^{(n)}}_{m_2^{(n)}} > \dots \sigma_{I_n} > 0.$$

for all $n = 1, ..., N$*. Then the following and only the following forms are also HOSVD-based TP forms*

$$f(\mathbf{p}) = \left(\mathcal{S} \mathop{\boxtimes}_{n=1}^{N} \mathbf{T}^{(n)}\right) \mathop{\boxtimes}_{n=1}^{N} \left(\mathbf{u}^{(n)}(\mathbf{p}^{(n)})\mathbf{T}^{(n)T}\right), \tag{3.14}$$

where $\mathbf{T}^{(n)}$ *is a block-diagonal matrix constructed by arbitrary orthogonal matrices with sizes* $m_1^{(n)} \times m_1^{(n)}$*,* $m_2^{(n)} \times m_2^{(n)}$*, ...*

It allows the reduction of n-mode size with minimal error.

Lemma 3.19 (Complexity reduction)**.** *Reduce the rank of a TP function with n-mode ranks* $(I_1, \dots, I_N)$ *given in HOSVD based TP form to ranks* $(r_1, \dots, r_N)$*, where* $r_n \leq I_n$ *can be obtained by disregarding the* $(r_n+1), ..., I_n$*-th n-mode subtensors of the core tensor and the corresponding n-mode weighting functions for all* $n = 1, ..., N$*. In this case, the approximation error can be bounded as*

$$\int_{\mathbf{p}\in\Omega} ||f(\mathbf{p}) - f^{reduced}(\mathbf{p})||^2 dp_1 \dots dp_N \leq \sum_{i=r_1+1}^{I_1} \sigma_i^{(1)2} + \cdots + \sum_{i=r_N+1}^{I_N} \sigma_i^{(N)2} = \sum_{n=1}^{N} \sum_{i=r_n+1}^{I_n} \sigma_i^{(n)2}.$$

Its properties and derivation are detailed in [14, 16].

Numeric reconstruction of HOSVD based form

The subsection shows the algorithm for obtaining approximating reconstruction of HOSVD based TP form for functions given not in TP form. The algorithm is based on discretization on an equidistant grid, as its densities are increased to infinity, the error of the form converges to zero, [160].

Specifically, we consider the bounded hyperrectangular parameter domain given by

$$\Omega = [\underline{p}_1, \overline{p}_1] \times [\underline{p}_2, \overline{p}_2] \times \cdots \times [\underline{p}_N, \overline{p}_N] \subset \mathbb{R}^N, \tag{3.15}$$

as the transformation space.

Definition 3.20 (Discretisation grid)**.** *The equidistant rectangular grid in the transformation space with sizes $M_1 \times M_2 \times \cdots \times M_N$ which points can be described as*

$$\mathbf{g}_{m_1,m_2,\ldots,m_N} = \begin{bmatrix} g^{(1)}_{m_1} & g^{(2)}_{m_2} & \ldots & g^{(N)}_{m_N} \end{bmatrix}, \tag{3.16}$$

where $g^{(n)}_m = \underline{p}_n + \frac{\overline{p}_n - \underline{p}_n}{M_n - 1}(m-1)$.

Definition 3.21 (Discretised function)**.** *The tensor $\mathcal{F}^{D(\Omega,M)} \in \mathbb{R}^{M_1 \times \cdots \times M_N}$, which denotes the discretised values of $f(\mathbf{p})$ function in the gridpoints on Ω domain with sizes $\mathbf{M}$ is given as*

$$f^{D(\Omega,M)}_{m_1,\ldots,m_N} = f(\mathbf{g}_{m_1,\ldots,m_N}). \tag{3.17}$$

The goal of TP model transformation is to transform a given function $f(\mathbf{p})$ into HOSVD based TP form in a given transformation space. The fundamental idea is to reconstruct the function not only at the grid points of the hyperrectangular grid but providing the weighting functions on more and denser grids.

Algorithm 3.22 (TP model transformation)**.**

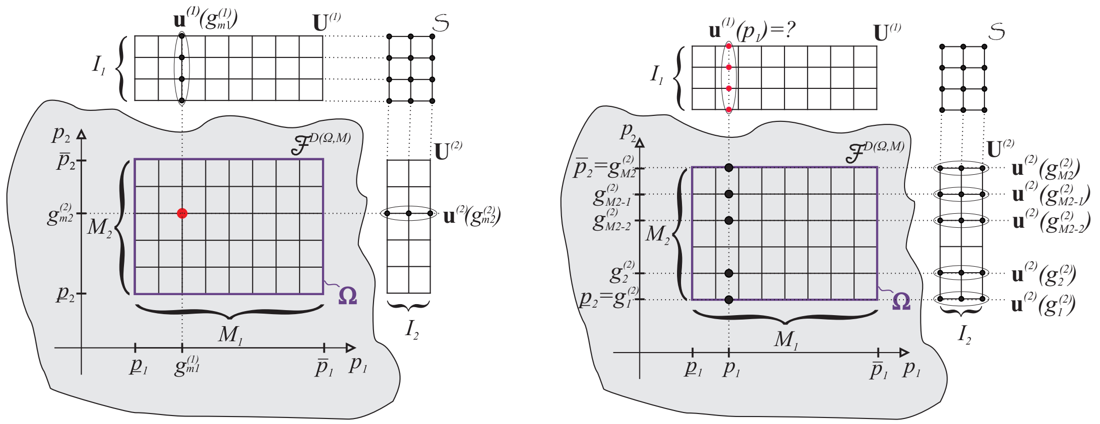
Step 1 (Discretisation)**.** *This step creates the discretized tensor for the function.*

First choose the Ω transformation domain and the $\mathbf{M}$ grid sizes that defines the discreatisation grid. Then obtain the tensor $\mathcal{F}^{D(\Omega,M)} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_N}$ from the values at the gridpoints.

Step 2 (Extracting the discretised TP function)**.** *This step reveals the TP structure of the given function. We use HOSVD to find the TP structure of the function. Obtain CHOSVD on tensor $\mathcal{F}^{D(\Omega,M)}$ in the following form*

$$\mathcal{F}^{D(\Omega,M)} = \mathcal{S} \mathop{\boxtimes}_{n=1}^{N} \mathbf{U}^{(n)}, \tag{3.18}$$

where the size of tensor $\mathcal{D}$ is $R_1 \times R_2 \times \cdots \times R_N$ and $R_n = \operatorname{rank}_n(\mathcal{F}^{D(\Omega,M)})$.



(a) Discretisation and the discrete TP form on equidistant grid

(b) Derivation of continuous weighting functions

Figure 3.2: TP model transformation (N=2)

Step 3 (Reconstruction of the continuous TP function)**.** *The rows of the* $\mathbf{U}^{(n)}$ *matrices give the weighting function values at the gridpoints as* $\mathbf{u}^{(n)}_{m_n} = \mathbf{u}^{(n)}(g^{(n)}_{m_n})$*. Furthermore, approximating* $\mathbf{u}^{(n)}(p_n)$ *value between them (in general) can be determined in the following way: Choose the* X *set from the* Ω *domain as for all* $\mathbf{x} \in X$*:* $x_n = p_n$ *and the other values are on the gridlines (there exists* m_l *such that* $x_l = g^{(l)}_{m_l}$ *for all* $l \neq n$*). Then the following equation should be fulfilled for all* $\mathbf{x} \in X$

$$f(\mathbf{x}) = \left(\mathcal{S} \underset{l=1, l\neq n}{\overset{N}{\boxtimes}} \mathbf{u}^{(l)}(x_l) \right) \times_n \mathbf{u}^{(n)}(p_n). \tag{3.19}$$

Then the best approximation can be achieved after applying n*-mode unfold, vectorizing the equations and applying pseudo-inverse as*

$$\mathbf{u}^{(n)}(p_n) = \left[\dots \quad f(\mathbf{x}) \quad \dots \right]_{\mathbf{x}\in X} \left[\dots \quad \left(\mathcal{S} \underset{l=1, l\neq n}{\overset{N}{\boxtimes}} \mathbf{u}^{(l)}(x_l) \right)_{(n)} \quad \dots \right]^{\dagger}_{\mathbf{x}\in X}. \tag{3.20}$$

The discretisation and the resulting discrete TP form is shown in Figure 3.2a and the determination of continuous weighting functions in Figure 3.2b.

In practice, interpolation is applied instead of Step 3. For its background, see [40] and, for more details about convergence of discretisation based algorithms, see [160].

Approximation of the original function

In some cases, the function can exactly be described only via a TP function with sizes up to infinity – theoretically. Their exact numerical reconstruction is not possible with finite M_n grid sizes, while only approximating descriptions can be achieved.

Approximation of HOSVD properties

The core tensor always holds the orthogonality and ordering properties. However, the reconstructed weighting functions do not hold orthonormality conditions. The error of orthogonality converges to zero as M_n values are increased to infinity, while their norms converge to zero in the meantime.

Derivation of Polytopic TP Forms

For control purposes, the polytopic forms are of special significance.

Definition 3.23 (Polytopic TP form)**.** *The following TP function*

$$f(\mathbf{p}) = \mathcal{B} \boxtimes_{n=1}^{N} \mathbf{w}^{(n)}(p_n) \tag{3.21}$$

is called polytopic if the weights $\mathbf{w}^{(n)}(p_n)$ denote convex combinations for all $n = 1, ..., N$ and $p_n \in [\underline{p}_n, \overline{p}_n]$ as

$$\sum_{i=1}^{I_n} w_i^{(n)}(p_n) = 1 \qquad w_i^{(n)}(p_n) \geq 0 \quad \forall n = 1, ..., N, \quad i = 1, ..., I_n, \quad p_n \in [\underline{p}_n, \overline{p}_n]. \tag{3.22}$$

It is derived from the HOSVD-based TP form via algebraic manipulation of the weighting matrices as

$$\mathbf{u}^{(n)}(p_n) = \mathbf{w}^{(n)}(p_n)\mathbf{T}^{(n)}, \tag{3.23}$$

where $\mathbf{T}^{(n)}$ is called transformation matrix.

By performing the operation for all $n = 1, ..., N$, the polytopic TP form can be obtained as

$$f(\mathbf{p}) = \mathcal{S} \boxtimes_{n=1}^{N} \left(\mathbf{w}^{(n)}(p_n)\mathbf{T}^{(n)}\right) = \left(\mathcal{S} \boxtimes_{n=1}^{N} \mathbf{T}^{(n)}\right) \boxtimes_{n=1}^{N} \mathbf{w}^{(n)}(p_n). \tag{3.24}$$

If complexity reduction was applied, better approximation can be obtained by computing the core tensor as

$$\mathcal{S}^{poly} = \mathcal{F}^{D(\Omega,M)} \boxtimes_{n=1}^{N} \mathbf{W}^{(n)\dagger}, \tag{3.25}$$

where the $\mathbf{W}^{(n)}$ matrices are the new discretised weighting functions.

The following criteria are used to characterize the polytopic TP forms based on their weighting functions.

Definition 3.24 (SNNN type weighting functions)**.** *Weighting functions which denote convex combinations: they have positive values, and the sum of their values is one for all p_n.*

Definition 3.25 (Normal (NO) type weighting functions)**.** *The maximal values of the weighting functions are one, they have positive values, and the sum of the values is one for all p_n.*

Definition 3.26 (Inverse Relaxed normal (IRNO) type weighting functions)**.** *The minimum of the weighting functions are zeros, their maximal values are the same, and the sum of the values is one for all p_n.*

Definition 3.27 (Close to Normal (CNO) type weighting functions)**.** *The maximal values are as large as possible, they have positive values, and the sum of the values is one for all p_n.*

It is easy to see that the defined forms are not unique. They are used for outputs of the corresponding algorithms, which were called manipulation of the weighting functions or convex hull generation, see [12, 180, 17, 15].

TP model transformation for (q)LPV models

The first motivation of the methodology is its control oriented application to derive polytopic models for LPV/qLPV models and to apply LMI based controller design [15, 19]. Recall the formalism of (q)LPV models

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \\ \mathbf{z}(t) \end{bmatrix} = \mathbf{S}(\mathbf{p}(t)) \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \\ \mathbf{v}(t) \end{bmatrix}, \tag{3.26}$$

where $\mathbf{S}(\mathbf{p}) : \Omega \to \mathbb{R}^{O\times I}$ and $\Omega = [\underline{p}_1, \overline{p}_1] \times \cdots \times [\underline{p}_N, \overline{p}_N]$. The main goal is to find polytopic TP form for the $\mathbf{S}(\mathbf{p})$ function as

$$\mathbf{S}(\mathbf{p}) = \mathcal{S}^{poly} \mathop{\boxtimes}_{n=1}^{N} \mathbf{w}^{(n)}(p_n), \tag{3.27}$$

where $\mathcal{S}^{poly} \in \mathbb{R}^{I_1\times\cdots\times I_N\times O\times I}$ and the values of $\mathbf{w}^{(n)} : [\underline{p}_n, \overline{p}_n] \to \mathbb{R}^{I_n}$ functions denote convex combinations, which is called Polytopic TP model.

Remark 3.28 (Notation)**.** *The tensor-product form* (3.27) *results in a tensor with sizes* $(\underbrace{1\times\cdots\times 1}_{N} \times O \times I)$*. After the multiplications, an* $(N+1)$*-mode unfold should be performed to obtain the desired matrix with size* $(O\times I)$ *as*

$$\mathbf{S}(\mathbf{p}) = \left(\mathcal{S} \mathop{\boxtimes}_{n=1}^{N} \mathbf{u}^{(n)}(p_n)\right)_{(N+1)}. \tag{3.28}$$

But in the remaining part of the chapter, we do not denote this operation as the related literature.

The HOSVD based TP model can be similarly defined for Def. 3.17. For its numeric derivation, the TP model transformation can be applied with the following modifications:

- The discretised tensor $\mathcal{F}^{D(\Omega,M)}$ has the size $M_1 \times \cdots \times M_N \times O \times I$.
- HOSVD is performed only in the $n = 1, ..., N$ modes of tensor $\mathcal{F}^{D(\Omega,M)}$.
- In Step 3, the equation

  $$\mathbf{S}(\mathbf{x}) = \left(\mathcal{S} \underset{l=1, l \neq n}{\overset{N}{\boxtimes}} \mathbf{u}^{(l)}(x_l) \right) \times_n \mathbf{u}^{(n)}(p_n)$$

  that should be fulfilled for all $\mathbf{x} \in X \subset \Omega$, can be unfold and vectorized as

  $$\begin{bmatrix} \ldots & (\mathbf{S}(\mathbf{x}))_{(3)} & \ldots \end{bmatrix}_{\mathbf{x} \in X} = \mathbf{u}^{(n)}(p_n) \begin{bmatrix} \ldots & \left(\mathcal{S} \underset{l=1, l \neq n}{\overset{N}{\boxtimes}} \mathbf{u}^{(l)}(x_l) \right)_{(n)} & \ldots \end{bmatrix}_{\mathbf{x} \in X},$$

  which can be approximated as

  $$\mathbf{u}^{(n)}(p_n) = \begin{bmatrix} \ldots & (\mathbf{S}(\mathbf{x}))_{(3)} & \ldots \end{bmatrix}_{\mathbf{x} \in X} \begin{bmatrix} \ldots & \left(\mathcal{S} \underset{l=1, l \neq n}{\overset{N}{\boxtimes}} \mathbf{u}^{(l)}(x_l) \right)_{(n)} & \ldots \end{bmatrix}_{\mathbf{x} \in X}^{\dagger}.$$

There are a few algorithms to derive the HOSVD based form without constructing the (typically large) $\mathcal{F}^{D(\Omega,M)}$ tensor (for more details, see [129]) and to decrease computational load by performing truncations in sequences of HOSVD, see [136].

Identification is used in discretisation step for time-delayed systems in the so-called TP^τ model transformation [63] and for other description forms in the so-called Multi TP model transformation [167, 166]. However, if different state variables are used in the grid points, the LMI based control analysis and synthesis does not guarantee the stability and the performance.

The methods in the previous section can be similarly applied to obtain polytopic TP model from the HOSVD based form. There are attempts to interpolate between polytopic TP models in the space of their weighting functions [67, 161, 162].

Summary

The TP model transformation methodology allows applying polytopic model-based control design methods to LPV/qLPV models via numerical transformation. First, the HOSVD based form is obtained, and then the polytopic TP form is constructed. The multi-polytopic structure of the form is not exploited in control design, but only for computational purposes.

Part II

Theoretical Achievements

Motivation

As the previous chapters have shown, the TP model transformation based controller design was born by the marriage of HOSVD for functions, algorithms to derive Polytopic TP model from the HOSVD-based TP form and the polytopic model-based control analysis and synthesis methods – inheriting their properties.

The following questions and problems had arisen in practical applications:

- The separation of parameter dependencies (inheriting from HOSVD based form) is not exploited during controller design, but it can blow up the complexity of the model (measured here as the n-mode ranks) although the property is used only to ensure more simple representation of the form.
- Previous methods to obtain polytopic forms are rarely optimized, and they derive – geometrically – simplex polytopes. It is not investigated how their properties influence the achievable performance and how it could be improved in general, and how the specialties of an actual control problem could be taken into account.
- After complexity reduction, the omitted part should also be taken into consideration in control design. Otherwise, the applied analysis or synthesis method cannot provide the guarantee for stability and performance.
- The benefits of the HOSVD-based form are its compactness and possibility for complexity reduction with minimal error - although this error does not characterize the obtained polytopic form if pseudo inverse was applied. Furthermore, centralization and SVD has to be performed again to derive a polytopic form. These reasons question the necessity of the HOSVD-based form and suggest that another intermediate form should be defined that directly leads to the Polytopic TP forms.

Furthermore, the existing methods have the following computational and theoretical issues:

- The formalism is problematic for non-scalar functions (see Remark 3.28).
- The error of numeric reconstruction converges to zero only as the density of discretization tends to infinity. There is always bias in the approximation.
- The generalization from discretized to continuous weighting functions is laborious.

Motivated by these facts and by benefits of former structure, the dissertation builds up the methodology from scratch to construct an established framework with plenty of tools for various practical problems avoiding the unnecessary conservativeness.

Chapter 4 shows that the derivation of polytopic forms leads to the determination of enclosing polytope for point sets on a usually higher dimensional affine subspace and it concludes the specialties of simplex and non-simplex enclosing polytopes.

Chapter 5 extends the definition of Polytopic TP to avoid the unnecessary separations such that it depends on arbitrarily chosen parameter sets. For its derivation, a multi-affine form is defined based on Chapter 4 and then only enclosing polytopes must be determined. The new definition allows using parameter dependencies with higher multiplicities, which are further applied for constructing LMI criteria and handling the error of necessary complexity reductions caused by separation.

Chapter 6 describes how the vertices of polytopic models influence the achievable performance, and furthermore introduces new methods to derive (near) Minimal Volume Simplex and Non-Simplex enclosing polytopes and gives methods to take into account the control properties of the considered LPV/qLPV models.

Finally, Chapter 7 establishes the Polytopic TP model-based control analysis and synthesis by accounting that the variables can also be in Polytopic TP form on the same multi-polytopic structure, but with arbitrary multiplicities. In this way, special controller candidates and Lyapunov-function candidates can be used, which do not depend on specific parameter sets, while allowing for higher multiplicities on the other ones. The proposed TP algebra can be used to describe the derived definite conditions, and a recursive method is provided to extract them into LMIs/BMIs.

Figure 3.3 visualizes the role of the chapters within the control design workflow.

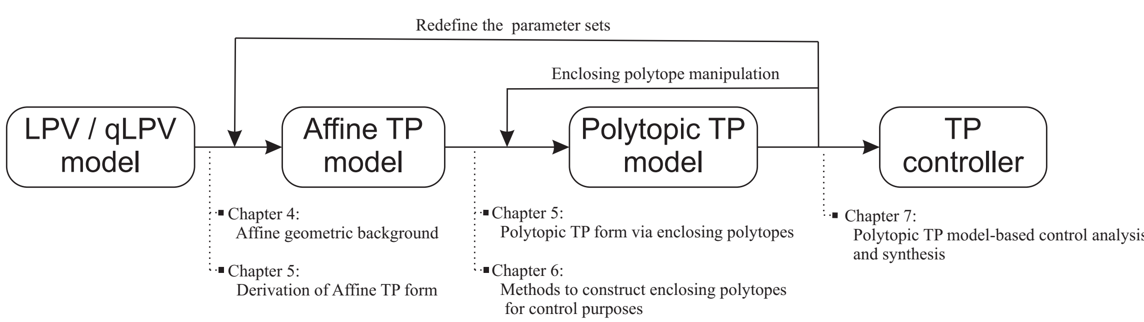


Figure 3.3: The new control design workflow and the role of the chapters

Chapter 4

Geometric interpretation of polytopic form derivation

The chapter considers the problem of determining polytopic form for a multivariate function on a given domain, and it shows its deep connection with the affine geometry. It proposes a factorization called Affine Singular Value Decomposition (ASVD) to represent the affine structure of the image set: It shows the dimension of the affine hull, and it contains an orthogonal basis for it and the offset part as well. The weighting functions are homogeneous and orthonormal coordinates on this basis and offset. Furthermore, it shows the properties of SVD in uniqueness and complexity trade-off opportunity.

Then the chapter shows that polytopic form can be obtained based on the defined, unique affine description by determining an enclosing polytope for the orthonormal coordinates on the affine hull.

The chapter is structured as follows: first Section 4.1 discusses the basic concepts such as the notation of the considered functions and their polytopic description, then Section 4.2 presents the affine geometric background and introduces the Affine Singular Value Decomposition with its numerical reconstruction. After that, Section 4.3 describes how polytopic forms can be obtained by constructing enclosing polytopes on the affine hull. Then Section 4.4 concludes the chapter. Finally, Section 4.5 briefly discusses the relevant proofs for the chapter.

4.1 Basic concepts

Consider a function $\mathfrak{c} : X \to H$ function, where X is a hyperrectangle on real numbers $X = [\underline{x}_1, \overline{x}_1] \times \cdots \times [\underline{x}_L, \overline{x}_L] \subset \mathbb{R}^L$, and H is a Hilbert-space, see Figure 4.1. Furthermore, denote the image set of the function $\mathfrak{c}$ as

$$\mathfrak{C} = \{\mathfrak{c}(\mathbf{x}) | \mathbf{x} \in X\} \subset H, \tag{4.1}$$

and the measure of a set $A \subseteq X$ as $V(A) = \int_A dx_1 \cdot dx_2 \dots dx_L$ and $V(d\mathbf{x})$ will be used as $V(d\mathbf{x}) = dx_1 \cdot dx_2 \dots dx_L$.

First, we define the polytopic description for this set.

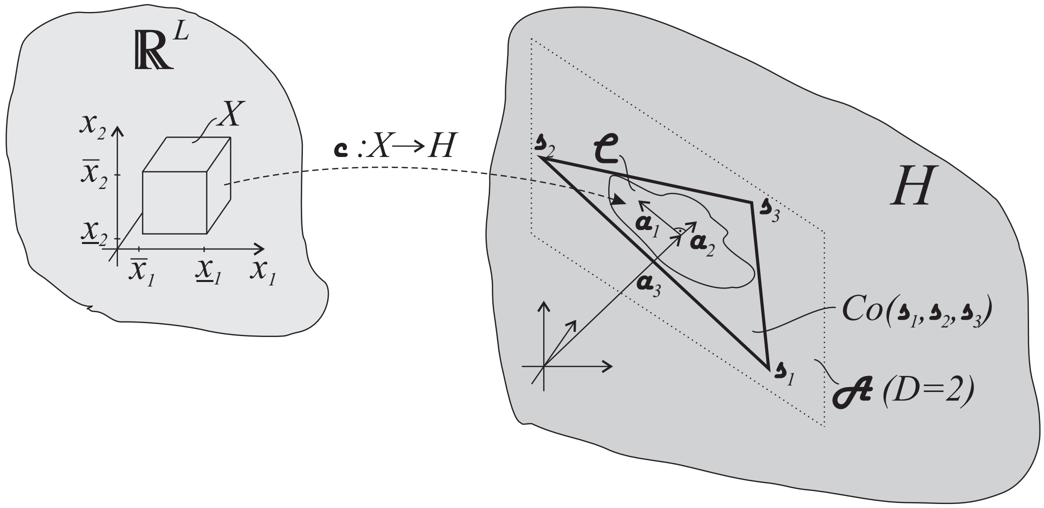


Figure 4.1: Illustration of the used notations: the considered function $\mathfrak{c} : X \to H$, the affine hull $\mathfrak{A}$ of the image set, that is here $D = 2$ dimensional, orthogonal basis $(\mathfrak{a}_1, \mathfrak{a}_2)$, offset to it $\mathfrak{a}_3$, the convex hull of the image set $\mathfrak{C}$, enclosing polytope with vertices $(\mathfrak{s}_1, \mathfrak{s}_2, \mathfrak{s}_3)$

Definition 4.1 (Polytopic description). *The following form*

$$\mathfrak{c}(\mathbf{x}) = \sum_{j=1}^{J} w_j(\mathbf{x})\mathfrak{s}_j, \quad \forall \mathbf{x} \in X \tag{4.2}$$

is called polytopic description if for all $\mathbf{x} \in X$, the $w_1(\mathbf{x}), \ldots, w_J(\mathbf{x})$ values describe convex combinations of the vertices $\mathfrak{s}_j \in H$ as

$$\sum_{j=1}^{J} w_j(\mathbf{x}) = 1, \quad w_j(\mathbf{x}) \geq 0 \qquad j = 1, ..., J \;\; \forall \mathbf{x} \in X.$$

Then, in the geometric sense, the $\{\mathfrak{s}_1, \ldots, \mathfrak{s}_J\}$ vertices construct an enclosing polytope for the $\mathfrak{C}$ set as Figure 4.1 shows: its elements are inside the polytope because they can be given as the convex combination of the vertices.

4.2 Affine geometry of polytopic form derivation problem

Although the considered Hilbert-space can be higher (occasionally infinite) dimensional, the polytopic description may be given by a finite number of vertices. It depends on the dimension of the so-called affine hull, that is, the affine subspace that contains every object and has minimal dimension. It can be easily expressed as the set of affine combinations of the $\mathfrak{c}(\mathbf{x})$ values

$$\mathfrak{A} = \left\{ \int_{\mathbf{x}\in X} \alpha(\mathbf{x})\mathfrak{c}(\mathbf{x})V(d\mathbf{x}) \middle| \int_{\mathbf{x}\in X} \alpha(\mathbf{x})V(d\mathbf{x}) = 1 \right\}. \tag{4.3}$$

The dimension of the affine hull is called affine dimension and denoted by D. Then the image set of $\mathfrak{c}(\mathbf{x})$ can be given as a sum of a value on a $(\mathfrak{a}_1, \ldots, \mathfrak{a}_D)$ basis and an $\mathfrak{a}_{D+1}$ offset, as

$$\mathfrak{c}(\mathbf{x}) = \sum_{d=1}^{D} u_d(\mathbf{x})\mathfrak{a}_d + \mathfrak{a}_{D+1} = \mathbf{v}(\mathbf{x}) \begin{bmatrix} \mathfrak{a}_1 \\ \vdots \\ \mathfrak{a}_{D+1} \end{bmatrix}, \tag{4.4}$$

where $\mathbf{v}(\mathbf{x}) = [\mathbf{u}(\mathbf{x}) \quad 1]$ is a homogeneous coordinate. With this description, the (occasionally infinite dimensional) objects are characterized by coordinates $\mathbf{u}(\mathbf{x}) = [u_1(\mathbf{x}), ..., u_D(\mathbf{x})]$ on the affine subspace.

The following subsections define an affine decomposition called Affine Singular Value Decomposition (ASVD) that provides a unique orthogonal basis for the affine hull and inherits the advantages of SVD.

4.2.1 Affine Singular Value Decomposition

First of all, we define the inner product and norm of $X \to H$ functions.

Definition 4.2 (Inner product and norm)**.** *For the inner product of $\mathfrak{b}, \mathfrak{c} : X \to H$ functions, we will use the following quantity*

$$< \mathfrak{b}, \mathfrak{c} >= \frac{1}{V(X)} \int_{\mathbf{x}\in X} < \mathfrak{b}(\mathbf{x}), \mathfrak{c}(\mathbf{x}) > V(d\mathbf{x}), \tag{4.5}$$

and for their norm: $||\mathfrak{c}|| = \sqrt{< \mathfrak{c}, \mathfrak{c} >}$.

In the following, the decomposition

$$\mathfrak{c}(\mathbf{x}) = \sum_{i=1}^{I} f_i(\mathbf{x})\mathfrak{c}_i$$

will be called

- orthonormal, if the weighting functions are orthonormal, namely, $< f_i, f_j >= \delta_{i,j}$ for all $i, j = 1..I$,
- homogeneous, if the last weighting function has constant one value, namely, $f_I(\mathbf{x}) = 1$ for all $\mathbf{x} \in X$.

The orthonormal decomposition will have particular significance because in this case, the computation of inner product and norm leads to computation of inner product of $\mathfrak{c}_i \in H$ values.

Lemma 4.3 (Inner product and norm of orthonormal decompositions)**.** *Given $\mathfrak{b}, \mathfrak{c} : X \to H$ functions with the same orthonormal weighting functions, their inner product can be computed as*

$$< \mathfrak{b}, \mathfrak{c} >= \sum_{i=1}^{I} < \mathfrak{b}_i, \mathfrak{c}_i >, \tag{4.6}$$

furthermore their norm as

$$||\mathfrak{c}|| = \sqrt{\sum_{i=1}^{I} ||\mathfrak{c}_i||^2}.$$

Now, we can define the following decomposition for the form in (4.4) such that the weighting functions are orthonormal and that represents the affine properties as well.

Definition 4.4 (Affine Singular Value Decomposition (ASVD))**.** *The affine form in* (4.4) *is called left-hand side affine SVD of $\mathfrak{c}$ function if it is a homogeneous, orthonormal decomposition, and the $\mathfrak{a}_i$ bases are orthogonal*

$$< \mathfrak{a}_i, \mathfrak{a}_j >= \delta_{ij}\sigma_i^2 \quad \forall i, j = 1, ..., D \tag{4.7}$$

and ordered as $\sigma_1 \geq \cdots \geq \sigma_D > 0$.

In the following, the abbreviation ASVD will refer to this Affine SVD instead of the so-called Adaptive SVD [157].

The form has an illustrative geometric interpretation: the vector $\mathfrak{a}_{D+1}$ is the same offset for all $\mathfrak{c}(\mathbf{x})$ value, the other $\mathfrak{a}_1, .., \mathfrak{a}_D$ objects give an D-dimensional ordered orthogonal basis on the affine subspace and the functions $v_1(\mathbf{x})$,...,$v_D(\mathbf{x})$ are coordinates for them, the values of which are dimensionless.

The decomposition's uniqueness property comes from the uniqueness of SVD trivially.

Proposition 4.5 (Uniqueness)**.** *The singular values $\sigma_1, ..., \sigma_D$ and the offset $\mathfrak{a}_{D+1}$ are unique.*

Considering the ordered singular values and denoting their multiplicities as $(m_1, m_2, ...)$ such that

$$\underbrace{\sigma_1 = \cdots = \sigma_{m_1}}_{m_1} > \underbrace{\sigma_{m_1+1} = \cdots = \sigma_{m_1+m_2}}_{m_2} > \ldots \sigma_D > 0,$$

the forms

$$\mathfrak{c}(\mathbf{x}) = \mathfrak{a}_{D+1} + \sum_{d=1}^{D} v'_d(\mathbf{x})\mathfrak{a}'_d \tag{4.8}$$

and only these forms are ASVD of $\mathfrak{c}(\mathbf{x})$*, where*

$$\begin{bmatrix} v_1'(\mathbf{x}) & \dots & v_D'(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} v_1(\mathbf{x}) & \dots & v_D(\mathbf{x}) \end{bmatrix} \mathbf{T}, \quad \begin{bmatrix} a_1'(\mathbf{x}) \\ \dots \\ a_D'(\mathbf{x}) \end{bmatrix} = \mathbf{T}^T \begin{bmatrix} a_1(\mathbf{x}) \\ \dots \\ a_D(\mathbf{x}) \end{bmatrix},$$
$$\mathbf{T} = \mathrm{blockdiag}(\mathbf{Q}_1, \mathbf{Q}_2, ...),$$

and $\mathbf{Q}_i$ *are arbitrary orthogonal matrices with size* $m_i \times m_i$*, respectively.*

This way, if all singular values are different, only the signs of $\mathfrak{a}_d$ objects, $v_d(\mathbf{x})$ functions ($d = 1, ..., D$) can be varied by considering matrices $\mathbf{T} = \mathrm{diag}(\pm 1, \pm 1, \dots)$ in Proposition 4.5.

The following proposition shows that the structure of ASVD allows for reducing the geometric dimension with minimal error in the defined norm.

Proposition 4.6 (Complexity trade-off)**.** *Consider the affine SVD* (4.4) *with* D *singular values, then* D *is the dimension of the affine hull.*

The best d *dimensional approximation with* $d < D$ *(in terms of the defined norm in Definition 4.2) can be obtained as*

$$\hat{\mathfrak{c}}(\mathbf{x}) = \mathfrak{a}_{D+1} + \sum_{l=1}^{d} v_l(\mathbf{x}) \mathfrak{a}_l. \tag{4.9}$$

4.2.2 Numerical reconstruction of Affine SVD

In this section, we will show, that the determination of ASVD can be performed by SVD computation. As the key of the method, consider the following Lemma.

Lemma 4.7 (ASVD from homogeneous, orthonormal decomposition.)**.** *Consider a function* $\mathbf{s} : X \to \mathbb{R}^{1\times R}$*, and suppose that it has the following, homogeneous, orthonormal decomposition*

$$\mathbf{s}(\mathbf{x}) = \sum_{m=1}^{M} f_m(\mathbf{x}) \mathbf{k}_m, \qquad \mathbf{k}_m \in \mathbb{R}^{1\times R} \tag{4.10}$$

which can be written in matrix form as $\mathbf{s}(\mathbf{x}) = \mathbf{f}(\mathbf{x})\mathbf{K}$*, where*

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & \dots & f_M(\mathbf{x}) \end{bmatrix}, \qquad \mathbf{K} = \begin{bmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_M \end{bmatrix}.$$

Then the ASVD of $\mathbf{s}(\mathbf{x})$*, denoted by* $\mathbf{s}(\mathbf{x}) = \mathbf{v}(\mathbf{x})\mathbf{F}$*, can be obtained as*

$$\mathbf{v}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \mathbf{S}\mathbf{V}^T \\ \mathbf{k}_M \end{bmatrix}, \tag{4.11}$$

where $\mathbf{U}, \mathbf{S}, \mathbf{V}$ *matrices come from SVD computation as*

$$\mathbf{USV}^T = \text{svd}\left(\begin{bmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_{M-1} \end{bmatrix}\right), \tag{4.12}$$

where the zero singular values and the corresponding columns of singular matrices are omitted.

Based on orthonormalization and the previous lemma, the following method can determine the ASVD or at least approximate it by applying discretization.

Algorithm 4.8 (Numerical reconstruction of ASVD)**.**

Step 1. *Consider a function* $\mathfrak{s} : X \to H$. *By describing its image on an orthonormal basis (denoted as* $\mathfrak{b}_1$*,...,* $\mathbf{b}_R$*), it is enough to consider the* $\mathbf{s} : X \to \mathbb{R}^R$ *function.*

Step 2 (Initial form)**.** *If the function is analytically given, a trivial initial decomposition can be constructed as* $\mathbf{s}(\mathbf{x}) = \mathbf{s}(\mathbf{x})\mathbf{E}$. *In this case, the decomposition will be exact.*

Otherwise, only an approximation

$$\hat{\mathbf{s}}(\mathbf{x}) = \boldsymbol{\alpha}(\mathbf{x})\mathbf{D} \qquad \boldsymbol{\alpha} : X \to \mathbb{R}^{1\times M}, \quad \mathbf{D} \in \mathbb{R}^{M\times R}$$

can be constructed by discretising the function on the X *domain in* $\{\mathbf{x}_1, \dots, \mathbf{x}_M\} \subset X$ *points. In this case, the m-th row of* $\mathbf{D}$ *contains the function value at* $\mathbf{x}_m$*:* $\mathbf{D}_m = \mathbf{s}(\mathbf{x}_m)$ *and the functions* $\boldsymbol{\alpha}(\mathbf{x}) = [\alpha_1(\mathbf{x}), \dots, \alpha_M(\mathbf{x})]$ *interpolate between these values.*

Step 3 (Homogeneous orthonormalization)**.** *Obtain* $\boldsymbol{\gamma} : X \to \mathbb{R}^L$ *homogeneous, orthonormal weighting functions, where* $<\gamma_i, \gamma_j>= \delta_{ij}$*,* $\gamma_L(\mathbf{x}) = 1$*, as* $\boldsymbol{\gamma}(\mathbf{x})\mathbf{T} = \boldsymbol{\alpha}(\mathbf{x})$*, so* $\hat{\mathbf{s}}(\mathbf{x})$ *can be written as*

$$\hat{\mathbf{s}}(\mathbf{x}) = \boldsymbol{\gamma}(\mathbf{x})\mathbf{C}, \tag{4.13}$$

where $\mathbf{C} = \mathbf{TD}$.

E.g.: Gram-Schmidt orthogonalization[27], the Householder transformation [39, 74] or the Givens rotation [24].

Step 4 (ASVD)**.** *By applying Proposition 4.7 to* (4.13)*, the ASVD of the approximation can be obtained*

$$\hat{\mathbf{s}}(\mathbf{x}) = \mathbf{v}(\mathbf{x})\mathbf{S}. \tag{4.14}$$

Then

$$\hat{\mathfrak{s}}(\mathbf{x}) = \mathfrak{a}_{D+1} + \sum_{d=1}^{D} v_d(\mathbf{x})\mathfrak{a}_d(\mathbf{x}) \quad \text{where} \quad \begin{bmatrix} \mathfrak{a}_1 \\ \vdots \end{bmatrix} = \mathbf{S} \cdot \begin{bmatrix} \mathfrak{b}_1 \\ \vdots \end{bmatrix}. \tag{4.15}$$

The algorithm can be used to derive the SVD based form, as well, by applying SVD in Step 4 and optionally simple orthogonalization in Step 3.

4.3 Polytopic form by determining enclosing polytope on the affine hull

Now having a compact, unique description on the affine hull, we can return to the problem of polytopic form derivation.

By constructing an enclosing polytope for the point set

$$\mathfrak{U} = \{\mathbf{u}(\mathbf{x}) | \mathbf{x} \in X\} \subset \mathbb{R}^D \tag{4.16}$$

on the D dimensional affine hull with vertices $\{\mathbf{r}_1, .., \mathbf{r}_J\}$ as

$$\mathfrak{U} \subseteq \mathrm{Co}(\mathbf{r}_1, .., \mathbf{r}_J), \tag{4.17}$$

the vertices

$$\mathfrak{s}_j = \begin{bmatrix} \mathbf{r}_j & 1 \end{bmatrix} \begin{bmatrix} \mathfrak{a}_1 \\ \vdots \\ \mathfrak{a}_{D+1} \end{bmatrix}, \tag{4.18}$$

contruct an enclosing polytope for $\mathfrak{c}(\mathbf{x})$, see Fig. 4.1. Furthermore, the $\mathbf{v}(\mathbf{x})$ homogeneous coordinates can be expressed as convex combinations of the vertices

$$\mathbf{v}(\mathbf{x}) = \mathbf{w}(\mathbf{x})\mathbf{R} \quad \text{where} \quad \mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & 1 \\ \vdots & \vdots \\ \mathbf{r}_J & 1 \end{bmatrix}, \tag{4.19}$$

$$w_k(\mathbf{x}) \geq 0, \ \sum_j w_j(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in X, \quad k = 1, ..., J,$$

and the $\mathfrak{c}(\mathbf{x})$ values can also be given as convex combinations of the $\mathfrak{s}_j$ vertices with the same $w_j(\mathbf{x})$ weighting functions, because

$$\mathfrak{c}(\mathbf{x}) = \mathbf{v}(\mathbf{x}) \begin{bmatrix} \mathfrak{a}_1 \\ \vdots \\ \mathfrak{a}_{D+1} \end{bmatrix} = \mathbf{w}(\mathbf{x})\mathbf{R} \begin{bmatrix} \mathfrak{a}_1 \\ \vdots \\ \mathfrak{a}_{D+1} \end{bmatrix} = \mathbf{w}(\mathbf{x}) \begin{bmatrix} \mathfrak{s}_1 \\ \vdots \\ \mathfrak{s}_J \end{bmatrix}. \tag{4.20}$$

This way, by considering the D dimensional geometric problem, polytopic description can be constructed for the original problem. Furthermore, the arrangement of the trajectory and the vertices is the same.

The last step of polytopic form generation is to find an appropriate enclosing polytope for the image set of $\mathbf{u}(\mathbf{x})$ coordinates (or $\mathbf{v}(\mathbf{x})$ homogeneous coordinates). For practical reasons, we must distinguish simplex and non-simplex ones.

4.3.1 Simplex enclosing polytopes

In the D dimensional space, the polytopes with non-zero volume and minimal number of vertices are the so-called simplex polytopes (line segment, triangle, tetrahedron, 4-simplex, etc.). The number of their vertices and facets are the same $J = F = D+1$, and the index lists of the facets are trivial.

For any simplex, the $\mathbf{u}$ points of the space can be uniquely given by affine combinations of its $\mathbf{r}_1, ..., \mathbf{r}_J$ vertices with $w_1, ..., w_J$ weights: because

$$\begin{bmatrix} w_1 & \dots & w_J \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_J \end{bmatrix} = \mathbf{u}, \quad \begin{bmatrix} w_1 & \dots & w_J \end{bmatrix} \mathbf{1}^J = 1, \tag{4.21}$$

the weights can be computed as

$$\begin{bmatrix} w_1 & \dots & w_J \end{bmatrix} = \begin{bmatrix} \mathbf{u} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & 1 \\ \vdots & \\ \mathbf{r}_J & 1 \end{bmatrix}^{-1}. \tag{4.22}$$

Based on the so determined weights the following cases can be distinguished

- If the weights are positive, the point is inside the simplex. Especially, if one or more weights are zero, it is on a facet.
- Otherwise, if there is a negative weight, the point is outside of the simplex.

This way, the enclosing property can be checked by computing the weighting functions as

$$\mathbf{w}(\mathbf{x}) = \mathbf{v}(\mathbf{x}) \begin{bmatrix} \mathbf{r}_1 & 1 \\ \vdots & \vdots \\ \mathbf{r}_J & 1 \end{bmatrix}^{-1}. \tag{4.23}$$

The volume as a geometric measure of the enclosing polytope can be computed as determinant of the matrix of vertices: $\mathrm{Vol} = \mathrm{abs}(\det(\mathbf{R}))/(D!)$ [156].

Numerical enclosing simplex generation methods within the TP model transformation are the so-called SNNN, CNO, IRNO methods, defined by properties of weighting functions [12, 17, 128, 180], based on the property above. The geometric interpretation of these methods are elaborated in Appendix A.

Furthermore, the problem of minimal volume enclosing simplex generation is deeply investigated in other fields of science such as signal processing that resulted in methods like MVSA, MVES, NMF-MVT, etc. see [42, 44, 172, 191]. Based on them, Section 6.3 will propose methods for Minimal Volume Enclosing Simplex Generation and Manipulation.

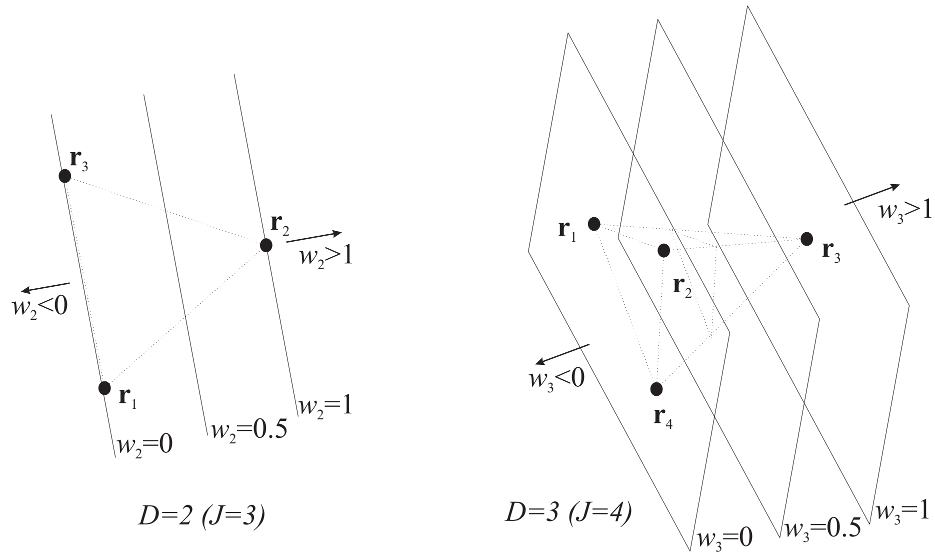


Figure 4.2: Illustration of unique weighting functions of simplex polytopic forms

4.3.2 Non-simplex enclosing polytopes

The polytopes with $J > D + 1$ vertices are called non-simplex polytopes.

It can be easily seen that the equation (4.21) is underdetermined in these cases, and the vertices do not define uniquely the $\mathbf{w}(\mathbf{x})$ weighting functions. If there exists at least one $\mathbf{w}(\mathbf{x})$ weighting system that denotes convex combinations, the polytope constructed from the vertices is enclosing.

Remark 4.9. *If the elements of a decomposition* (4.2) *constructs a non-simplex polytope, it can be enclosing even if the (actual) weighting functions do not define convex combinations.*

The number of topological questions grows with the D dimension, but the index lists and decompositions to simplices can be computed in higher dimensions as well with the tools of computational geometry like convex hull algorithms [9] and triangulation methods [87] (see, e.g., the QuickHull [20] and the Delaunay-triangulation [9] methods). The polytope is enclosing if each points are inside any of the simplices. This property can be easily verified. The volume can also be computed by summing the volumes of the simplices.

4.4 Summary

The chapter discovers the relevance of affine geometry and geometric enclosing polytope determination in the derivation of polytopic forms for multivariate functions. It introduced the Affine Singular Value Decomposition that uniquely shows the affine geometric properties and provides an opportunity for its reduction with the minimal numerical error.

4.5 Proofs

Proof of Lemma 4.3. *Based on Def. 4.2 the expression can be written as*

$$< \mathfrak{b}, \mathfrak{c} >= \frac{1}{V(X)} \int_{\mathbf{x}\in X} \langle \sum_{i=1}^{I} f_i(\mathbf{x})\mathfrak{b}_i, \sum_{i=1}^{I} f_i(\mathbf{x})\mathfrak{c}_i \rangle V(d\mathbf{x}) =$$

$$= \sum_{i=1}^{I}\sum_{j=1}^{I} \langle \mathfrak{b}_i, \mathfrak{c}_j \rangle \frac{1}{V(X)} \int_{\mathbf{x}\in X} f_i(\mathbf{x}) f_j(\mathbf{x}) V(d\mathbf{x}) = \sum_{i=1}^{I}\sum_{j=1}^{I} \langle \mathfrak{b}_i, \mathfrak{c}_j \rangle \delta_{ij} = \sum_{i=1}^{I} < \mathfrak{b}_i, \mathfrak{c}_i > .$$

Then the value of the norm is trivial. ■

Proof of Prop. 4.5. *This kind of decompositions are also ASVD, because*

– *The resulted weigthing functions are orthogonal: for all $i, j = 1, ..., D$*

$$< v'_i, v'_j >=< \left(\sum_{a=1}^{D} T_{ia} v_a\right), \left(\sum_{b=1}^{D} T_{jb} v_b\right) >= \sum_{a=1}^{D}\sum_{b=1}^{D} T_{ia}T_{jb} < v_a, v_b >=$$

$$= \sum_{a=1}^{D}\sum_{b=1}^{D} T_{ia}T_{jb}\delta_{ab} = \sum_{a=1}^{D} T_{ia}T_{ja} = (\mathbf{T}\mathbf{T}^T)_{ij} = \delta_{ij}.$$

and because $v'_{D+1}(\mathbf{x}) = v_{D+1}(\mathbf{x})$, for all $d = 1, ..., D$

$$< v'_d, v'_{D+1} >=< \left(\sum_{a=1}^{D} T_{da} v_a\right), v'_{D+1} >= \sum_{a=1}^{D} T_{da} < v_a, v'_{D+1} >=$$

$$= \sum_{a=1}^{D} T_{da}\delta_{a,(D+1)} = 0.$$

– *The $\mathfrak{a}'_d$ bases ($d = 1..D$) also keep their orthogonality, and the singular values do not change because for all $i, j = 1, ..., D$*

$$< \mathfrak{a}'_i, \mathfrak{a}'_j >=< \sum_{a=1}^{D} T_{ai}\mathfrak{a}_a, \sum_{b=1}^{D} T_{bj}\mathfrak{a}_b >= \sum_{a=1}^{D}\sum_{b=1}^{D} T_{ai}T_{bj} < \mathfrak{a}_a, \mathfrak{a}_b >=$$

$$= \sum_{a=1}^{D}\sum_{b=1}^{D} T_{ai}T_{bj}\delta_{ab}\sigma_a^2 = \sum_{d=1}^{D} T_{di}T_{dj}\sigma_d^2$$

and $T_{di} = 0$ if $\sigma_d \neq \sigma_i$, this way

$$< \mathfrak{a}'_i, \mathfrak{a}'_j >= \cdots = \sum_{d=1}^{D} T_{di}T_{dj}\sigma_d^2 = \sigma_i^2 \sum_{d=1}^{D} T_{di}T_{dj} = \sigma_i^2 (\mathbf{T}^T\mathbf{T})_{ij} = \sigma_i^2\delta_{ij}.$$

Only this kind of decompositions are ASVD, because

– *If* $v'_{D+1}(\mathbf{x}) = 1$ *and the* $v'_d(\mathbf{x})$ *functions are orthonormal, the offset part is unique:*

$$\int_{\mathbf{x}\in X} \mathfrak{c}(\mathbf{x})V(d\mathbf{x})/V(X) = \int_{\mathbf{x}\in X} \left(\mathfrak{a}_{D+1} + \sum_{d=1}^{D} v_d(\mathbf{x})\mathfrak{a}_d.\right) V(d\mathbf{x})/V(X) =$$
$$= \mathfrak{a}_{D+1} + \sum_{d=1}^{D} \mathfrak{a}_d \underbrace{\int_{\mathbf{x}\in X} v_d(\mathbf{x})V(d\mathbf{x})}_{=0} /V(X) = \mathfrak{a}_{D+1}$$

– *The remaining part must be SVD of function* $(\mathfrak{c}(\mathbf{x}) - \mathfrak{a}_{D+1})$ *inheriting its uniqueness properties, which results in the structure of* $\mathbf{T}$. ■

Proof of Prop. 4.6. *The mean value of* $\mathfrak{c}$ *function is*

$$\int_{\mathbf{x}\in X} \mathfrak{c}(\mathbf{x})V(d\mathbf{x})/V(X) = \int_{\mathbf{x}\in X} \left(\mathfrak{a}_{D+1} + \sum_{d=1}^{D} v_d(\mathbf{x})\mathfrak{a}_d.\right) V(d\mathbf{x})/V(X) =$$
$$= \mathfrak{a}_{D+1} + \sum_{d=1}^{D} \mathfrak{a}_d \int_{\mathbf{x}\in X} v_d(\mathbf{x})V(d\mathbf{x})/V(X) = \mathfrak{a}_{D+1}$$

such that $\mathfrak{a}_{D+1}$ *is the best* $d = 0$ *dimensional approximation.*

Moreover, if the best d dimensional approximation is known, the best $d+1$ *dimensional can be obtained by adding a product with the maximal possible norm (as in the Eckhart-Young theorem [57]).* ■

Proof of Lemma 4.7. *The weighting functions are orthonormal, because*

$$\frac{1}{V(X)} \int_{\mathbf{x}\in X} \mathbf{v}^T(\mathbf{x})\mathbf{v}(\mathbf{x})V(d\mathbf{x}) = \begin{bmatrix} \mathbf{U}^T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \frac{1}{V(X)} \int_{\mathbf{x}\in X} \mathbf{f}^T(\mathbf{x})\mathbf{f}(\mathbf{x})V(d\mathbf{x}) \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} =$$
$$= \begin{bmatrix} \mathbf{U}^T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} [< f_i, f_j >]_{ij} \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{U}^T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{E} \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{E}.$$

The orthogonality and the order of the $d = 1, ..., D$ *bases come from the properties of SVD: the rows of* $(\mathbf{SV}^T)$ *are orthogonal and ordered by norm.* ■

Chapter 5

New Polytopic TP form definition and Affine TP Model Transformation

The chapter extends the results of the previous chapter to derive complex, multi-polytopic structures ordered in a tensor product form. This Polytopic TP form is the relaxation of the original definition given by Baranyi [15, 19] to serve as a generalized polytopic form. Its main property is that it can contain separated variable/parameter dependencies, but it is not necessary: it can depend on arbitrarily chosen sets of parameters with arbitrary multiplicity. This way, the former Tensor Product model definition, where all of the parameter dependencies are separated, is a particular case of this definition as well as the commonly used polytopic form, where they are not separated at all.

The motivation of this extension is the fact that the parameter separation can increase the complexity of decompositions up to infinity in special cases. For example the $\sin(p_1 p_2)$ function cannot exactly be described as finite number of products of univariate functions. Furthermore, there usually emerge polytopic summations with higher multiplicities during construction of the LMI criteria for control analysis or synthesis. With the following extensions, the TP algebra can serve as a compact notation system and a framework for shaping these criteria as well.

For the derivation of Polytopic TP forms, the chapter generalizes the former results: it defines an intermediate TP form called Affine TP Form (because of its strong connection to affine geometry) based on the previously defined ASVD and gives a method to obtain it. This numerical reconstruction can be performed through discretization and interpolation on an equidistant grid, or the discretization points can be arbitrarily chosen. Furthermore, if the parameter separation can be done analytically, exact forms can be obtained with less computational cost. Then the Polytopic TP form can be derived by determining enclosing polytopes.

The chapter is structured as follows: First Section 5.1 discusses the new tensor algebra notations extended to elements of Hilbert spaces in general. Then Section 5.2 introduces the extended definition of polytopic TP form. Following that, Section 5.3 proposes the definition of Affine TP form to derive Polytopic TP forms. It shows its main properties, and it provides methods to obtain it for multivariate functions in general. After that, Section 5.4 concludes the results. Finally, Section 5.5 shortly details the proofs of the described lemmas and theorems.

5.1 Tensor algebra extension

In this section, some important concepts of tensor algebra are recalled based on [49], and they are extended for elements of a Hilbert space H in general, according to future use.

First, let us recall the definition of the k-mode subtensor.

Definition 5.1 (k-mode subtensor)**.** *The i-th k-mode subtensor of a tensor $\mathcal{A} \in H^{I_1 \times \cdots \times I_K}$, is a tensor on H with sizes $I_1 \times \cdots \times I_{k-1} \times 1 \times I_{k+1} \times \cdots \times I_K$ obtained by fixing its k-th index to i, and it is denoted as $\mathcal{A}_{i_k=i}$.*

The k-mode unfold matrix is constructed from the elements of the k-mode sub-tensors ordered into rows (see Fig. 5.1a) allowing for matrix operations (as SVD, QR, etc) along the k-th index and then for restoring the results to a tensor.

Definition 5.2 (k -mode unfold matrix of tensor)**.** *Assume a K order tensor $\mathcal{A} \in H^{I_1 \times \cdots \times I_K}$, where the elements can be described on an orthonormal basis with finite number R of elements. Then they can be written as a vector with R coordinates, and the tensor can be handled as a real tensor of size $I_1 \times \cdots \times I_K \times R$. Then its k-mode matrix unfolding*

$$\mathbf{A}_{(k)} \in \mathbb{R}^{I_k \times (R I_{k+1} \dots I_K I_1 \dots I_{k-1})}$$

contains the r-th coordinate of $\mathfrak{a}_{i_1,\dots,i_K}$ element at the position (i_k, j_k), where

$$j_k = \sum_{l=k+1}^{K} (i_l - 1) R \left(\prod_{m=l+1}^{K} I_m \right) \left(\prod_{m=1}^{k-1} I_m \right) + \sum_{l=1}^{k-1} (i_l - 1) R \prod_{m=l+1}^{k-1} I_m + r.$$

The k-mode tensor product with a real matrix combines the k-mode subtensors, see the following definition and Figure 5.1b.

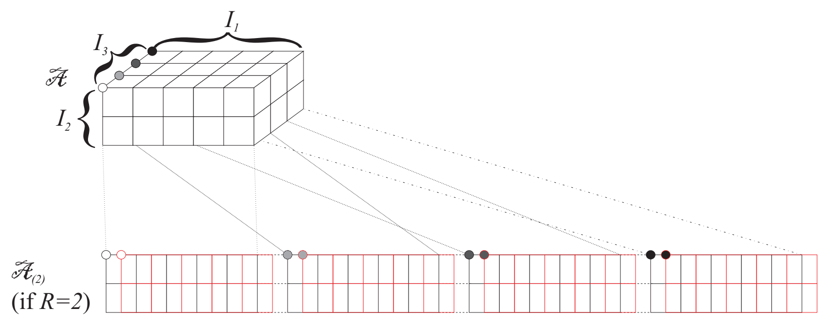
Definition 5.3 (k-mode tensor product)**.** *The k-mode product of a tensor $\mathcal{A} \in H^{I_1 \times \cdots \times I_K}$ and the matrix $\mathbf{U} \in \mathbb{R}^{L \times I_k}$, denoted by $\mathcal{A} \times_k \mathbf{U}$, is a tensor with size $I_1 \times \cdots \times I_{k-1} \times L \times I_{k+1} \times \cdots \times I_K$ of which are given by*

$$(\mathcal{A} \times_k \mathbf{U})_{i_1,\dots,i_{k-1},l,i_{k+1},\dots,i_K} = \sum_{i_n} \mathfrak{a}_{i_1,\dots,i_K} u_{l,i_k}.$$

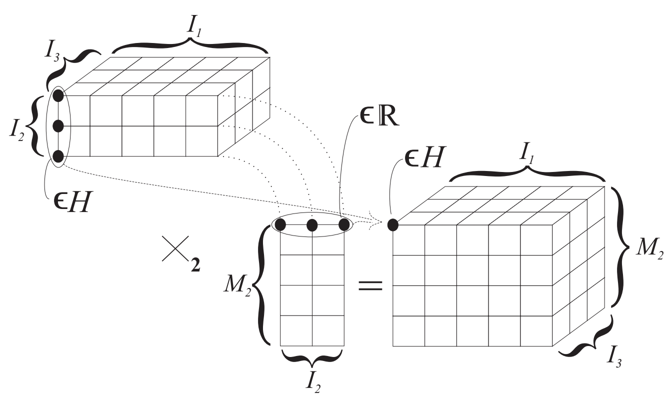
The definition implies the following properties.

Lemma 5.4 (Tensor product commutativity)**.** *Given a $\mathcal{A} \in H^{I_1 \times \cdots \times I_K}$ tensor and $\mathbf{U} \in \mathbb{R}^{J \times I_k}$, $\mathbf{V} \in \mathbb{R}^{N \times I_l}$ matrices , ($k \neq l$), one has*

$$(\mathcal{A} \times_k \mathbf{U}) \times_l \mathbf{V} = (\mathcal{A} \times_l \mathbf{V}) \times_k \mathbf{U}.$$



(a) Example for 2-mode tensor unfold ($I_1 = 6,\ I_2 = 3,\ I_3 = 4,$ $R = 2$)



(b) Example for 2-mode tensor product ($I_1 = 6$, $I_2 =$ $3,\ I_3 = 4,\ M_2 = 5$)

Figure 5.1: Illustrations of extended tensor operations

Lemma 5.5 (Multiple tensor products)**.** *Given the tensor* $\mathcal{A} \in H^{I_1\times\cdots\times I_K}$ *and the matrices* $\mathbf{U} \in \mathbb{R}^{J\times I_k}$*,* $\mathbf{V} \in \mathbb{R}^{M\times J}$*, one has*

$$(\mathcal{A} \times_k \mathbf{U}) \times_n \mathbf{V} = \mathcal{A} \times_k (\mathbf{V}\mathbf{U}).$$

The inner product and norm can be also defined.

Definition 5.6 (Inner product)**.** *The inner product* $< \mathcal{A}, \mathcal{B} >$ *of two tensors* $\mathcal{A}, \mathcal{B} \in H^{I_1\times\cdots\times I_K}$ *is defined as*

$$< \mathcal{A}, \mathcal{B} >= \sum_{i_1} \cdots \sum_{i_K} < \mathfrak{a}_{i_1,\ldots,i_K}, \mathfrak{b}_{i_1,\ldots,i_K} > .$$

Definition 5.7 (Norm)**.** *The Frobenius-norm of a tensor* $\mathcal{A}$ *is given by* $||\mathcal{A}|| = \sqrt{< \mathcal{A}, \mathcal{A} >}$*.*

5.2 Definition of relaxed Polytopic TP form

This section extends the former Polytopic TP form definition (often referred to as Convex TP form, see [15, 19]). Consider the multivariate function

$$\mathfrak{g} : \Omega \to H, \qquad \text{where } \Omega = \left[\underline{p}_1, \overline{p}_1\right] \times \cdots \times \left[\underline{p}_N, \overline{p}_N\right] \subset \mathbb{R}^N, \tag{5.1}$$

and the $k = 1, ..., K$ sets of the parameters, where the indices of parameters in the kth set are denoted with $i_1^{(k)}, i_2^{(k)}, ... \in \{1, .., N\}$ as

$$\mathbf{p}^{(k)} = \begin{bmatrix} p_{i_1^{(k)}} & p_{i_2^{(k)}} & \cdots \end{bmatrix} = \begin{bmatrix} p_1^{(k)} & p_2^{(k)} & \ldots \end{bmatrix}.$$

The parameter domain, which belongs to the k-th parameter domain, is denoted as

$$\Omega_k = [\underline{p}_1^{(k)}, \overline{p}_1^{(k)}] \times [\underline{p}_2^{(k)}, \overline{p}_2^{(k)}] \times \ldots .$$

With these notations, the new, generalized Polytopic TP form can be defined, which is only separated to these parameter sets.

Definition 5.8 (Polytopic TP form)**.** *The following form of function of* (5.1)

$$\mathfrak{g}(\mathbf{p}) = \mathcal{G} \mathop{\boxtimes}_{k=1}^{K} \mathbf{w}^{(k)}(\mathbf{p}^{(k)}) \tag{5.2}$$

is called Polytopic TP form, if the core tensor $\mathcal{G}$ is a tensor on H with sizes with $J_1 \times \cdots \times J_K$, and the $\mathbf{w}^{(k)} : \Omega_k \to \mathbb{R}^{J_K}$ functions are called weighting functions such that

$$\mathfrak{g}(\mathbf{p}) = \sum_{j=1}^{J_k} \left(\mathcal{G}_{j_k=j} \mathop{\boxtimes}_{l=1, l \neq k}^{K} \mathbf{w}^{(l)}(\mathbf{p}^{(l)}) \right) w_j^{(k)}(\mathbf{p}^{(k)}) \tag{5.3}$$

is polytopic description for all $k = 1, ..., K$ modes.

This way, the original TP form recalled in Chapter 3, which is separated to univariate parameter dependencies, and the form, where the parameter dependencies are not separated at all are special cases of this definition, which shows its generality.

The definition does not limit the parameter sets to be disjoint. This way, it can even depend on the same parameter set with the same weighting functions more times, that results in multiple polytopic summations which usually appears in polytopic model-based control. The Lyapunov-function and controller candidate can be on a polytopic structure with higher multiplicity, the advantages of which will be shown in Chapter 7. For this reason, we introduce the multiplicity of dependencies and a compact notation system for it.

Notation 5.9. *If a Polytopic TP form depends on $\mathbf{w}^{(l)}(\mathbf{p}^{(l)})$ $l = 1, ..., L$ weighting* *functions with $M_1, ..., M_L$ multiplicities, it can be written as*

$$\mathfrak{g}(\mathbf{p}) = \mathcal{G} \underbrace{\times_1 \mathbf{w}^{(1)}(\mathbf{p}^{(1)}) \times_2 \mathbf{w}^{(1)}(\mathbf{p}^{(1)}) \cdots \times_{M_1} \mathbf{w}^{(1)}(\mathbf{p}^{(1)})}_{M_1}$$
$$\underbrace{\times_{M_1+1} \mathbf{w}^{(2)}(\mathbf{p}^{(2)}) \cdots \times_{M_1+M_2} \mathbf{w}^{(2)}(\mathbf{p}^{(2)})}_{M_2} \times_{M_1+M_2+1} \cdots =$$
$$= \mathcal{G} \mathop{\boxtimes}_{m=1}^{M_1} \mathbf{w}^{(1)}(\mathbf{p}^{(1)}) \mathop{\boxtimes}_{m=M_1+1}^{M_1+M_2} \mathbf{w}^{(2)}(\mathbf{p}^{(2)}) \ldots, \quad (5.4)$$

and by introducing the following functions

$$K(\mathbf{M}) = \sum_i M_i, \quad (5.5)$$

$$l(k, \mathbf{M}) = i \quad \text{where} \quad \sum_{a=1}^{i-1} M_a < k \leq \sum_{a=1}^{i} M_a, \quad (5.6)$$

it can be written in the form

$$\mathfrak{g}(\mathbf{p}) = \mathcal{G} \mathop{\boxtimes}_{k=1}^{K(\mathbf{M})} \mathbf{w}^{(l(k,\mathbf{M}))}(\mathbf{p}^{(l(k,\mathbf{M}))}). \quad (5.7)$$

The new form allows for performing many operations like sum, difference, product, etc. and resulting in a polytopic TP form with the same weighting functions but only the elements of the tensor and its multiplicities different.

5.2.1 Affine operators on Polytopic TP forms

Here we discuss the applied affine operators, and we show that their results can be written into Polytopic TP form on the original polytopic structures by leading back the operations to the elements of the core tensor. These tools will be relevant in the next chapters by allowing that control-related definite criteria can be handled as definite conditions on TP forms.

Consider $\mathfrak{f}(\mathbf{p})$ and $\mathfrak{g}(\mathbf{p})$ polytopic TP forms on a Hilbert space H in general and denote their weighting functions as $\mathbf{w}^{(1)}(\mathbf{p}^{(1)}), \mathbf{w}^{(2)}(\mathbf{p}^{(2)}), \ldots$ and multiplicities $\mathbf{f}, \mathbf{g} \in \mathbb{N}^L$, respectively, so

$$\mathfrak{f}(\mathbf{p}) = \mathcal{F} \mathop{\boxtimes}_{k=1}^{K(\mathbf{f})} \mathbf{w}^{(l(k,\mathbf{f}))}(\mathbf{p}^{(l(k,\mathbf{f}))}), \quad (5.8)$$

$$\mathfrak{g}(\mathbf{p}) = \mathcal{G} \mathop{\boxtimes}_{k=1}^{K(\mathbf{g})} \mathbf{w}^{(l(k,\mathbf{g}))}(\mathbf{p}^{(l(k,\mathbf{g}))}). \quad (5.9)$$

Lemma 5.10. *The univariate linear operators denoted as* $\mathrm{op} : H \to H$ *on polytopic*

TP form (5.9) *can be written into a TP form as*

$$\mathfrak{h}(\mathbf{p}) \equiv \mathrm{op}(\mathfrak{g}(\mathbf{p})) = \mathcal{H} \mathop{\boxtimes}_{k=1}^{K(\mathbf{g})} \mathbf{w}^{(l(k,\mathbf{g}))}(\mathbf{p}^{(l(k,\mathbf{g}))}), \tag{5.10}$$

where the multiplicities are the same and for elements of core tensors:

$$\mathfrak{h}_{j_{1,1},\dots,j_{L,g_L}} \equiv \mathrm{op}(\mathfrak{g}_{j_{1,1},\dots,j_{L,g_L}}). \tag{5.11}$$

For example, if the considered Hilbert space is the space of square matrices with given sizes, operations like transpose, trace, matrix multiplication, etc. are applicable.

Lemma 5.11. *Sum, difference and vectorization of polytopic TP forms* (5.8) *and* (5.9) *can be written into a TP form as*

$$\mathfrak{h}(\mathbf{p}) = \mathcal{H} \mathop{\boxtimes}_{k=1}^{K(\mathbf{h})} \mathbf{w}^{(l(k,\mathbf{h}))}(\mathbf{p}^{(l(k,\mathbf{h}))}), \tag{5.12}$$

where the multiplicities are $h_k = \max(f_k, g_k)$ *and by denoting the*

$$(j_{1,1}, j_{1,2}, \dots, j_{1,h_1}, j_{2,1}, \dots, j_{2,h_2}, j_{3,1}, \dots)\text{−th} \tag{5.13}$$

element of the $\mathcal{H}$ *core tensor with* $\mathfrak{h}$, *which can be computed from the*

$$(j_{1,1}, j_{1,2}, \dots, j_{1,f_1}, j_{2,1}, \dots, j_{2,f_2}, j_{3,1}, \dots)\text{−th} \tag{5.14}$$

element of $\mathcal{F}$ *denoted by* $\mathfrak{f}$ *and the*

$$(j_{1,1}, j_{1,2}, \dots, j_{1,g_1}, j_{2,1}, \dots, j_{2,g_2}, j_{3,1}, \dots)\text{−th} \tag{5.15}$$

element of $\mathcal{G}$ *denoted by* $\mathfrak{g}$

- *for addition* $(\mathfrak{h}(\mathbf{p}) = \mathfrak{f}(\mathbf{p}) + \mathfrak{g}(\mathbf{p}))$ *as* $\mathfrak{h} \equiv \mathfrak{f} + \mathfrak{g}$,
- *for subtraction* $(\mathfrak{h}(\mathbf{p}) = \mathfrak{f}(\mathbf{p}) - \mathfrak{g}(\mathbf{p}))$ *as* $\mathfrak{h} \equiv \mathfrak{f} - \mathfrak{g}$,
- *for vectorization* $(\mathfrak{h}(\mathbf{p}) = \begin{bmatrix}\mathfrak{f}(\mathbf{p}) & \mathfrak{g}(\mathbf{p})\end{bmatrix})$ *as* $\mathfrak{h} \equiv \begin{bmatrix}\mathfrak{f} & \mathfrak{g}\end{bmatrix}$.

Lemma 5.12. *Multiplication of polytopic TP forms (5.8-5.9) can be written into a TP form as*

$$\mathfrak{f}(\mathbf{p}) \cdot \mathfrak{g}(\mathbf{p}) = \mathfrak{h}(\mathbf{p}) = \mathcal{H} \mathop{\boxtimes}_{k=1}^{K(\mathbf{h})} \mathbf{w}^{(l(k,\mathbf{h}))}(\mathbf{p}^{(l(k,\mathbf{h}))}), \tag{5.16}$$

where the multiplicities are $h_k = f_k + g_k$ *and by denoting the*

$$(j_{1,1}, j_{1,2}, \dots, j_{1,f_1}, i_{1,1}, i_{1,2}, \dots, i_{1,g_1}, j_{2,1}, \dots, j_{2,f_2}, i_{2,1}, \dots, i_{2,g_2}, j_{3,1}, \dots)\text{−th}$$

element of the $\mathcal{H}$ *by* $\mathfrak{h}$, *which can be computed from the*

$$(j_{1,1}, j_{1,2}, \dots, j_{1,f_1}, j_{2,1}, \dots, j_{2,f_2}, j_{3,1}, \dots)\text{−th} \tag{5.17}$$

element of $\mathcal{F}$ with $\mathfrak{f}$, the

$$(i_{1,1}, i_{1,2}, \ldots, i_{1,g_1}, i_{2,1}, \ldots, i_{2,g_2}, i_{3,1}, \ldots)\text{−th} \tag{5.18}$$

element of $\mathcal{G}$ with $\mathfrak{g}$ as $\mathfrak{h} \equiv \mathfrak{f} \cdot \mathfrak{g}$.

These methods will be relevant in Chapter 7 for deriving and interpreting LMIs in polytopic TP forms: The conditions can generally be written into definite conditions on Polytopic TP forms.

5.3 Derivation of Polytopic TP forms for multivariate functions

This section introduces the Affine TP Model Transformation as a method to obtain Polytopic TP forms for a multivariate function. Here we assume that the demanded parameter sets are given, their practical role during controller design will be clarified in Chapter 7.

5.3.1 Definition and properties of Affine TP form

Motivated by the previous chapter, here we define a multi-ASVD form called Affine TP form as an intermediate stage to derive the desired multi-polytopic one.

Definition 5.13 (Affine TP form)**.** *Consider the following form of function of* (5.1) *with the desired parameter sets $\mathbf{p}^{(k)}$*

$$\mathfrak{g}(\mathbf{p}) = \mathcal{G}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}), \tag{5.19}$$

in which the $\mathcal{G}^{aff}$ core tensor is on H as $\mathcal{G}^{aff} \in H^{(D_1+1)\times\cdots\times(D_K+1)}$, and the D_k ($k = 1, ..., K$) values are called k-mode dimensions. Its expansion

$$\mathfrak{g}(\mathbf{p}) = \sum_{d=1}^{D_k+1} \left(\mathcal{G}^{aff}_{d_k=d} \mathop{\boxtimes}_{l=1,l\neq k}^{K} \mathbf{v}^{(l)}(\mathbf{p}^{(l)}) \right) v_d^{(k)}(\mathbf{p}^{(k)}) \tag{5.20}$$

is the ASVD with $\sigma_1^{(k)}, ..., \sigma_{D_k}^{(k)}$ singular values for all k.

Then the Polytopic form (5.2) can be obtained by determining enclosing polytopes for the images of $\mathbf{v}^{(k)}(\mathbf{p}^{(k)})$ functions in the belonging D_k dimensional spaces for all $k = 1, ..., K$, see the next theorem.

Theorem 5.14 (Derivation of Polytopic TP form). *If for all $k = 1, ..., K$ the $\mathbf{r}_1^{(k)}, ..., \mathbf{r}_{J_k}^{(k)}$ vertices construct an enclosing polytope for image of $\mathbf{v}^{(k)}(\mathbf{p}^{(k)})$, it can be expressed as $\mathbf{v}^{(k)}(\mathbf{p}^{(k)}) = \mathbf{w}^{(k)}(\mathbf{p}^{(k)})\mathbf{R}^{(k)}$ where*

$$\mathbf{R}^{(k)} = \begin{bmatrix} \mathbf{r}_1^{(k)} & 1 \\ \vdots & \vdots \\ \mathbf{r}_{J_k}^{(k)} & 1 \end{bmatrix}. \tag{5.21}$$

Then

$$\mathfrak{g}(\mathbf{p}) = \mathcal{G}^{aff} \mathop{\boxtimes}_{k=1}^{K} \left(\mathbf{w}^{(k)}(\mathbf{p}^{(k)})\mathbf{R}^{(k)}\right) = \underbrace{\left(\mathcal{G}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{R}^{(k)}\right)}_{\mathcal{G}^{poly}} \mathop{\boxtimes}_{k=1}^{K} \mathbf{w}^{(k)}(\mathbf{p}^{(k)}) \tag{5.22}$$

which is the Polytopic TP form.

The uniqueness of the Affine TP form can be formalized for cases where the parameter sets are disjoint, see the following theorem.

Theorem 5.15 (Uniqueness). *Consider an Affine TP model, where the parameter sets are disjoint.*

Then the $\sigma_d^{(k)}$ singular values are unique. Denote their multiplicities by $(m_1^{(k)}, m_2^{(k)}, \dots)$ as in Proposition 4.5 for all $k = 1, ..., K$.

If (5.19) is an Affine TP form, the following and only the following forms are also Affine TP models

$$\mathfrak{g}(\mathbf{p}) = \left(\mathcal{G}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{T}^{(k)}\right) \mathop{\boxtimes}_{k=1}^{K} \left(\mathbf{v}^{(k)}(\mathbf{p}^{(k)})\mathbf{T}^{(k)T}\right), \tag{5.23}$$

where the matrices are defined as $\mathbf{T}^{(k)} = \mathrm{diag}(\mathbf{T}_0^{(k)}, 1)$ and $\mathbf{T}_0^{(k)}$ is a blockdiagonal matrix constructed by arbitrary orthogonal marices with sizes $m_1^{(k)} \times m_1^{(k)}$, $m_2^{(k)} \times m_2^{(k)}$ etc. as in Proposition 4.5.

5.3.2 Derivation of Affine TP form

The Affine TP form with the given parameter sets can be numerically reconstructed via the following algorithm.

Algorithm 5.16. ***(Numerical reconstruction of Affine TP form)***

The first step is to obtain an initial TP form with the desired parameter sets. Here we describe two approaches for it.

Step 1a. ***(Analytical initial form)*** *If the function is analytically given, the initial form with the desired parameter separation may be constructed analytically, which is* *denoted as*

$$\hat{\mathfrak{g}}(\mathbf{p}) = \mathcal{D} \mathop{\boxtimes}_{k=1}^{K} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}). \tag{5.24}$$

Step 1b. ***(Discretisation based initial form)***

The function can be approximated by a TP form via discretisation in general: For each $\mathbf{p}^{(k)}$ *parameter sets, choose* M_k *discrete points denoted as* $\{..., \mathbf{g}^{(k)}_{m_k}, ..\}$ *and appropriate* $\boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)})$ *by interpolation (as Lagrange-polynomials, piecewise linear/constant functions, etc.), see Fig. 5.2.*

Then the initial TP form

$$\hat{\mathfrak{g}}(\mathbf{p}) = \mathcal{D} \mathop{\boxtimes}_{k=1}^{K} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}) = \sum_{m_1=1}^{M_1} \cdots \sum_{m_K=1}^{M_K} \mathfrak{d}_{m_1,\dots,m_K} \prod_{k=1}^{K} \alpha^{(k)}_{m_k}(\mathbf{p}^{(k)}) \tag{5.25}$$

can be constructed to approximate the function by choosing elements of the core tensor $\mathcal{D} \in H^{M_1 \times M_2 \times \cdots \times M_K}$ *denoted by the* $\mathfrak{d}_{m_1,\dots,m_K}$ *as the values of* $\mathfrak{g}(\mathbf{p})$ *function at* $(\mathbf{g}^{(1)}_{m_1}, \mathbf{g}^{(2)}_{m_2}, ...)$*.*

Step 2. ***(Homogeneous orthonormalization)*** *Determine the* $\boldsymbol{\gamma}^{(k)} : \Omega_k \to \mathbb{R}^{L_k}$ *homogeneous, orthonormal weighting functions as* $\boldsymbol{\gamma}^{(k)}(\mathbf{p}^{(k)})\mathbf{T}^{(k)} = \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)})$*, where* $\gamma^{(k)}_{L_k}(\mathbf{p}^{(k)}) = 1$*, to obtain the following orthonormal TP form*

$$\hat{\mathfrak{g}}(\mathbf{p}) = \mathcal{F} \mathop{\boxtimes}_{k=1}^{K} \boldsymbol{\gamma}^{(k)}(\mathbf{p}^{(k)}),$$

where

$$\mathcal{F} = \mathcal{D} \mathop{\boxtimes}_{k=1}^{K} \mathbf{T}^{(k)}.$$

E.g.: Gram-Schmidt orthogonalization[27], the Householder transformation[74, 39] or the Givens rotation[24].

Step 3. ***(Sequential ASVD)*** *Denote the TP form as*

$$\hat{\mathfrak{g}}(\mathbf{p}) = \mathcal{K} \mathop{\boxtimes}_{k=1}^{K} \mathbf{f}^{(k)}(\mathbf{p}^{(k)}), \tag{5.26}$$

which has initial value $\mathcal{K} = \mathcal{F}$ *and* $\mathbf{f}^{(k)}(\mathbf{p}^{(k)}) = \boldsymbol{\gamma}^{(k)}(\mathbf{p}^{(k)})$ *for* $k = 1, ..., K$*.*

Then for $k = 1$*, compute the ASVD of* $\mathbf{f}^{(k)}(\mathbf{p}^{(k)})\mathbf{K}_{(k)}$ *form as*

$$\mathbf{f}^{(k)}(\mathbf{p}^{(k)})\mathbf{K}_{(k)} = \mathbf{v}^{(k)}(\mathbf{p}^{(k)})\mathbf{K}'$$

(see Lemma 4.7), and restore tensor $\mathcal{K}$ *from the unfolded matrix* $\mathbf{K}'$*,* $\mathbf{f}^{(k)}(\mathbf{p}^{(k)}) := \mathbf{v}^{(k)}(\mathbf{p}^{(k)})$ *and* $k \leftarrow k+1$ *until* $k \le K$*.*

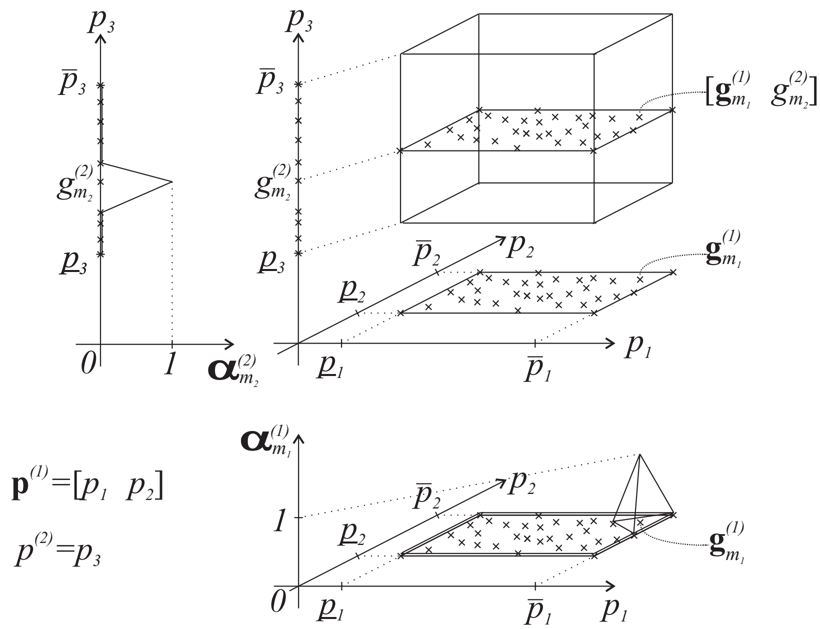


Figure 5.2: Illustration of discretisation using piecewise linear interpolatory functions

The resulting form is affine, with the introduced notations

$$\hat{\mathfrak{g}}(\mathbf{p}) = \mathcal{G}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}). \tag{5.27}$$

and it approximates the original $\mathfrak{c}(\mathbf{p})$ *function as the TP form in* (5.24)*.*

It is easy to see that this method extends the previous approach [136, 155, 160] by allowing exact analytical separation or the application of discretisation with varying density along the Ω parameter domain with different interpolation strategies (e.g. linear, polynomial, trigonometric). see [155].

The sequential truncation approach (see [136]) can also be applied by using the complexity reductions in iterations of Step 3 to decrease the computational cost.

Furthermore, by applying SVD instead of ASVD in Step 3 (and optionally simple orthonormalization in Step 2), the algorithm can be used to determine the so-called HOSVD based form, as well.

5.3.3 Complexity reduction

The Affine TP form also allows reducing the k-mode dimension with the following error in the norm given in Definition 4.2.

Theorem 5.17 (Complexity reduction)**.** *The k-mode dimension from D_k to $D'_k < D_k$ can be reduced by disregarding the $(D'_k+1),...,D_k$-th subtensors of $\mathcal{G}^{aff}$ and the* *corresponding elements of* $\mathbf{v}^{(k)}(\mathbf{p}^{(k)})$*. The resulting description is an approximating, Affine TP form and the error is*

$$||\mathfrak{g} - \hat{\mathfrak{g}}||^2 = \sum_{d=D'_k+1}^{D_k} \sigma_d^{(k)2}, \tag{5.28}$$

and it is minimal if the parameter sets are disjoint.

By reducing the k-mode dimensions for $k \in \mathfrak{K} \subseteq \{1, ..., K\}$ *from* D_k *to* $D'_k \leq D_k$*, and performing Step 3 of Algorithm 5.16, the resulting description is an approximating, Affine TP form and the error can be bounded as*

$$||\mathfrak{g} - \hat{\mathfrak{g}}||^2 \leq \sum_{k \in \mathfrak{K}} \sum_{d=D'_k+1}^{D_k} \sigma_d^{(k)2}, \tag{5.29}$$

and furthermore, if the parameter sets are disjoint, it is minimal.

In practice, complexity reduction can be applied to cut off the

5.3.4 Complexity reduction preserving exactness

If Algorithm 5.16 resulted in too complex structures with D_k dimensions up to infinity, which can be handled only via complexity reduction resulting in an approximating description, the following method is capable to transform the discrepancy of the approximating description into a new parameter dependency, resulting in a finite, exact Affine TP form, where "exact" means that the functions describe the function accurately.

Algorithm 5.18 (Complexity reduction preserving exactness)**.**

Step 1. *Consider the multivariate function* (5.1) *with given parameter sets and assume that Algorithm 5.16 with complexity reduction results in an approximating TP form*

$$\hat{\mathfrak{g}}(\mathbf{p}) = \mathcal{G}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}). \tag{5.30}$$

Step 2. *Denote the error function of the description as*

$$\mathfrak{e}(\mathbf{p}) = \mathfrak{g}(\mathbf{p}) - \hat{\mathfrak{g}}(\mathbf{p}) \tag{5.31}$$

and obtain its affine TP form without parameter separation by Algorithm 5.16 as

$$\mathfrak{e}(\mathbf{p}) = \mathcal{E}^{aff} \times_1 \mathbf{v}^{(K+1)}(\mathbf{p}). \tag{5.32}$$

Step 3. *Then construct the following TP form*

$$\mathfrak{g}(\mathbf{p}) = \mathcal{G}^{aff2} \mathop{\boxtimes}_{k=1}^{K+1} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}), \tag{5.33}$$

where

$$\begin{aligned}
\mathfrak{g}^{aff2}_{i_1,...,i_{K+1}} &= h(i_{K+1}) \cdot \mathfrak{g}^{aff}_{i_1,...,i_K} + f(i_1,...,i_K) \cdot \mathfrak{e}^{aff}_{i_{K+1}}, \\
f(i_1,...,i_K) &= \begin{cases} 1 & if \ i_k = D_k + 1 \ \forall k = 1,...,K, \\ 0 & otherwise, \end{cases} \\
h(i_{K+1}) &= \begin{cases} 1 & if \ i_{K+1} = D_{K+1} + 1, \\ 0 & otherwise, \end{cases} \\
\mathbf{p}^{(K+1)} &= \mathbf{p},
\end{aligned}$$

and $\mathfrak{g}^{aff2}_{i_1,...,i_{K+1}}$, $\mathfrak{g}^{aff}_{i_1,...,i_K}$, $\mathfrak{e}^{aff}_{i_{K+1}}$ *refer to the elements of tensors* $\mathcal{G}^{aff2}$, $\mathcal{G}^{aff}$ *and* $\mathcal{E}$, *respectively.*

Now the parameter sets are not disjoint, and it is easy to see the lack of uniqueness: different complexity reductions lead to different, but exact Affine TP forms.

5.3.5 Remarks on numerical reconstruction

Reduce computational cost of Algorithm 5.16

Consider Step 2 of Algorithm 5.16 and the orthonormalization operations where scalar products of weighting functions must be computed $M_k(M_k + 1)/2$ times, because the interpolatory functions are linearly independent. The scalar product of functions was defined by Def. 4.2 as a multivariate integral on the $\mathbf{p}^{(k)}$ parameter set that causes a large computational burden. The following lemma allows for reducing this burden by decreasing the size of the TP form before the orthonormalization.

Lemma 5.19 (Reducing the l-mode size based on the core tensor)**.** *Consider the following TP form*

$$\mathfrak{g}(\mathbf{p}) = \mathcal{G} \mathop{\boxtimes}_{k=1}^{K} \boldsymbol{\gamma}^{(k)}(\mathbf{p}^{(k)}), \tag{5.34}$$

where the l-mode size is denoted by M_l.

If the core tensor $\mathcal{G}$ is not full l-mode rank, its size can be decreased by performing QR (or SVD, etc.) factorization on $\mathbf{G}_{(k)}$ as

$$[\mathbf{Q}, \mathbf{R}] = \mathrm{QR}(\mathbf{G}_{(k)}), \tag{5.35}$$

where the zero rows of matrix $\mathbf{R}$ and corresponding columns of $\mathbf{Q}$ are omitted. Then

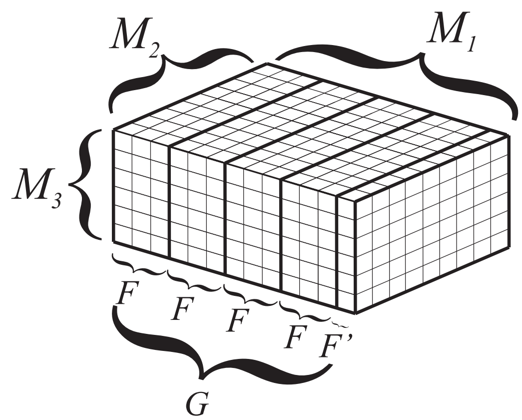


Figure 5.3: Illustration of partitions if $K = 3$, $M_1 = 13$, $M_2 = 9$, $M_3 = 6$, $F = 3$, $F' = 1$, $G = 5$

by restoring a tensor $\mathcal{R}$ from matrix $\mathbf{R}$, the function can be written as

$$\mathfrak{g}(\mathbf{p}) = \mathcal{R} \underset{k=1,k\neq l}{\overset{K}{\boxtimes}} \boldsymbol{\gamma}^{(k)}(\mathbf{p}^{(k)}) \times_l \boldsymbol{\gamma}'^{(l)}(\mathbf{p}^{(l)}), \tag{5.36}$$

where $\boldsymbol{\gamma}'^{(l)}(\mathbf{p}^{(l)}) = \boldsymbol{\gamma}^{(l)}(\mathbf{p}^{(l)})\mathbf{Q}$ has $I_l < M_l$ number of elements.

Reduce memory need of Algorithm 5.16

Consider a function with N parameters and assume that they are collected into K parameter sets. The k-th parameter set includes the parameters with indices $n_1^{(k)}, n_2^{(k)}, ...$ and it needs to be sampled in M_k points.

So the discretised TP form would be

$$\mathfrak{f}(\mathbf{p}) = \mathcal{D} \overset{K}{\underset{k=1}{\boxtimes}} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}), \quad \mathcal{D} \in H^{M_1 \times \cdots \times M_K}, \quad \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}) : \Omega_k \to \mathbb{R}^{M_k}. \tag{5.37}$$

Simple case. Assume that the discretised model would be too large for the available memory, and furthermore, the expected maximal 1-mode size of the Affine TP form is I_1 and two TP forms with sizes $F \times M_2 \times M_3 \times \cdots \times M_K$ and $(I_1 + F) \times M_2 \times M_3 \times \cdots \times M_K$ can be stored where $F \geq I_1$.

Now partitionate the $M_1 \times \cdots \times M_K$ tensor of points to be discretised into ones with sizes $F \times M_2 \times M_3 \times \cdots \times M_K$, denote the 1-mode size of the remaining part with F', and denote the number of partitions with G, see Fig. 5.3.

This way, we can consider G discretisation tasks as

$$\mathfrak{f}(\mathbf{p}) = \sum_{g=1}^{G} \mathfrak{f}^{(g)}(\mathbf{p}), \tag{5.38}$$

where

$$\begin{aligned} &\mathfrak{f}^{(g)}(\mathbf{p}) = \mathcal{D}^{(g)} \times_1 \boldsymbol{\alpha}^{(1,g)}(\mathbf{p}^{(1)}) \mathop{\boxtimes}_{k=2}^{K} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}), \\ &\mathcal{D}^{(g)}_{f,l_2,\ldots,l_K} = \mathcal{D}_{l_1,l_2,\ldots,l_K} \quad \text{where } l_1 = f + F(g-1), \\ &\boldsymbol{\alpha}^{(1,g)}(\mathbf{p}^{(1)}) = \begin{bmatrix} \alpha^{(1)}_{(g-1)F+1}(\mathbf{p}^{(1)}) & \ldots & \alpha^{(1)}_{\min(gF,M_1)}(\mathbf{p}^{(1)}) \end{bmatrix} \quad \forall g = 1, ..., G. \end{aligned} \tag{5.39}$$

It is easy to see that the computation can be led back to a summation as

$$\mathfrak{s}^{(g')}(\mathbf{p}) = \sum_{g=1}^{g'} \mathfrak{f}^{(g)}(\mathbf{p}) = \mathfrak{s}^{(g'-1)}(\mathbf{p}) + \mathfrak{f}^{(g')}(\mathbf{p}). \tag{5.40}$$

For its computation, see the following method illustrated in Fig. 5.4.

Algorithm 5.20. *The initial size of the TP model is* $0 \times M_2 \times \cdots \times M_K$ *and it can be written as*

$$\mathfrak{s}^{(0)}(\mathbf{p}) = \mathcal{S}^{(0)} \times_1 \boldsymbol{\delta}^{(1,0)}(\mathbf{p}^{(1)}) \mathop{\boxtimes}_{k=2}^{K} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}), \quad \textit{where } \boldsymbol{\delta}^{(1,0)} : \Omega_1 \to \mathbb{R}^0. \tag{5.41}$$

Now, let $g = 1$ *and perform the following steps:*

1. *Construct the actual* $\mathfrak{f}^{(g)}(\mathbf{p})$*, see* (5.39).
2. *Construct TP form* $\mathfrak{s}^{(g)}(\mathbf{p}) = \mathfrak{s}^{(g-1)}(\mathbf{p}) + \mathfrak{f}^{(g)}(\mathbf{p})$ *by concatenating their* 1*-mode weighting functions and their core tensor as*

$$\mathfrak{s}^{(g)}(\mathbf{p}) = \mathcal{F} \times_1 \begin{bmatrix} \boldsymbol{\delta}^{(1,g-1)}(\mathbf{p}^{(1)}) & \boldsymbol{\alpha}^{(1,g)}(\mathbf{p}^{(1)}) \end{bmatrix} \mathop{\boxtimes}_{k=2}^{K} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)})$$

   and with H *denoting the size of* $\boldsymbol{\delta}^{(1,g-1)}(\mathbf{p}^{(1)})$,

$$\mathfrak{f}_{i_1,l_2,..,l_K} = \begin{cases} \mathfrak{s}^{(0)}_{i_1,l_2,..,l_K} & if \;\; i_1 \leq H, \\ \mathfrak{d}^{(g)}_{i_1-H,l_2,..,l_K} & \text{otherwise.} \end{cases} \tag{5.42}$$

3. *Apply Lemma 5.19 and orthonormalization of* 1*-mode weighting functions to reduce its size to* $I_1' \times M_2 \times \ldots M_K$ *where* $I_1' \leq I_1$*:*

$$\mathfrak{s}^{(g)}(\mathbf{p}) = \mathcal{S}^{(g)} \times_1 \boldsymbol{\delta}^{(1,g)}(\mathbf{p}^{(1)}) \mathop{\boxtimes}_{k=2}^{K} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}).$$

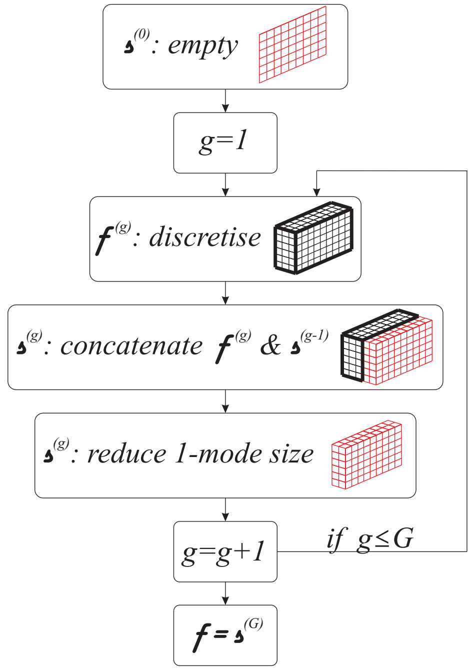


Figure 5.4: Illustration of Algorithm 5.20 considering the case depicted in Fig. 5.3

Increase the value of g and perform it again until $g > G$. Then $\mathfrak{f}(\mathbf{p}) = \mathfrak{s}^{(G)}(\mathbf{p})$.

The resulting TP form is exact, and its weighting functions are orthonormal. By performing the Sequential ASVD (Step 3 of Algorithm 5.16), the exact Affine TP form can be obtained.

General case. Assume that the discretised model would be too large for our capabilities, and furthermore, the expected maximal size of the Affine TP form is $I_1 \times \cdots \times I_K$ and there are assigned one or more modes $k \in \mathfrak{Q} = \{q_1, ..., q_R\} \subset \{1, ..., K\}$, where smaller F_r: $M_{q_r} > F_r \geq I_{q_r}$ sizes should be used.

Now for all $r = 1, ..., R$, divide the original M_{q_r} sizes of the TP form into G_r parts with size F_r (the remainder last one can be smaller). This way, the original discretisation problem is partitioned into $G_1 \times ... \times G_R$ parts and their summation.

Denoting the $(g_1, ..., g_R)$-th part as

$$\mathfrak{f}^{(g_1,...,g_R)}(\mathbf{p}) = \mathcal{D}^{(g_1,...,g_R)} \underset{k \notin \mathfrak{Q}}{\boxtimes} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}) \underset{r=1}{\overset{R}{\boxtimes}} \boldsymbol{\alpha}^{(q_r,g_r)}(\mathbf{p}^{(q_r)}), \tag{5.43}$$

$$\mathcal{D}^{(g_1,...,g_R)}_{f_1,...,f_K} = \mathcal{D}_{d_1,...,d_K} \quad \text{where } d_k = \begin{cases} f_k + F_r(g_k - 1) & \text{if } \exists r : q_r = k, \\ f_k & \text{if } k \notin \mathfrak{Q}, \end{cases}$$

$$\boldsymbol{\alpha}^{(q_r,g_r)}(\mathbf{p}^{(q_r)}) = \left[\alpha^{(q_r)}_{(g_r-1)F_r+1}(\mathbf{p}^{(q_r)}) \quad \ldots \quad \alpha^{(q_r)}_{\min(g_r F_r, M_{q_r})}(\mathbf{p}^{(q_r)}) \right],$$

and then the function can be written as

$$\mathfrak{f}(\mathbf{p}) = \sum_{g_1=1}^{G_1} \cdots \sum_{g_R=1}^{G_R} \mathfrak{f}^{(g_1,\dots,g_R)}(\mathbf{p}). \tag{5.44}$$

The summation is partitioned to summations along one index, the following notation system will be applied to them.

Notation 5.21.

$$\begin{aligned}
\mathfrak{s}^{(R,g_1,\dots,g_{R-1},\overline{g}_R)}(\mathbf{p}) &= \sum_{g_R=1}^{\overline{g}_R} \mathfrak{f}^{(g_1,\dots,g_R)}(\mathbf{p}) = \mathfrak{s}^{(R,g_1,\dots,g_{R-1},\overline{g}_R-1)}(\mathbf{p}) + \mathfrak{f}^{(g_1,\dots,\overline{g}_R)}(\mathbf{p}) \\
\mathfrak{s}^{(R-1,g_1,\dots,g_{R-2},\overline{g}_{R-1})}(\mathbf{p}) &= \sum_{g_{R-1}=1}^{\overline{g}_{R-1}} \mathfrak{s}^{(R,g_1,\dots,g_{R-1},G_R)}(\mathbf{p}) = \\
&= \mathfrak{s}^{(R-1,g_1,\dots,g_{R-2},\overline{g}_{R-1}-1)}(\mathbf{p}) + \mathfrak{s}^{(R,g_1,\dots,\overline{g}_{R-1},G_R)}(\mathbf{p}) \\
&\vdots \\
\mathfrak{s}^{(1,\overline{g}_1)}(\mathbf{p}) &= \sum_{g_1=1}^{\overline{g}_1} \mathfrak{s}^{(2,g_1,G_2)}(\mathbf{p}) = \mathfrak{s}^{(1,\overline{g}_1-1)}(\mathbf{p}) + \mathfrak{s}^{(2,\overline{g}_1,G_2)}(\mathbf{p})
\end{aligned} \tag{5.45}$$

and then

$$\mathfrak{s}^{(r,g_1,\dots,g_{r-1},\overline{g}_r)}(\mathbf{p}) = \sum_{g_r=1}^{\overline{g_r}} \sum_{g_{r+1}=1}^{G_{r+1}} \cdots \sum_{g_R=1}^{G_R} \mathfrak{f}^{(g_1,\dots,g_R)}(\mathbf{p})$$

$\mathfrak{s}^{(1,G_1)}(\mathbf{p}) = \mathfrak{f}(\mathbf{p})$.

The following Lemma describes, what TP forms will be preferred for them and how much is their size.

Lemma 5.22. *Consider the term* $\mathfrak{s}^{(r,g_1,\dots,g_{r-1},\overline{g}_r)}(\mathbf{p})$. *It can be written into a TP form*

$$\mathfrak{s}^{(r,g_1,\dots,g_{r-1},\overline{g}_r)}(\mathbf{p}) = \mathcal{T} \boxtimes_{k\notin\mathfrak{Q}} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}) \boxtimes_{a=1}^{r-1} \boldsymbol{\alpha}^{(q_a,g_a)}(\mathbf{p}^{(q_a)}) \boxtimes_{a=r}^{R} \boldsymbol{\epsilon}^{(a)}(\mathbf{p}^{(q_a)}) \tag{5.46}$$

with sizes $T_1 \times \cdots \times T_K$ *where*

$$T_k = \begin{cases} M_k & \text{if } k \notin \mathfrak{Q}, \\ F_r \text{ or } F'_r & \text{if } \exists r' < r: \; q_{r'} = k, \\ I'_k(\leq I_k) & \text{otherwise.} \end{cases}$$

Furthermore, it can be derived from any discretisation-based initial TP form via orthogonalization of the weighting functions and compression in modes $q_r, \dots, q_R$.

Then the following lemma gives insight into the summation of TP forms given in the upper form.

Algorithm 5.23. *Consider the summation problem*

$$\mathfrak{s}^{(r,g_1,\dots,g_{r-1},g_r)}(\mathbf{p}) = \mathfrak{s}^{(r,g_1,\dots,g_{r-1},g_r-1)}(\mathbf{p}) + \mathfrak{s}^{(r+1,g_1,\dots,g_{r-1},g_r,G_{r+1})}(\mathbf{p}), \tag{5.47}$$

where the terms are denoted as

$$\begin{aligned}
\mathfrak{s}^{(r,g_1,\dots,g_{r-1},g_r-1)}(\mathbf{p}) &= \mathcal{T} \underset{k\notin\mathfrak{Q}}{\boxtimes} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}) \overset{r-1}{\underset{a=1}{\boxtimes}} \boldsymbol{\alpha}^{(q_a,g_a)}(\mathbf{p}^{(q_a)}) \overset{R}{\underset{a=r}{\boxtimes}} \boldsymbol{\epsilon}^{(a)}(\mathbf{p}^{(q_a)}),\\
\mathfrak{s}^{(r+1,g_1,\dots,g_{r-1},g_r,G_{r+1})}(\mathbf{p}) &= \mathcal{S} \underset{k\notin\mathfrak{Q}}{\boxtimes} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}) \overset{r-1}{\underset{a=1}{\boxtimes}} \boldsymbol{\alpha}^{(q_a,g_a)}(\mathbf{p}^{(q_a)}) \overset{R}{\underset{a=r}{\boxtimes}} \boldsymbol{\delta}^{(a)}(\mathbf{p}^{(q_a)})
\end{aligned}$$

with sizes in the $k \in \{q_r, ..., q_R\}$ dimensions $T_{q_r}, ..., T_{q_R}$, and $S_{q_r}, ..., S_{q_R}$, respectively. Their sum can be written in TP form

$$\mathfrak{s}^{(r,g_1,\dots,g_{r-1},g_r)}(\mathbf{p}) = \mathcal{Z} \underset{k\notin\mathfrak{Q}}{\boxtimes} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}) \overset{r-1}{\underset{a=1}{\boxtimes}} \boldsymbol{\alpha}^{(q_a,g_a)}(\mathbf{p}^{(q_a)}) \overset{R}{\underset{a=r}{\boxtimes}} \boldsymbol{\eta}^{(a)}(\mathbf{p}^{(q_a)}) \tag{5.48}$$

and its sizes in the $k \in \{q_r, ..., q_R\}$ dimensions are $(T_{q_r} + S_{q_r}), ..., (T_{q_R} + S_{q_R})$, the other sizes are the same as the previous TP forms, and

$$\boldsymbol{\eta}^{(a)}(\mathbf{p}^{(q_a)}) = \begin{bmatrix}\boldsymbol{\epsilon}^{(a)}(\mathbf{p}^{(q_a)}) & \boldsymbol{\delta}^{(a)}(\mathbf{p}^{(q_a)})\end{bmatrix} \tag{5.49}$$

$$\mathcal{Z}_{z_1,\dots,z_K} = \begin{cases} \mathcal{T}_{z_1,\dots,z_K} & if\ \forall a = r, ..., R: \quad z_{q_a} \le T_{q_a}, \\ \mathcal{S}_{s_1,\dots,s_K} & if\ \forall a = r, ..., R: \quad z_{q_a} > T_{q_a}\ , \\ 0 & otherwise, \end{cases} \tag{5.50}$$

$$where \quad s_k = \begin{cases} z_k - T_k & if\ \exists a : r \le a \le R\ \&\ k = q_a, \\ z_k & otherwise, \end{cases} \tag{5.51}$$

which is the concatenation of the weighting function and the hyper-diagonal copy of $\mathcal{S}$ in the increased $\mathcal{T}$ tensor.

This way, the discretization can be done, see the following theorem.

Theorem 5.24. *Consider the summations in* (5.45)*, and initialize empty TP forms as*

$$\begin{aligned}
\mathfrak{s}^{(R)} &= \mathfrak{s}^{(R,g_1,\dots,g_{R-1},0)}(\mathbf{p}),\\
\mathfrak{s}^{(R-1)} &= \mathfrak{s}^{(R-1,g_1,\dots,g_{R-2},0)}(\mathbf{p}),\\
&\vdots\\
\mathfrak{s}^{(1)} &= \mathfrak{s}^{(1,0)}(\mathbf{p}).
\end{aligned}$$

Based on Algorithm 5.23, the summations in (5.45) *can be performed resulting in the* $\mathfrak{f}(\mathbf{p})$ *value in variable* $\mathfrak{s}^{(1)}$*, meanwhile the necessary maximal size of the r-th stored form will be $T_1, \dots, T_R$ in the summed modes and $T_a = F_a$ if $a < r$ and $T_a = 2I_a$ if $a \ge r$, exhausting the benefits of Lemma 5.22.*

This way, the number of weighting functions to be stored for the k-th parameter dependency is

$$\begin{cases} M_k & \text{if } k \notin \mathfrak{Q} \\ F_r + 2rI_k & \text{otherwise (where } q_r = k) \end{cases}$$

and the number of tensor elements to be stored is

$$\left(\prod_{k\notin\mathfrak{Q}} M_k\right)\left(\prod_{r=1}^{R} F_r + (2I_R)\prod_{r=1}^{R-1} F_r + (2I_R)(2I_{R-1})\prod_{r=1}^{R-2} F_r + \cdots + \prod_{r=1}^{R}(2I_r)\right) \leq$$
$$\leq \left(\prod_{k\notin\mathfrak{Q}} M_k\right)\left(\prod_{r=1}^{R} F_r\right)(2^{R+1}-1).$$

Increase sampling density of a given Affine TP form

The density of discretization points can be increased on a given Affine TP form as in Step 3. of the original TP model transformation via the following method.

Algorithm 5.25. *The approximating value $\mathbf{v}^{(k)}(\mathbf{p}^{(k)}) = \begin{bmatrix}\mathbf{u}^{(k)}(\mathbf{p}^{(k)}) & 1\end{bmatrix}$ between the discretisation points (in general) can be determined in the following way:*

Choose an X subset from the Ω domain, such that the parameter sets $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ obtained from vectors $\mathbf{x} \in X$ fulfil the following condition:

- $\mathbf{x}^{(k)} = \mathbf{p}^{(k)}$,
- *the other $\mathbf{x}^{(l)}$ vectors ($l \neq k$) are in discretisation points, so*

$$\exists m_l \text{ such that } \mathbf{x}^{(l)} = \mathbf{g}^{(l)}_{m_l}. \tag{5.52}$$

Then, the best approximation (on the X set) can be obtained as

$$\mathbf{u}^{(k)}(\mathbf{p}^{(k)}) = \begin{bmatrix}\cdots & (\mathfrak{f}(\mathbf{x}))_{(1)} & \cdots\end{bmatrix}_{\mathbf{x}\in X}\begin{bmatrix}\cdots & \left(\mathcal{D} \underset{l=1,l\neq k}{\overset{K}{\boxtimes}} \mathbf{v}^{(l)}(\mathbf{x}^{(l)})\right)_{(k)} & \cdots\end{bmatrix}^{+}_{\mathbf{x}\in X}, \tag{5.53}$$

where

$$\mathfrak{f}(\mathbf{x}) = \mathfrak{g}(\mathbf{x}) - \mathcal{S}_{d_k=D_k+1} \underset{l=1,l\neq k}{\overset{K}{\boxtimes}} \mathbf{v}^{(l)}(\mathbf{x}^{(l)}), \tag{5.54}$$

$$\mathcal{D} = \mathcal{S} \times_k \begin{bmatrix}\mathbf{E}^{D_k} & \mathbf{0}^{D_k\times 1}\end{bmatrix}, \tag{5.55}$$

and $\mathfrak{f}(\mathbf{x}) \in H^{1\times 1\times\cdots}$, so $(\mathfrak{f}(\mathbf{x}))_{(1)} \in \mathbb{R}^{1\times R}$, where R is as in Definition 5.2.

5.4 Summary

The chapter extended the definition of Polytopic TP form and introduced a new tensor algebraic notation system according to the recently emerged practical reasons.

For its numerical derivation, the Affine TP form was defined, and its properties were shown. Finally, more algorithms were proposed for its numerical reconstruction.

5.5 Proofs

Proof of Lemma 5.4 and 5.5. *See [49].* ■

Proof of Lemma 5.10. *Because for linear operators* $\mathrm{op}(\cdot)$*:* $\mathrm{op}(\sum \alpha_i \mathfrak{a}_i) = \sum \alpha_i \mathrm{op}(\mathfrak{a}_i)$. ■

Proof of Lemma 5.11. *Considering only addition for sake of brevity:*

$$\mathfrak{h}(\mathbf{p}) = \sum_{j_{1,1}} \cdots \sum_{j_{1,h_1}} \sum_{j_{2,1}} \cdots \sum_{j_{K,h_K}} \mathfrak{h}_{j_{1,1},\ldots,j_{K,h_k}} \prod_{k=1}^{K} \prod_{h=1}^{h_k} w_{j_{k,h}}^{(k)}(\mathbf{p}^{(k)}) =$$

$$= \sum_{j_{1,1}} \cdots \sum_{j_{1,h_1}} \sum_{j_{2,1}} \cdots \sum_{j_{K,h_K}} (\mathfrak{f}_{j_{1,1},\ldots,j_{K,f_k}} + \mathfrak{g}_{j_{1,1},\ldots,j_{K,g_k}}) \prod_{k=1}^{K} \prod_{h=1}^{h_k} w_{j_{k,h}}^{(k)}(\mathbf{p}^{(k)}) =$$

$$= \sum_{j_{1,1}} \cdots \sum_{j_{1,f_1}} \sum_{j_{2,1}} \cdots \sum_{j_{K,f_K}} \mathfrak{f}_{j_{1,1},\ldots,j_{K,f_k}} \prod_{k=1}^{K} \prod_{f=1}^{f_k} w_{j_{k,f}}^{(k)}(\mathbf{p}^{(k)}) \sum \cdots \sum \prod_{k=1}^{K} \prod_{h=f_k+1}^{h_k} w_{j_{k,h}}^{(k)}(\mathbf{p}^{(k)}) +$$

$$+ \sum_{j_{1,1}} \cdots \sum_{j_{1,g_1}} \sum_{j_{2,1}} \cdots \sum_{j_{K,g_K}} \mathfrak{g}_{j_{1,1},\ldots,j_{K,g_k}} \prod_{k=1}^{K} \prod_{g=1}^{g_k} w_{j_{k,g}}^{(k)}(\mathbf{p}^{(k)}) \sum \cdots \sum \prod_{k=1}^{K} \prod_{h=g_k+1}^{h_k} w_{j_{k,h}}^{(k)}(\mathbf{p}^{(k)}) =$$

$$= \sum_{j_{1,1}} \cdots \sum_{j_{1,f_1}} \sum_{j_{2,1}} \cdots \sum_{j_{K,f_K}} \mathfrak{f}_{j_{1,1},\ldots,j_{K,f_k}} \prod_{k=1}^{K} \prod_{f=1}^{f_k} w_{j_{k,f}}^{(k)}(\mathbf{p}^{(k)}) +$$

$$+ \sum_{j_{1,1}} \cdots \sum_{j_{1,g_1}} \sum_{j_{2,1}} \cdots \sum_{j_{K,g_K}} \mathfrak{g}_{j_{1,1},\ldots,j_{K,g_k}} \prod_{k=1}^{K} \prod_{g=1}^{g_k} w_{j_{k,g}}^{(k)}(\mathbf{p}^{(k)}) = \mathfrak{f}(\mathbf{p}) + \mathfrak{g}(\mathbf{p}). \blacksquare$$

Proof of Lemma 5.12. *It is easy to see, that*

$$\mathfrak{h}(\mathbf{p}) = \sum_{j_{1,1}} \cdots \sum_{j_{1,h_1}} \sum_{j_{2,1}} \cdots \sum_{j_{K,h_K}} \mathfrak{h}_{j_{1,1},\ldots,j_{K,h_k}} \prod_{k=1}^{K} \prod_{h=1}^{h_k} w_{j_{k,h}}^{(k)}(\mathbf{p}^{(k)}) =$$

$$= \sum_{m_{1,1}} \cdots \sum_{m_{1,f_1}} \sum_{n_{1,1}} \cdots \sum_{n_{1,g_1}} \sum_{m_{2,1}} \cdots \sum_{n_{K,g_K}} \mathfrak{f}_{m_{1,1},\ldots,m_{K,f_k}} \mathfrak{g}_{n_{1,1},\ldots,n_{K,g_k}} \prod_{k=1}^{K} \prod_{f=1}^{f_k} w_{m_{k,f}}^{(k)}(\mathbf{p}^{(k)})$$

$$\prod_{k=1}^{K} \prod_{g=1}^{g_k} w_{n_{k,g}}^{(k)}(\mathbf{p}^{(k)}) =$$

$$\left(\sum_{m_{1,1}}\cdots\sum_{m_{1,f_1}}\sum_{m_{2,1}}\cdots\sum_{m_{K,f_K}}\mathfrak{f}_{m_{1,1},\dots,m_{K,f_k}}\prod_{k=1}^{K}\prod_{f=1}^{f_k}w^{(k)}_{m_{k,f}}(\mathbf{p}^{(k)})\right)\cdot$$

$$\cdot\left(\sum_{n_{1,1}}\cdots\sum_{n_{1,g_1}}\sum_{n_{2,1}}\cdots\sum_{n_{K,g_K}}\mathfrak{g}_{n_{1,1},\dots,n_{K,g_k}}\prod_{k=1}^{K}\prod_{g=1}^{g_k}w^{(k)}_{n_{k,g}}(\mathbf{p}^{(k)})\right)=\mathfrak{f}(\mathbf{p})\cdot\mathfrak{g}(\mathbf{p}).\blacksquare$$

Proof of Theorem 5.14. *By substituting the polytopic forms into the Affine TP form for all parameter dependencies and applying Lemma 5.5.* $\blacksquare$

The following, fundamental lemmas will be necessary to prove Theorem 5.15 and 5.17.

Lemma 5.26. ***(Inner product and norm of orthonormal TP forms)*** *If there are two TP functions given on the same, orthonormal weighting function system and disjoint parameter sets, as*

$$\mathfrak{c}(\mathbf{p})=\mathcal{C}\boxtimes_{k=1}^{K}\mathbf{f}^{(k)}(\mathbf{p}^{(k)}),\quad \mathfrak{d}(\mathbf{p})=\mathcal{D}\boxtimes_{k=1}^{K}\mathbf{f}^{(k)}(\mathbf{p}^{(k)})$$

then their inner product can be written as

$$<\mathfrak{c},\mathfrak{d}>=<\mathcal{C},\mathcal{D}>.$$

Furthermore, the norm

$$||\mathfrak{c}||=||\mathcal{C}||.$$

Proof. *The inner product can be written as*

$$<\mathfrak{c},\mathfrak{d}>=\frac{1}{\prod_{k=1}^{K}V(\Omega_k)}\int_{\mathbf{p}^{(1)}\in\Omega_1}\cdots\int_{\mathbf{p}^{(K)}\in\Omega_K}<\mathfrak{c}(\mathbf{p}),\mathfrak{d}(\mathbf{p})>\prod_{k=1}^{K}V(d\mathbf{p}^{(k)})=$$

$$=\int_{\mathbf{p}^{(1)}\in\Omega_1}\cdots\int_{\mathbf{p}^{(K)}\in\Omega_K}<\mathcal{C}\boxtimes_{k=1}^{K}\mathbf{f}^{(k)}(\mathbf{p}^{(k)}),\mathcal{D}\boxtimes_{k=1}^{K}\mathbf{f}^{(k)}(\mathbf{p}^{(k)})>\prod_{k=1}^{K}\frac{V(d\mathbf{p}^{(k)})}{V(\Omega_k)}=$$

$$\sum_{m_1=1}^{M_1}\cdots\sum_{m_K=1}^{M_K}\sum_{n_1=1}^{M_1}\cdots\sum_{n_K=1}^{M_K}<\mathcal{C}_{m_1,\dots,m_K},\mathcal{D}_{n_1,\dots,n_K}>\prod_{k=1}^{K}\int_{\mathbf{p}^{(k)}\in\Omega_k}f^{(k)}_{m_k}(\mathbf{p}^{(k)})f^{(k)}_{n_k}(\mathbf{p}^{(k)})\frac{V(d\mathbf{p}^{(k)})}{V(\Omega_k)}=$$

$$=\sum_{m_1=1}^{M_1}\cdots\sum_{m_K=1}^{M_K}\sum_{n_1=1}^{M_1}\cdots\sum_{n_K=1}^{M_K}<\mathcal{C}_{m_1,\dots,m_K},\mathcal{D}_{n_1,\dots,n_K}>\prod_{k=1}^{K}\delta_{m_kn_k}=$$

$$=\sum_{m_1=1}^{M_1}\cdots\sum_{m_K=1}^{M_K}<\mathcal{C}_{m_1,\dots,m_K},\mathcal{D}_{m_1,\dots,m_K}>=<\mathcal{C},\mathcal{D}>.\blacksquare$$

This way, the orthogonality and norm of functions along the parameter sets can be lead back to the properties of the core tensor.

This way, the function's orthogonality and norm along the parameter sets depends only on the orthogonality and norm of the core tensors. Based on this property, the following lemma helps to understand the structure of the Affine TP form and to obtain it.

Lemma 5.27 (k-mode ASVD)**.** *If there are $\mathbf{f}^{(l)}(\mathbf{p}^{(l)})$ orthonormal weighting functions for $l = 1..K$, the following statements are equivalent:*

- *The following form is an ASVD along $\mathbf{p}^{(k)}$ parameter*

$$\left(\mathcal{K} \mathop{\boxtimes}_{l=1,l\neq k}^{K} \mathbf{f}^{(l)}(\mathbf{p}^{(l)})\right) \times_k \mathbf{f}^{(k)}(\mathbf{p}^{(k)}),$$

- *The form is an ASVD along $\mathbf{p}^{(k)}$ parameter*

$$\mathcal{K} \times_k \mathbf{f}^{(k)}(\mathbf{p}^{(k)}).$$

Furthermore, if the parameter sets are disjoint, the $\sigma_1^{(k)}, \ldots, \sigma_{D_k}^{(k)}$ singular values are the same.

Proof. *The requirements for $\mathbf{f}^{(k)}(\mathbf{p}^{(k)})$ weighting functions are the same. Furthermore, for inner product of the n-mode subtensors are same*

$$< \mathcal{K}_{d_n=i} \mathop{\boxtimes}_{l=1,l\neq k}^{K} \mathbf{f}^{(l)}(\mathbf{p}^{(l)}), \mathcal{K}_{d_n=j} \mathop{\boxtimes}_{l=1,l\neq k}^{K} \mathbf{f}^{(l)}(\mathbf{p}^{(l)}) >=< \mathcal{K}_{d_n=i}, \mathcal{K}_{d_n=j} >$$

from Lemma 5.26. This way, their orthogonality and order are the same for ASVD. ■

This way, the k-mode singular values can be obtained as norm of the k-mode subtensors of the core tensor and the ASVD on $\mathbf{p}^{(k)}$ parameter dependency is invariant for inner transformations between orthonormal decomposition on other $\mathbf{p}^{(k)}$ parameter dependencies.

Proof of Theorem 5.15. *These forms are Affine TP forms because*

$$\mathfrak{g}(\mathbf{p}) = \left(\mathcal{G}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{T}^{(k)} \mathop{\boxtimes}_{k=1,k\neq l}^{K} \left[\mathbf{v}^{(k)}(\mathbf{p}^{(k)})\mathbf{T}^{(k)T}\right]\right) \times_l \left(\mathbf{v}^{(l)}(\mathbf{p}^{(l)})\mathbf{T}^{(l)T}\right)$$

is ASVD, because

$$\left(\mathcal{G}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{T}^{(k)}\right) \times_l \left(\mathbf{v}^{(l)}(\mathbf{p}^{(l)})\mathbf{T}^{(l)T}\right)$$

is ASVD (based on Lemma 5.27), because

$$\left(\mathcal{G}^{aff} \times_l \mathbf{T}^{(l)}\right) \times_l \left(\mathbf{v}^{(l)}(\mathbf{p}^{(l)})\mathbf{T}^{(l)T}\right)$$

is ASVD based on Proposition 4.5.

Furthermore, if the parameter sets are disjoint only these forms are Affine TP forms, because of the form

$$\mathfrak{g}(\mathbf{p}) = \mathcal{G}'^{aff} \boxtimes_{k=1}^{K} \mathbf{v}'^{(k)}(\mathbf{p}^{(k)})$$

is an Affine TP form only if

$$\left(\mathcal{G}'^{aff} \boxtimes_{k=1,k\neq l}^{K} \mathbf{v}'^{(k)}(\mathbf{p}^{(k)})\right) \times_l \mathbf{v}'^{(l)}(\mathbf{p}^{(l)})$$

is an ASVD. If the parameter sets are disjoint, its uniqueness comes from Prop. 4.5 for all $l = 1..K$. ■

Property 5.28. *Algorithm 5.16 provides an Affine TP form with same exactness as the initial TP form derived in Step 1.*

Proof. *It comes from Lemma 5.27.* ■

Proof of Theorem 5.17. *Construct the $\hat{\mathcal{G}}^{aff}$ tensor with the same sizes as $\mathcal{G}^{aff}$, that contains zeros in the disregarded subtensors. Then, if $\Delta G^{aff} = \mathcal{G}^{aff} - \hat{\mathcal{G}}^{aff}$, the approximation error function can be written as*

$$\mathfrak{g}(\mathbf{p}) - \hat{\mathfrak{g}}(\mathbf{p}) = \Delta\mathcal{G}^{aff} \boxtimes_{k=1}^{K} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}).$$

If only one k-mode dimension is decreased, the error function of the approximation can be written as (based on Proposition 4.6)

$$||\mathfrak{g} - \hat{\mathfrak{g}}||^2 = \sum_{d=1}^{D_k+1} ||\Delta\mathcal{G}^{aff}_{d_k=d}||^2 = \sum_{d=D_k-\Delta D_k+1}^{D_k} \sigma_d^{(k)2}$$

that is minimal as Proposition 4.6 said. Furthermore, if more k-mode dimension is decreased the worst case error is the sum of theses values.
The error to be minimized can be written as

$$\begin{aligned} e^2 = ||\mathfrak{g} - \hat{\mathfrak{g}}||^2 = ||\mathfrak{g}||^2 + ||\hat{\mathfrak{g}}||^2 - 2 < \mathfrak{g}, \hat{\mathfrak{g}} >= \\ = ||\mathcal{G} \boxtimes_{k=1}^{K} \boldsymbol{\gamma}^{(k)}||^2 + ||\hat{\mathcal{G}} \boxtimes_{k=1}^{K} \hat{\boldsymbol{\gamma}}^{(k)}||^2 - 2 < \mathcal{G} \boxtimes_{k=1}^{K} \boldsymbol{\gamma}^{(k)}, \hat{\mathcal{G}} \boxtimes_{k=1}^{K} \hat{\boldsymbol{\gamma}}^{(k)} > \end{aligned}$$

By writing the new weighting functions in the following form

$$\hat{\boldsymbol{\gamma}}^{(k)}(\mathbf{p}^{(k)}) = \begin{bmatrix} \boldsymbol{\gamma}^{(k)}(\mathbf{p}^{(k)}) & \boldsymbol{\gamma}_{\perp}^{(k)}(\mathbf{p}^{(k)}) \end{bmatrix} \begin{bmatrix} \mathbf{U}^{(k)} \\ \mathbf{B}^{(k)} \end{bmatrix}$$

and exploiting their orthonormality the error can be written as

$$e^2 = ||\mathcal{G}||^2 + ||\hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \mathbf{U}^{(k)}||^2 + ||\hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \begin{bmatrix} \mathbf{U}^{(k)} \\ \mathbf{B}^{(k)} \end{bmatrix} - \hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \begin{bmatrix} \mathbf{U}^{(k)} \\ \mathbf{0} \end{bmatrix} ||^2 - 2 < \mathcal{G} \underset{k=1}{\overset{K}{\boxtimes}} \boldsymbol{\gamma}^{(k)}, \hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \hat{\boldsymbol{\gamma}}^{(k)} >$$

and the last term can be expanded as

$$< \mathcal{G} \underset{k=1}{\overset{K}{\boxtimes}} \boldsymbol{\gamma}^{(k)}, \hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \hat{\boldsymbol{\gamma}}^{(k)} >=$$

$$= \int_{\mathbf{p}^{(1)}} \cdots \int_{\mathbf{p}^{(K)}} \sum_{i_1,\ldots,i_K} \sum_{j_1,\ldots,j_K} < \mathfrak{g}_{i_1,\ldots,i_K}, \hat{\mathfrak{g}}_{j_1,\ldots,j_K} > \prod_k \gamma_{i_k}^{(k)}(\mathbf{p}^{(k)}) \hat{\gamma}_{j_k}^{(k)}(\mathbf{p}^{(k)}) \frac{V(d\mathbf{p}^{(k)})}{V(\Omega_k)} =$$
$$= \sum_{i_1,\ldots,i_K} \sum_{j_1,\ldots,j_K} < \mathfrak{g}_{i_1,\ldots,i_K}, \hat{\mathfrak{g}}_{j_1,\ldots,j_K} > \prod_k < \gamma_{i_k}^{(k)}, \hat{\gamma}_{j_k}^{(k)} >=$$
$$= \sum_{i_1,\ldots,i_K} \sum_{j_1,\ldots,j_K} < \mathfrak{g}_{i_1,\ldots,i_K}, \hat{\mathfrak{g}}_{j_1,\ldots,j_K} > \prod_k \left(\sum_{d=1}^{D_k+1} u_{d,j_k}^{(k)} < \gamma_{i_k}^{(k)}, \gamma_d^{(k)} > + \right.$$
$$\left. + \sum_{d=0}^{0} b_{d,j_k}^{(k)} < \gamma_{i_k}^{(k)}, \gamma_{\perp,d}^{(k)} > \right) = \sum_{i_1,\ldots,i_K} \sum_{j_1,\ldots,j_K} < \mathfrak{g}_{i_1,\ldots,i_K}, \hat{\mathfrak{g}}_{j_1,\ldots,j_K} > \prod_k \left(\sum_{d=1}^{D_k+1} u_{d,j_k}^{(k)} \delta_{i_k,d} + \right.$$
$$\left. + \sum_{d=0}^{0} b_{d,j_k}^{(k)} 0 \right) = \sum_{i_1,\ldots,i_K} \sum_{j_1,\ldots,j_K} < \mathfrak{g}_{i_1,\ldots,i_K}, \hat{\mathfrak{g}}_{j_1,\ldots,j_K} > \prod_k u_{i_k,j_k}^{(k)} =< \mathcal{G}, \hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \mathbf{U}^{(k)} > .$$

This way,

$$e^2 = ||\mathcal{G}||^2 - 2 < \mathcal{G}, \hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \mathbf{U}^{(k)} > + ||\hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \mathbf{U}^{(k)}||^2 + ||\hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \begin{bmatrix} \mathbf{U}^{(k)} \\ \mathbf{B}^{(k)} \end{bmatrix} - \hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \begin{bmatrix} \mathbf{U}^{(k)} \\ \mathbf{0} \end{bmatrix} ||^2 =$$
$$= ||\mathcal{G} - \hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \mathbf{U}^{(k)}||^2 + ||\hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \begin{bmatrix} \mathbf{U}^{(k)} \\ \mathbf{B}^{(k)} \end{bmatrix} - \hat{\mathcal{G}} \underset{k=1}{\overset{K}{\boxtimes}} \begin{bmatrix} \mathbf{U}^{(k)} \\ \mathbf{0} \end{bmatrix} ||^2$$

where the second term is minimal (zero) by choosing $\mathbf{B}^{(k)} = \mathbf{0}$ *and the remaining part is a* $r_1, .., r_K$ *rank approximation problem. By using*

$$\mathbf{U}^{(k)} = \begin{bmatrix} \mathbf{U}_0^{(k)} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$$

orthogonal matrix candidates, the new weighting functions will be homogeneous orthonormal functions and it results in the best $d_1, .., d_K$ *dimension approximation.* ■

Property 5.29. *Algorithm 5.18 provides an Affine TP form with decreased complexity by adding one more parameter dependency reserving the exactness.*

Proof. *The constructed TP form is exact:*

$$\mathcal{G}^{aff2} \mathop{\boxtimes}_{k=1}^{K+1} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}) = \sum_{d_1}^{D_1+1} \cdots \sum_{d_{K+1}}^{D_{K+1}+1} \prod_{k=1}^{K+1} v_{d_k}^{(k)}(\mathbf{p}^{(k)}) \left(h(d_{K+1}) \cdot \mathfrak{g}_{d_1,\dots,d_K}^{aff} + f(d_1,..,d_K) \cdot \mathfrak{e}_{d_{K+1}}^{aff} \right) =$$

$$= \sum_{d_1}^{D_1+1} \cdots \sum_{d_K}^{D_K+1} \prod_{k=1}^{K} v_{d_k}^{(k)}(\mathbf{p}^{(k)}) \cdot 1 \cdot \mathfrak{g}_{d_1,\dots,d_K}^{aff} + \sum_{d}^{D_{K+1}+1} 1 \cdot v_d^{(K+1)}(\mathbf{p}^{(K+1)}) \mathfrak{e}_d^{aff} =$$

$$= \mathcal{G}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}) + \mathcal{E}^{aff} \times_1 \mathbf{v}^{(K+1)}(\mathbf{p}) = \hat{\mathfrak{g}}(\mathbf{p}) + \mathfrak{e}(\mathbf{p}) = \mathfrak{g}(\mathbf{p}),$$

where we used the fact that $v_{D_k+1}^{(k)}(\mathbf{p}^{(k)}) = 1$ *for all* $k = 1..K$.

The form is affine, because the weighting functions are orthonormal and homogeneous coordinates, furthermore for the $i,j \leq D_k$*-th* $k \leq K$*-mode subtensors*

$$\mathcal{G}_{d_k=i,d_{K+1}=d}^{aff2} = \delta_{d,D_{K+1}+1} \mathcal{G}_{d_k=i}^{aff}.$$

This way, the subtensors are orthogonal and they have the original norms:

$$< \mathcal{G}_{d_k=i}^{aff2}, \mathcal{G}_{d_k=j}^{aff2} >= \sum_{d=1}^{D_{K+1}+1} < \mathcal{G}_{d_k=i,d_{K+1}=d}^{aff2}, \mathcal{G}_{d_k=j,d_{K+1}=d}^{aff2} >=$$

$$=< \mathcal{G}_{d_k=i}^{aff}, \mathcal{G}_{d_k=j}^{aff} >= \delta_{ij} \sigma_i^{(k)2}.$$

For $k = K+1$ *similarly: for all* $i,j \leq D_{K+1}$

$$\mathfrak{g}_{d_1,\dots,d_K,i}^{aff2} = \mathfrak{e}_i \prod_{k=1}^{K} \delta_{d_k,D_k+1}.$$

This way, the subtensors are orthogonal and they has the original norms:

$$< \mathcal{G}_{d_k=i}^{aff2}, \mathcal{G}_{d_k=j}^{aff2} >= \sum_{d_1=1}^{D_1+1} \cdots \sum_{d_K=1}^{D_K+1} < \mathfrak{g}_{d_1,\dots,d_K,i}^{aff2}, \mathfrak{g}_{d_1,\dots,d_K,j}^{aff2} >=< \mathfrak{e}_i, \mathfrak{e}_j >= \delta_{ij} \sigma_i^{(K+1)2}. \blacksquare$$

Proof of Lemma 5.19. *From the properties of tensor unfold.* ■

Proof of Lemma 5.22. *The concept is the same as derivation of Affine TP form.* ■

Proof of Theorem 5.24. *From Algorithm 5.23 and Lemma 5.22.* ■

Property 5.30. *Algorithm 5.25 provides the best approximating values for the weighting functions.*

Proof. *Behind the method, there is the fact, that the equation*

$$\mathfrak{g}(\mathbf{x}) = \left(\mathcal{S} \underset{l=1, l\neq k}{\overset{K}{\boxtimes}} \mathbf{v}^{(l)}(\mathbf{x}^{(l)}) \right) \times_k \mathbf{v}^{(k)}(\mathbf{p}^{(k)}),$$

should be guaranteed for all $\mathbf{x} \in X$, that can be written as

$$\mathfrak{f}(\mathbf{x}) = \left(\mathcal{D} \underset{l=1, l\neq k}{\overset{K}{\boxtimes}} \mathbf{v}^{(l)}(\mathbf{x}^{(l)}) \right) \times_k \mathbf{u}^{(k)}(\mathbf{p}^{(k)}),$$

after unfolds

$$(\mathfrak{f}(\mathbf{x}))_{(1)} = \mathbf{u}^{(k)}(\mathbf{p}^{(k)}) \left(\mathcal{D} \underset{l=1, l\neq k}{\overset{K}{\boxtimes}} \mathbf{v}^{(l)}(\mathbf{x}^{(l)}) \right)_{(k)},$$

ordering the equations into vector-matrix form the pseudoinverse gives the best approximation of $\mathbf{u}^{(k)}(\mathbf{p}^{(k)})$. ■

Chapter 6

Polytopic Tensor-Product Models for control purposes

The chapter deals with the control oriented application of affine tensor-product model transformation. For sake of readability, LPV/qLPV models are considered, but the concept and the methods can be applied on any system description that constitutes a Hilbert-space with an appropriately chosen scalar product (e.g., polynomial systems for Sum of Squares optimisation based control design, [189].)

The main issue is that an inadequate polytopic model may highly reduce the performance that can be achieved using a given control design method. Problems are caused by the inclusion of irrelevant and often non-stabilizable LTI systems into the polytope [73, 102, 153, 174]. For example, if such a polytopic model is chosen which includes an uncontrollable system description, the polytopic model is uncontrollable for the control design methods, even if the LPV/qLPV was controllable.

That is, it is essential to avoid or at least minimize their presence without significantly increasing the number of vertices. Since the exact convex hull contains too many (or infinitely many) vertices, the overall reasonable goal is to find an approximating polytope with a small number of vertices. The minimal volume enclosing polytope with given number of vertices is a distinct interpretation of tight enclosing polytope. However, its exact shape and geometric alignment around the actual systems are also essential.

The chapter – according to the problem – proposes methods to determine suitable enclosing polytopes for TP Model Transformation. It provides a method to generate (near) minimal volume enclosing simplex polytopes and to manipulate them to increase the achievable performance of control design based on the polytopic model. Furthermore, a method is proposed for generating (locally) minimal volume non-simplex enclosing polytopes. Because it considers the polytope as the intersection of half-spaces, it allows the simple addition of new half-spaces to cut off irrelevant regions or optimization of their orientation. The methods are defined for higher dimensional Euclidean spaces in general.

Based on the chapter, polytopic TP forms of LPV/qLPV models can be determined in a systematic and computationally efficient way. The resulting polytopic forms can effectively serve as direct input for Linear Matrix Inequality-based synthesis methods in the next chapter.

This chapter is structured as follows: First Section 6.1 discusses the sources of conservativeness around polytopic TP model-based controller design as the motivation of the followings. Then Section 6.2 describes the concept of polytopic TP model generation and manipulation. Following that Section 6.3 proposes methods for generation and manipulation of simplex enclosing polytopes and Section 6.4 provides methods for non-simplex enclosing polytopes. Finally, Section 6.5 concludes the chapter and Section 6.6 briefly discusses the corresponding proofs.

6.1 Problem formulation

Consider the following typical form of LPV/qLPV models

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \\ \mathbf{z}(t) \end{bmatrix} = \mathbf{S}(\mathbf{p}(t)) \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \\ \mathbf{v}(t) \end{bmatrix}, \tag{6.1}$$

where

- $\mathbf{x}(t)$ denotes the state variables, $\mathbf{u}(t)$ the controlled inputs, $\mathbf{v}(t)$ the disturbances, $\mathbf{y}(t)$ the measured outputs and $\mathbf{z}(t)$ the performance outputs.

- $\mathbf{S}(\mathbf{p}(t))$ can be partitioned to $\mathbf{A}(\mathbf{p}(t))$, $\mathbf{B}(\mathbf{p}(t))$, $\mathbf{C}(\mathbf{p}(t))$ etc. matrices, which describe the dynamical behaviour and the function is defined over a hyper-rectangular parameter domain:

$$\mathbf{p} \in \Omega = \left[\underline{p}_1, \overline{p}_1\right] \times \cdots \times \left[\underline{p}_N, \overline{p}_N\right] \subset \mathbb{R}^N. \tag{6.2}$$

- The space of $\mathbf{S}(\mathbf{p}(t))$ is denoted by $\mathbb{S}$ and it is a Hilbert-space with elementary product as if $\mathbf{S}, \mathbf{P} \in \mathbb{S}$: $<\mathbf{S}, \mathbf{P}>= \mathrm{Trace}(\mathbf{S}^T\mathbf{P})$.

For sake of simplicity, assume the most simple polytopic TP form here, where $K = 1$, $\mathbf{p}^{(1)} = \mathbf{p}$. As such, $\mathbf{S}(\mathbf{p}(t))$ can be written as

$$\mathbf{S}(\mathbf{p}) = \sum_{j=1}^{J} w_j(\mathbf{p})\mathbf{S}_j, \tag{6.3}$$

where J denotes the number of vertex systems.

From a high-level view, the control system analysis and synthesis methods determine a scalar performance aspect or design a controller that minimizes it (e.g., LQ cost, H_2, H_∞ norm, decay rate, etc.) The methods consider the whole polytope constructed by the vertices, in which way, they can give guarantees for all $\mathbf{p} \in \Omega$ parameter values or all parameter trajectories $\mathbf{p}(t) \in \Omega$.

But the envelope of the polytopic model usually includes a larger set of LTI systems than the actual LPV/qLPV model does, so

$$\big\{\mathbf{S}(\mathbf{p})\big|\mathbf{p} \in \Omega\big\} \subseteq \mathrm{Conv}(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_J). \tag{6.4}$$

From the aspects of system analysis and synthesis, only the elements of $\big\{\mathbf{S}(\mathbf{p})\big|\mathbf{p} \in \Omega\big\}$ are relevant. Additional systems included in the envelope are irrelevant.

The conservativeness of polytopic model-based methods, in general, originates from the facts that

1. the irrelevant LTI systems that are not stabilizable or limit the achievable performance,
2. the LMI methods (derived from methodologies for LTI systems) cannot express necessary conditions either for stability and performance of the polytopic model [148].

The possible relaxations of LMI based design include various aspects: In general the stability conditions are relaxed by applying parameter-dependent Lyapunov-function candidates [77, 85], and parameter-dependant controller candidates with increasing complexity [135]. The extraction of multiple polytopic sums in semidefinite constraints can also cause conservativeness [61, 84, 118, 150, 169, 178], as well as the special derivations of different methods of output feedback controller [51, 53, 110].

The conservativeness caused by the presence of irrelevant dynamics is not profoundly investigated in the literature. There are published results about excluding volume from the polytope during the LMI-based design [147, 149, 151] although it is not an easy task to define the volumes to be excluded in higher dimensional spaces in general.

The methods defined in the following allow the determination of (near) minimal volume enclosing polytopes to minimize the amount of included irrelevant systems. Then (if the performance with the model is not satisfactory and the vertices can be identified near which the problematic regions of irrelevant systems are) fine manipulation can be applied to improve the achievable performance.

6.2 Polytopic TP Model Generation and Manipulation

For constructing an appropriate polytopic TP model with given parameter separation, the Affine TP model is determined first by using Algorithm 5.16.

Definition 6.1 (Affine TP model)**.** *The qLPV model in* (6.1) *is called an affine TP model if the system matrices are given as*

$$\mathbf{S}(\mathbf{p}) = \mathcal{S}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}), \tag{6.5}$$

where the definitions of the symbols are the same as those in Definition 5.13.

The uniqueness of the description is inherited from Theorem 5.15. Complexity (here dimensions of affine hulls) reduction can be done as in Theorem 5.17 but it must be mentioned that the theorem cannot bound the difference of the dynamics of the approximating model. Consequently, if the cut part is not only the numerical error, it is recommended to use additional robust methods taking into account the cut part as Algorithm 5.18 or as in [155].

The determination of polytopic model by taking into account only geometric properties is called here Polytopic TP Model Generation.

Corollary 6.2 (Polytopic Model Generation)**.** *The determination of vertices* $\mathbf{r}_1^{(k)}, \mathbf{r}_2^{(k)}, \ldots, \mathbf{r}_{J_k}^{(k)} \in \mathbb{R}^{D_k}$ *(*$J_k \geq D_k{+}1$*) for all* $k = 1, ..., K$*, that constructs an enclosing polytope for the image set of* $\left[v_1^{(k)}(\mathbf{p}^{(k)}) \quad \ldots \quad v_{D_k}^{(k)}(\mathbf{p}^{(k)})\right]$ *and the* $\mathbf{w}^{(k)}(\mathbf{p}^{(k)})$ *weighting functions (interpreting convex combination for all* $\mathbf{p}^{(k)}$ *) given in such a way that*

$$\mathbf{w}^{(k)}(\mathbf{p}^{(k)})\mathbf{R}^{(k)} = \mathbf{v}^{(k)}(\mathbf{p}^{(k)})$$

(as in (5.21)*).*

Then the polytopic TP model, where the parameter dependencies are separated into arbitrary groups as in (5.2)*, can be formalized with* $\mathbf{w}^{(k)}(\mathbf{p}^{(k)})$ *weighting functions and core tensor*

$$\mathcal{S} = \mathcal{S}^{aff} \mathop{\boxtimes}_{k=1}^{K} \mathbf{R}^{(k)}.$$

Related to TP Model Transformation, various methods were published for generating enclosing (simplex) polytopes SNNN, IRNO, CNO [12, 15, 17, 180]. Among these, SNNN and IRNO are not or only slightly optimized, while CNO is based on stochastic optimization. The main concerns about the CNO algorithm are its very low speed and non-deterministic operation. The geometric background of these methods are discussed in Appendix A. The classical convex hull methods in [9, 20] also can be applied, but they usually result in enclosing polytopes with too many vertices (up to infinity). The disadvantages of these methods motivated the Minimal Volume Simplex (MVS) and Non-Simplex generation methods in the following sections.

These methods see only a geometric problem, without taking into account the conservatism caused by regions of irrelevant system matrices. The achievable performance can be determined only after all of the enclosing polytope generations are obtained. If it is not satisfactory and a region of irrelevant systems around one or more vertices can be identified, the determined enclosing polytopes can be manipulated.

The manipulated polytopic TP model is defined as follows.

Corollary 6.3 (Polytopic Model Manipulation). *As manipulation of* $\mathfrak{K} \subseteq \{1,\dots,K\}$ *mode enclosing polytopes, determinate the* $\mathbf{r'}^{(k)}_1, \mathbf{r'}^{(k)}_2, \dots, \mathbf{r'}^{(k)}_{J'_k} \in \mathbb{R}^{D_k}$ *vertices* ($J'_k \geq D_k+1$) *that construct an enclosing polytope for the image set of* $\begin{bmatrix} v^{(k)}_1(\mathbf{p}^{(k)}) & \dots & v^{(k)}_{D_k}(\mathbf{p}^{(k)}) \end{bmatrix}$ *taking into account the control design experience with previous enclosing polytopes. Then obtain* $\mathbf{w'}^{(k)}(\mathbf{p}^{(k)})$ *weighting functions for all* $k \in \mathfrak{K}$ *in such a way that* $\mathbf{w'}^{(k)}(\mathbf{p}^{(k)})\mathbf{R'}^{(k)} = \mathbf{v}^{(k)}(\mathbf{p}^{(k)})$ *as in* (5.21).

Then the manipulated polytopic TP model can be written as

$$\mathbf{S}(\mathbf{p}) = \mathcal{S}^{man} \mathop{\boxtimes}_{k=1,k\notin\mathfrak{K}}^{K} \mathbf{w}^{(k)}(\mathbf{p}^{(k)}) \mathop{\boxtimes}_{k\in\mathfrak{K}} \mathbf{w'}^{(k)}(\mathbf{p}^{(k)}), \tag{6.6}$$

where

$$\mathcal{S}^{man} = \mathcal{S}^{aff} \mathop{\boxtimes}_{k=1,k\notin\mathfrak{K}}^{K} \mathbf{R}^{(k)} \mathop{\boxtimes}_{k\in\mathfrak{K}} \mathbf{R'}^{(k)}.$$

As it can be seen, the conservatism must be traced down to the enclosing polytope generations. One must choose the $\mathfrak{K} \subseteq \{1,\dots,K\}$ set of modes, and $\mathfrak{J}_k \subset \{1,\dots,J_k\}$ sets for all $k \in \mathfrak{K}$ in such a way that

- The $k \in \mathfrak{K}$-mode $j_k \in \mathfrak{J}_k$-th sub-tensors of the core tensor contain every vertex systems to be manipulated.
- For all $k \in \mathfrak{K}$, the $j_k \in \mathfrak{J}_k$-th vertex of the k-mode enclosing polytope can be close to the image set of $\mathbf{u}^{(k)}(\mathbf{p})$ functions.

The following sections provide methods for manipulation of MVS polytopes and manipulation by using cutting halfspaces.

6.3 Minimal Volume Simplex (MVS) Generation and Manipulation Methodology

The simplex enclosing polytopes have a distinguished role as methods in [12, 96, 128, 180]. Based on its algebraic properties (see Subsection 4.3.1 and 4.3.2), special methods can be defined for its generation and manipulation.

It is evident that in a $D = 1$ dimensional space, the MVS enclosing polytope is a line segment, the convex hull with $J = 2$ vertices. In higher dimensional spaces, derivation of the minimal volume enclosing simplex leads to a multivariate non-convex optimization problem usually incorporating several local minima. For these reasons, the minimal volume has only suboptimal meaning along this chapter, see [108] for more details.

At the level of underlying mathematics, a similar problem appears in material classification and detection in the field of hyperspectral signal processing [25]. Among the family of related methods, the Minimal Volume Simplex Analysis (MVSA) algorithm [3, 108] stands out in terms of computational time and efficiency [2, 42]. In general, it is not possible to find the global minimum, and in some cases, it is not unique at all. However, good sub-optimal solutions can be achieved, see [108].

In this section, the MVSA based method is revisited, and its concepts are fitted to the presented geometric interpretation, which leads to similar results as CNO, but it is a very fast (ca. 500 times faster than CNO) and deterministic method [97], which allows its systematic manipulation as well. Two types of fine manipulation approaches are introduced that make possible the tuning of the polytope along physically reasoned objectives: The Fitted-Vertex Constraints allows binding dedicated vertices to certain points within the quasi-continuous set of investigated systems, while the Closing-Vertex Constraints draw selected vertices closer to the actually considered dynamics.

6.3.1 Minimal Volume Enclosing Simplex Generation

Consider the enclosing polytope problem for the k-mode $\mathbf{v}^{(k)}(\mathbf{p}^{(k)}) = \begin{bmatrix}\mathbf{u}^{(k)}(\mathbf{p}^{(k)}) & 1\end{bmatrix}$, in general. For sake of simplicity, the (k) indices will not be denoted as $\mathbf{u}(\mathbf{p})$ in the following. The dimension of the problem will be denoted by D and the image of $\mathbf{u}(\mathbf{p})$ as

$$\mathfrak{U} = \{\mathbf{u}(\mathbf{p}) | \mathbf{p} \in \Omega\} \subset \mathbb{R}^D. \tag{6.7}$$

The aim is to find an enclosing polytope with $\mathbf{r}_1, ..., \mathbf{r}_J$ vertices, where $J = D + 1$. The key idea of the method is to use $\mathbf{Z} = \mathbf{R}^{-1}$ matrix variable instead of the vertex coordinates, resulting in the optimization problem

$$\underset{\mathbf{Z}}{\text{minimize}} \quad \mathrm{Vol}(\mathbf{R}) = 1/|\det(\mathbf{Z})| \tag{6.8}$$

subject to

$$\mathbf{w}(\mathbf{u}, \mathbf{Z}) \geq 0, \tag{6.9}$$

$$\sum_j w(\mathbf{u}, \mathbf{Z})_j = 1, \quad \forall \mathbf{u} \in \mathfrak{U}, \tag{6.10}$$

$$\text{where} \quad \mathbf{w}(\mathbf{u}, \mathbf{Z}) = \begin{bmatrix}\mathbf{u} & 1\end{bmatrix} \mathbf{Z},$$

where $\mathbf{w}(\mathbf{u}, \mathbf{Z})$ gives the weights for vertices $\mathbf{R} = \mathbf{Z}^{-1}$. The weights denote affine combinations if equation (6.10) is fulfilled and if all of the weights are not negative as inequality (6.9) requires, the vertices construct an enclosing polytope. The volume of the simplex can be computed as as $\det(\mathbf{R})/D!$. For this reason, $1/\det(\mathbf{Z}) = \det(\mathbf{R})$ is minimized.

The enclosing constraints are linear. In this way, the feasible region is convex. However, the objective function is concave. Thus only local optimization is possible.

The presented method based on MVSA consists of 3 steps. The first one decreases the computation load by reducing the $\mathfrak{U}$ set. The second one determines an initial polytope for optimization of the third step based on majorizer minimization with Sequential Quadratic Programming [29].

Algorithm 6.4. (Minimal Volume Enclosing Simplex Generation)

Step 1 (Reducing the number of points)**.** *This step reduces the number of points while taking into consideration only the outermost points of* $\mathfrak{U}$ *as those are sufficient when checking whether a polytope encloses every* $\mathbf{u} \in \mathfrak{U}$*. Convex hull algorithms (as so-called QuickHull algorithm [20]) are applied to determine the* $\mathfrak{U}^{red} \subseteq \mathfrak{U}$ *set of outer points.*

Step 2 (Determining the initial guess)**.** *The algorithm requires an initial enclosing polytope that is similarly oriented as the MVS. To obtain the initial guess, first, we determine a polytope with* $\mathbf{r}_1^{VCA}, ..., \mathbf{r}_J^{VCA}$ *(*$J = D+1$*) vertices chosen from elements of* $\mathfrak{U}^{red}$ *constructing the simplex with the largest possible volume by applying the Vertex Component Analysis (VCA) method [130].*

The initial enclosing polytope can now be determined by expanding ("inflating") the polytope that usually results in a good initial guess.

The $j = 1, ..., J$ *vertices of the expanded polytope are computed as*

$$\mathbf{r}_j^0 = \mathbf{r}^{mean} + (1+\theta_1)(1+\theta_2)(\mathbf{r}_j^{VCA} - \mathbf{r}^{mean}), \quad \textit{where} \quad \mathbf{r}^{mean} = \sum_{j=1}^{J} \mathbf{r}_j^{VCA}/J, \tag{6.11}$$

the θ_1 *is chosen to obtain an enclosing polytope*

$$\theta_1 = -J \min_{j, \mathbf{u} \in \mathfrak{U}^{red}} (\mathbf{w_u})_j, \quad \textit{where} \quad \mathbf{w_u} = \begin{bmatrix} \mathbf{u} & 1 \end{bmatrix} \mathbf{R}^{VCA^{-1}} \tag{6.12}$$

and the second expansion with a large coefficient ($3 \leq \theta_2 \leq 8$*) is chosen to ensure that the optimization will be able to change the alignment of the polytope significantly.*

Step 3 (Volume minimization)**.** *Initialize* $\mathbf{z}_0$ *as* $\mathbf{z}_0 = \operatorname{vec}(\mathbf{Z}_0)$*, where* $\mathbf{Z}_0 = \mathbf{R}^{VCA\ -1}$ *the* vec *operator is as*

$$\operatorname{vec}(\mathbf{Z}) = \operatorname{vec}(\begin{bmatrix} \mathbf{z}_1 & \dots & \mathbf{z}_J \end{bmatrix}) = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_J \end{bmatrix} = \mathbf{z}, \tag{6.13}$$

denote the vertex matrix of the actual enclosing simplex as $\mathbf{R}$*, and the corresponding variables of the optimisation as* $\mathbf{z} = \operatorname{vec}(\mathbf{Z})$*, where* $\mathbf{Z} = \mathbf{R}^{-1}$*.*

The enclosing constraints are linear and can be written as

$$(\mathbf{E}^J \otimes \begin{bmatrix}\mathbf{u} & 1\end{bmatrix})\mathbf{z} \geq \mathbf{0}^J, \quad \textit{for all } \mathbf{u} \in \mathfrak{U}^{red} \tag{6.14}$$

$$(\mathbf{1}^{1\times J} \otimes \mathbf{E}^J)\mathbf{z} = \begin{bmatrix}\mathbf{0}^D \\ 1\end{bmatrix}. \tag{6.15}$$

A majorizer function to $\log(\mathrm{Volume}(\mathbf{R}))$ *can be given as*

$$\sigma(\mathbf{z}, \mathbf{z}_0) = c + \mathbf{g}(\mathbf{z}_0) \cdot (\mathbf{z} - \mathbf{z}_0) + 0.5(\mathbf{z} - \mathbf{z}_0)^T \cdot \mathbf{H}(\mathbf{z}_0) \cdot (\mathbf{z} - \mathbf{z}_0), \tag{6.16}$$

$$\mathbf{g}(\mathbf{z}_0) = -\mathrm{vec}(\mathbf{Z}_0^{-T})^T, \quad (\mathbf{z}_0 = \mathrm{vec}(\mathbf{Z}_0)), \tag{6.17}$$

$$\mathbf{H}(\mathbf{z}_0) = \mathrm{diag}(g_1^2(\mathbf{z}_0), g_2^2(\mathbf{z}_0), \ldots, g_{J^2}^2(\mathbf{z}_0)), \tag{6.18}$$

and the value of c *is irrelevant during the computations.*

The value $\mathbf{z}$ *that minimizes the majorizer* $\sigma(\mathbf{z}, \mathbf{z}_0)$ *and fulfills the constraints can be obtained via Quadratic Programming, and an enclosing polytope of locally minimal volume can be obtained by performing it iteratively, following the scheme of the Sequential Quadratic Programming.*

From the resulting $\mathbf{z}$ *the vertex matrix* $\mathbf{R}$ *can be restored and the* $\mathbf{w}(\mathbf{p})$ *weighting functions can be computed as* $\mathbf{w}(\mathbf{p}) = \mathbf{v}(\mathbf{p})\mathbf{R}^{-1}$.

6.3.2 Minimal Volume Enclosing Simplex Manipulation

If the achievable performance with the polytopic TP model is not satisfactory and it is assumed that irrelevant regions around the vertices of the k-mode enclosing polytope are responsible for the problem, the following method can be used to manipulate the enclosing simplex.

First of all, a metrics for the relative distance of the j-th vertices and the $\mathfrak{U}$ set is defined as

$$\delta_j = 1 - \max_{\mathbf{u}\in\mathfrak{U}^{red}} (\mathbf{w_u})_j, \tag{6.19}$$

which is zero if the j-th vertex is one of the $\mathbf{u} \in \mathfrak{U}$ points. In this way, the geometry of the polytope (in higher dimensional spaces as well) can be characterized by $\delta_1, ..., \delta_J$ scalar values, which describe the overhang of the vertices.

The following algorithm comes from the previous method, but the initial polytope is the expanded MVS, and beside the enclosing constraints, it applies constraints for relative distances of the vertices that need to be closed to $\mathfrak{U}$ to reduce the irrelevant regions around them. Denote the set of indices of vertices to be manipulated as $\mathfrak{J} \subset \{1, \ldots, J\}$.

Algorithm 6.5. (MVS Enclosing Polytope Manipulation)

Step 1 (Reducing the number of points)**.** *The same as Step 1 of Algorithm 6.4.*

Step 2 (Determining the initial guess). *As in Step 2 of Algorithm 6.4, but instead of using the result of VCA, the vertices contained by*

$$\begin{bmatrix} \mathbf{r}_1^{MVS} \\ \vdots \\ \mathbf{r}_J^{MVS} \end{bmatrix} = \left(\sum_{\mathbf{u} \in \mathfrak{U}^{red}} \mathbf{w}_{\mathbf{u}}^T \mathbf{w}_{\mathbf{u}} \right)^{-1} \left(\sum_{\mathbf{u} \in \mathfrak{U}^{red}} \mathbf{w}_{\mathbf{u}}^T \mathbf{u} \right) \tag{6.20}$$

are expanded to obtain the initial guess.

Step 3 (Volume minimization). *As in Step 3 of Algorithm 6.4, but the constraints in* (6.14) *and* (6.15) *are amended with constraints for the $j \in \mathfrak{J}$ vertices:*

$$\left(\mathbf{e}_j^J \otimes [\boldsymbol{v}_j \quad 1] \right) \mathbf{z} = 1 - \delta_j \qquad \forall \ j \in \mathfrak{J}, \tag{6.21}$$

where

$$\boldsymbol{v}_j = \underset{\mathbf{u} \in \mathfrak{U}^{red}}{\operatorname{argmax}} (\mathbf{w}_{\mathbf{u}}^{MVS})_j, \tag{6.22}$$

and δ_j is the desired relative distance of the j-th vertex.

If there exists a polytope that satisfies the constraints, the resulting $\mathbf{Z}$ *matrix can be used as in Step 3. The volume of this polytope is larger than the volume of the original MVS-type enclosing polytope.*

Two possible strategies are discussed for the application of manipulation constraints.

Fitted-Vertex Constrain (FVC): Consider the constraints

$$\delta_j \approx 0 \quad \forall j \in \mathfrak{J}, \tag{6.23}$$

where some vertices are almost fitted to the corresponding $\boldsymbol{v}_j$ point in optimization (6.8). (Strict constraint could cause numerical issues during the SQP optimization while applying constraints that only make the vertices approach to $\mathfrak{U}$ points is usually more feasible.)

Closing-Vertex Constrain (CVC): Consider the constraints with values:

$$0 < \delta_j \quad \forall j \in \mathfrak{J}. \tag{6.24}$$

If $\delta_j < \delta_j^{MVS}$, the constraint moves a closer vertex to the exact convex hull, otherwise it moves the vertex farther.

This method provides further opportunity to optimize the simplex polytopes through

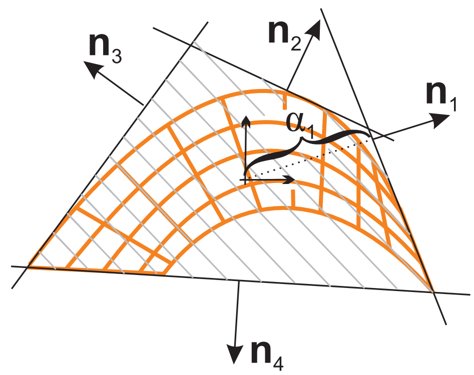


Figure 6.1: Enclosing polytope as intersection of halfspaces (given by $\mathbf{n}_i$ normals and α_i offsets)

δ_j relative distances:

$$\begin{aligned}
&\underset{\{\delta_j^{(k)}|j\in\mathfrak{J}_k,k\in\mathfrak{K}\}}{\text{maximize/minimize}} \quad \alpha \\
&\text{subject to} \\
&\quad (\mathbf{S}_1,\ldots,\mathbf{S}_r) = \text{MVS Polytopic Model applying CVC with values } (\ldots,\delta_{j_k}^{(k)},\ldots) \\
&\quad \alpha = \text{SDP}(\mathbf{S}_1,\ldots,\mathbf{S}_r)
\end{aligned} \tag{6.25}$$

In each step of the optimization, the guaranteed performance can be evaluated by performing the LMI-based design using the actual state of the polytopic model. This, in turn, leads to a cascade optimization if the $\delta_{j_k}^{(k)}$ is chosen according to, e.g., genetic algorithm.

6.4 Minimal Volume Non-Simplex Enclosing Polytope

The section provides methods for constructing non-simplex enclosing polytopes. In this case, the key idea of the MVSA method, which allows composing linear enclosing constraints, cannot be applied. For this reason, here we use another parametrization: The polytope is considered as the intersection of halfspaces, which are described with the normals ($\mathbf{n}$) and offsets (α) of the bounding hyperplanes, see Fig. 6.1. First of all, let us recall the duality of convex hull and intersection of halfspaces problems.

6.4.1 Duality of convex hull problem and the intersection of halfspaces

The subsection recalls some basic principles of convex polytopes, which will be applied in the following subsections, for more details, see [34].

Consider the set $\mathfrak{U}$ of points $\mathbf{u} \in \mathbb{R}^D$ on the D dimensional Euclidean space. The convex hull algorithms are looking for the outermost $\mathbf{r}_j \in \mathfrak{U}$, $j = 1, ..., J$ that cannot be given by the convex combination of other $\mathbf{u} \in \mathfrak{U}$ points and the vertex sets that define the facets.

The hyperplane that contains the facet (given with the vertex set $\mathbf{r}_1, ..., \mathbf{r}_L$ where $L \geq D$) can be described with its normal vector that is

$$\mathbf{n} = \text{nullspace}\left(\begin{bmatrix} \mathbf{r}_2 - \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_L - \mathbf{r}_1 \end{bmatrix}\right), \quad \text{such that } \ |\mathbf{n}| = 1 \ \ \& \ \ \mathbf{r}_1\mathbf{n} > 0 \tag{6.26}$$

and an offset $\alpha = \mathbf{r}_1\mathbf{n}$. It is easy to see that the polytope is enclosing if for all facets,

$$\alpha \geq \max_{\mathbf{u} \in \mathfrak{U}}(\mathbf{n}\mathbf{u}). \tag{6.27}$$

By assuming that the zero point is inside of the polytope, the offsets are positive $\alpha > 0$, and the halfspaces can be described with $\frac{\mathbf{n}}{\alpha}$ vectors: The included points are $\{\mathbf{u} | \frac{\mathbf{n}}{\alpha}\mathbf{u} \leq 1\}$.

With these notations, the convex hull problem (that looks for vertices and facets for a point set) can be reformulated in the following way.

Definition 6.6 (Convex hull problem)**.** *For considering the $\mathfrak{U}$ set of $\mathbf{u} \in \mathbb{R}^D$ points, find all $\mathbf{r}_j \in \mathfrak{U}$ vertices and $\frac{\mathbf{n}_f}{\alpha_f}$, $f = 1, ..., F$, hyperspaces (denote their amount with J and F, respectively) such that*

- *For all $j = 1, ..., J$, the set $\{f | \frac{\mathbf{n}_f}{\alpha_f}\mathbf{r}_j = 1\}$ has at least D elements (the vertices must be at intersection of facets).*
- *For all $f = 1, ..., F$, the set $\{j | \frac{\mathbf{n}_f}{\alpha_f}\mathbf{r}_j = 1\}$ has at least D elements (the facets must consist of vertices).*
- *For all $f = 1, ..., F$ and $\mathbf{u} \in \mathfrak{U}$, $\frac{\mathbf{n}_f}{\alpha_f}\mathbf{u} \leq 1$ (the facets must be bounding).*

And it can be computed by GiftWrapping, QuickHull, etc. algorithms [9, 20].

The intersection of halfspaces problem is to determine vertices and facets of a polytope defined as the intersection of halfspaces. The following formulation shows well its duality with the convex hull problem.

Definition 6.7 (Intersection of halfspaces problem). *For considering the $\mathfrak{H}$ set of $\frac{\mathbf{n}}{\alpha}$ hyperplanes, find all the $\frac{\mathbf{n}_f}{\alpha_f} \in \mathfrak{H}$ bounding hyperplanes and $\mathbf{r}_j$ vertices (denote their amount with F and J, respectively) such that*

- *For all $f = 1, ..., F$, the set $\{j | \frac{\mathbf{n}_f}{\alpha_f}\mathbf{r}_j = 1\}$ has at least D elements (the facets must consist of vertices).*

- *For all $j = 1, ..., J$, the set $\{f | \frac{\mathbf{n}_f}{\alpha_f}\mathbf{r}_j = 1\}$ has at least D elements (the vertices must be at intersection of facets).*

- *For all $j = 1, ..., J$ and $\frac{\mathbf{n}}{\alpha} \in \mathfrak{H}$, $\frac{\mathbf{n}}{\alpha}\mathbf{r}_j \leq 1$ (the facets must be bounding).*

Based on this duality, the convex hull algorithms can be used to determine the vertices and facets of the polytope, and furthermore, its volume and weighting functions can be obtained based on Subsection 4.3.2.

6.4.2 Local volume minimization on the normals

Consider a polytopic description

$$\mathbf{v}(\mathbf{p}) = \begin{bmatrix}\mathbf{u}(\mathbf{p}) & 1\end{bmatrix} = \mathbf{w}(\mathbf{p}) \begin{bmatrix}\mathbf{r}_1 & 1 \\ \vdots & \vdots \\ \mathbf{r}_J & 1\end{bmatrix} \tag{6.28}$$

for a k-mode weighting function of the affine TP form and denote the image set of $\mathbf{u}(\mathbf{p})$ function with $\mathfrak{U}$ and the hyperplanes of the facets with $\{\frac{\mathbf{n}_1}{\alpha_1}, ..., \frac{\mathbf{n}_F}{\alpha_F}\}$.

The following algorithm can locally minimize its volume on the normals of hyperplanes with indices $f \in \mathfrak{F} \subseteq \{1, ..., F\}$. Thus the number of irrelevant systems around the chosen facets can be minimized.

Algorithm 6.8 (Volume Minimalization of Non-Simplex Polytopes).

Step 1 (Normals to spherical coordinates). *Describe the $\mathbf{n} \in \mathbb{R}^D$ normals of $f \in \mathfrak{F}$ hyperplanes with $\boldsymbol{\varphi} \in \mathbb{R}^{D-1}$ spherical coordinates via transformation*

$$\boldsymbol{\varphi}(\mathbf{n}) = \begin{bmatrix} \arccos \frac{n_1}{\sqrt{n_1^2 + \cdots + n_D^2}} \\ \arccos \frac{n_2}{\sqrt{n_2^2 + \cdots + n_D^2}} \\ \arccos \frac{n_3}{\sqrt{n_3^2 + \cdots + n_D^2}} \\ \vdots \\ \arccos \frac{n_{D-2}}{\sqrt{n_{D-2}^2 + \cdots + n_D^2}}, \\ c(\mathbf{n}) \end{bmatrix}, \quad where \quad c(\mathbf{n}) = \begin{cases} \arccos \frac{n_{D-1}}{\sqrt{n_{D-1}^2 + n_D^2}} & if\ n_D \geq 0, \\ 2\pi - \arccos \frac{n_{D-1}}{\sqrt{n_{D-1}^2 + n_D^2}} & if\ n_D < 0. \end{cases}$$

Step 2 (Optimize on $\boldsymbol{\varphi}$ coordinates)**.** *Optimize the volume on these $\boldsymbol{\varphi}_f$ coordinate vectors*

$$\begin{aligned}
&\min_{\{\varphi_f\}_{f\in\mathfrak{F}}} \mathrm{Vol}\left\{\mathbf{u} \;\middle|\; \frac{\mathbf{n}(\boldsymbol{\varphi}_1)}{\alpha_1}\mathbf{u} \le 1,\ \frac{\mathbf{n}(\boldsymbol{\varphi}_2)}{\alpha_2}\mathbf{u} \le 1,\ ...,\ \frac{\mathbf{n}(\boldsymbol{\varphi}_F)}{\alpha_F}\mathbf{u} \le 1\right\}\\
&\textit{subject to}\quad \alpha_f = \max_{\mathbf{u}\in\mathfrak{U}}(\mathbf{n}(\boldsymbol{\varphi}_f)\cdot\mathbf{u}),
\end{aligned}$$

$$\mathbf{n}(\boldsymbol{\varphi}) = \begin{bmatrix} \cos(\varphi_1) \\ \sin(\varphi_1)\cos(\varphi_2) \\ \sin(\varphi_1)\sin(\varphi_2)\cos(\varphi_3) \\ \vdots \\ \sin(\varphi_1)\ldots\sin(\varphi_{D-2})\cos(\varphi_{D-1}) \\ \sin(\varphi_1)\ldots\sin(\varphi_{D-1}) \end{bmatrix}.$$

During the optimization, the change of the topology as the hyperplanes roll on the $\mathfrak{U}$ set is allowed. (See Fig. 6.1.)

Step 3 (Reconstruct enclosing polytope)**.** *From the resulting normals, reconstruct the vertices of the enclosing polytope and the weighting functions for the polytopic description.*

6.4.3 Cut off regions by additional halfspaces

Consider the k-mode weighting function

$$\mathbf{v}(\mathbf{p}) = \begin{bmatrix}\mathbf{u}(\mathbf{p}) & 1\end{bmatrix}, \quad \mathbf{p} \in \Omega \tag{6.29}$$

of the affine TP form, and denote the image set of $\mathbf{u}(\mathbf{p})$ function with $\mathfrak{U}$ to be enclosed by the polytope.

Assume that there is an affine subspace given as $\mathfrak{E}$ affine hull of $\mathbf{q}_1, \mathbf{q}_2, ...$ points that must not be enclosed, and furthermore, it should be as far from the polytope as possible. If it is a given point (for example, a vertex from the previous polytopic description), this affine subspace is zero-dimensional; if there is a line/plane, etc. that represents non-stabilizable irrelevant systems, the affine subspace is 1, 2, etc. dimensional.

Here we show how to determine an additional halfspace to achieve this goal.

Algorithm 6.9. *(Additional halfspace to exclude an affine subspace)*

Step 1. *(Orthogonal complemental space for normals) Denote the set of vectors orthogonal to $\mathfrak{E}$ as $\mathfrak{O}$, which is the orthogonal complement of $\mathfrak{E}$. It can be obtained as the* *null space of matrix* $\begin{bmatrix} \mathbf{q}_2 - \mathbf{q}_1 \\ \mathbf{q}_3 - \mathbf{q}_1 \\ \vdots \end{bmatrix}$ *and denote its dimension by* O *and an orthonormal basis for it by* $(\mathbf{n}_1, ..., \mathbf{n}_O)$.

Then the normal of a halfspace for excluding it, can be written as $\mathbf{n} = \sum_{o=1}^{O} \eta_o \mathbf{n}_o$, *where* $\sum_{o=1}^{O} \eta_o^2 = 1$.

With a given $\mathbf{n}$, *the offset of the touching halfspace can be computed as* $\alpha(\mathbf{n}) = \max_{\mathbf{u}\in\mathfrak{U}}(\mathbf{n} \cdot \mathbf{u})$.

The goal is to determine a halfspace that cuts the surrounding of $\mathfrak{E}$ *with as large radius as possible - without cutting off any* $\mathbf{u} \in \mathfrak{U}$ *point. The possible cut radius with a given* $\mathbf{n}$ *to be maximized can be written as* $r(\mathbf{n}) = \mathbf{n}^T \mathbf{q}_1 - \alpha(\mathbf{n})$.

Step 2. *(Initial normal) For the optimization here, an initial normal vector is computed.*

First, project $\mathbf{q}_1$ *to* $\mathfrak{O}$ *as* $\mathbf{n}^{(0)} = \sum_o \psi_o \mathbf{n}_o$, *where* $\nu_o = \mathbf{q}_1 \cdot \mathbf{n}_o$, *for* $o = 1, ..., O$, *and* $\boldsymbol{\psi} = \frac{\boldsymbol{\nu}}{|\nu|}$. *Then, describe it by spherical coordinates:*

$$\boldsymbol{\phi}^{(0)} = \begin{bmatrix} \arccos \frac{\psi_1}{\sqrt{\psi_1^2 + \cdots + \psi_O^2}} \\ \arccos \frac{\psi_2}{\sqrt{\psi_2^2 + \cdots + \psi_O^2}} \\ \arccos \frac{\psi_3}{\sqrt{\psi_3^2 + \cdots + \psi_O^2}} \\ \vdots \\ \arccos \frac{\psi_{O-2}}{\sqrt{\psi_{O-2}^2 + \cdots + \psi_O^2}}, \\ c(\boldsymbol{\psi}) \end{bmatrix}, \quad where \quad c(\boldsymbol{\psi}) = \begin{cases} \arccos \frac{\psi_{O-1}}{\sqrt{\psi_{O-1}^2 + \psi_O^2}} & if \ \psi_O \geq 0, \\ 2\pi - \arccos \frac{\psi_{O-1}}{\sqrt{\psi_{O-1}^2 + \psi_O^2}} & if \ \psi_O < 0. \end{cases}$$

Step 3. *(Halfspace optimalization) Now optimize the direction of* $\mathbf{n}$ *via the* $\boldsymbol{\phi}$ *sperical coordinates to maximize the* r *radius:*

$$\begin{aligned} \max_{\boldsymbol{\phi}} \ & r(\boldsymbol{\phi}) = \mathbf{n}(\boldsymbol{\phi})^T \mathbf{q}_1 - \alpha(\boldsymbol{\phi}) \\ \textit{subject to} \quad & \alpha(\boldsymbol{\phi}) = \max_{\mathbf{u}\in\mathfrak{U}}(\mathbf{n}(\boldsymbol{\phi}) \cdot \mathbf{u}), \\ & \mathbf{n}(\boldsymbol{\phi}) = \sum_{o=1}^{O} \eta_o(\boldsymbol{\phi}) \mathbf{n}_o, \\ & \boldsymbol{\eta}(\boldsymbol{\phi}) = \begin{bmatrix} \cos(\phi_1) \\ \sin(\phi_1)\cos(\phi_2) \\ \sin(\phi_1)\sin(\phi_2)\cos(\phi_3) \\ \vdots \\ \sin(\phi_1)\ldots\sin(\phi_{O-2})\cos(\phi_{O-1}) \\ \sin(\phi_1)\ldots\sin(\phi_{O-1}) \end{bmatrix}. \end{aligned}$$

The new enclosing polytope, which is the intersection of the original polytope and the derived cutting halfspaces, can be obtained via convex hull algorithms exhausting the duality explained in Subsection 6.4.1. Furthermore, its geometry can be refined via local volume optimization (see Algorithm 6.8) as well.

6.5 Summary

The chapter showed that the enclosing polytope generation is crucial at control oriented applications of Affine TP Model Transformation. Because of its complexity, it can be performed in two phase: first a generation and following that, based on the experiences, manipulation of the generated enclosing polytopes.

For these goals, two approaches were proposed:

1. The MVS approach provides enclosing simplices with (near) minimum volume and allows its manipulation via simple constraints.

2. The MVNS approach gives the opportunity to locally minimize the volume of a non-simplex polytope by considering the polytope as the intersection of halfspaces. Manipulation can be performed via cutting halfspaces to exclude the problematic regions. For its determination, a method was proposed as well.

6.6 Proofs

Proof of Algorithm: 6.4.

For Step 2. *By expanding with θ_1 the polytope becomes enclosing, because denoting the new weighting functions of $\mathbf{u}$ with $\mathbf{w}'_\mathbf{u}$, the following statements are true for the new and old weighting functions and vertices:*

$$\mathbf{w}_\mathbf{u}\begin{bmatrix}\mathbf{r}_j^{VCA} & 1\end{bmatrix}_j = \begin{bmatrix}\mathbf{u} & 1\end{bmatrix} \tag{6.30}$$

$$\mathbf{w}'_\mathbf{u}\begin{bmatrix}-\theta_1\mathbf{r}_{mean} + (1+\theta_1)\mathbf{r}_j^{VCA} & 1\end{bmatrix}_j = \begin{bmatrix}\mathbf{u} & 1\end{bmatrix} \tag{6.31}$$

this way

$$\mathbf{r}_j^{VCA}\left(\mathbf{w}_\mathbf{u} - (1+\theta_1)\mathbf{w}'_\mathbf{u} + \mathbf{1}^J\frac{\theta_1}{J}\right) = 0 \tag{6.32}$$

so

$$\mathbf{w}'_\mathbf{u} = \frac{\mathbf{w}_\mathbf{u} + \mathbf{1}^J\theta_1/J}{1+\theta_1}. \tag{6.33}$$

It is easy to see, that by choosing $\theta_1 = -J \min_{j,\mathbf{u}\in\mathfrak{U}^{red}} \mathbf{w_u}$

$$\min_{j,\mathbf{u}\in\mathfrak{U}^{red}} \mathbf{w}'_\mathbf{u} = \frac{\min_{j,\mathbf{u}\in\mathfrak{U}^{red}} \mathbf{w_u} - \min_{j,\mathbf{u}\in\mathfrak{U}^{red}} \mathbf{w_u}}{\theta_1 + 1} = 0, \tag{6.34}$$

$$\sum_{j=1}^{J} w'_{\mathbf{u},j} = \frac{1+\theta_1}{\theta_1 + 1} = 1. \tag{6.35}$$

For Step 3. *Based on* $\mathrm{vec}(\mathbf{AZB}) = (\mathbf{B}^T \otimes \mathbf{A})\mathbf{z}$ *the constraints can be written as*

$$\mathbf{w_u} = \begin{bmatrix}\mathbf{u} & 1\end{bmatrix} \mathbf{Z} \geq \mathbf{0}^J \quad \textit{for all } \mathbf{u} \in \mathfrak{U}, \tag{6.36}$$

$$\mathbf{Z}\mathbf{1}^J = \begin{bmatrix}\mathbf{0}^D \\ 1\end{bmatrix} \tag{6.37}$$

and then

$$\sum_{j=1}^{J} w_{\mathbf{u},j} = \mathbf{w_u}\mathbf{1} = \begin{bmatrix}\mathbf{u} & 1\end{bmatrix} \mathbf{Z}\mathbf{1}^J = \begin{bmatrix}\mathbf{u} & 1\end{bmatrix} \begin{bmatrix}\mathbf{0}^D \\ 1\end{bmatrix} = 1 \tag{6.38}$$

this way, they describe the enclosing constraints, furthermore, the majorizer function can be described as

$$\sigma(\mathbf{Z}, \mathbf{Z}_0) = f(\mathbf{Z}_0) + \sum_{i,j} \frac{\partial f(\mathbf{Z})}{\partial Z_{ij}}\Big|_{\mathbf{Z}_0} (Z_{ij} - Z_{0,ij}) + \frac{1}{2}\sum_{i,j} \left(\frac{\partial f(\mathbf{Z})}{\partial Z_{ij}}\Big|_{\mathbf{Z}_0}\right)^2 (Z_{ij} - Z_{0,ij})^2,$$

where the object function comes from

$$f(\mathbf{Z}) = \log(\mathrm{Vol}(\mathbf{r}_1, .., \mathbf{r}_J)) = \log|\det \mathbf{Z}^{-1}| = -\log|\det \mathbf{Z}|, \tag{6.39}$$

this way its derivative can be computed as

$$\frac{\partial f(\mathbf{Z})}{\partial Z_{ij}}\Big|_{\mathbf{Z}_0} = -\left(\mathbf{Z}_0{}^{-1}\right)_{ji}. \blacksquare$$

Proof of Algorithm 6.5.

For Step 2 *Because the affine hull of* $\mathfrak{U} \subset \mathbb{R}^D$ *is* D *dimensional, the affine hull of* $\mathfrak{U}^{red}$ *is* D *dimensional as well and* $\left(\sum_{\mathbf{u}\in\mathfrak{U}^{red}} \mathbf{w}_\mathbf{u}^T \mathbf{w_u}\right)$ *is invertible.*

Furthermore, because

$$\mathbf{w_u} \begin{bmatrix}\mathbf{r}_1^{MVS} \\ \vdots \\ \mathbf{r}_J^{MVS}\end{bmatrix} = \mathbf{u} \quad \forall \mathbf{u} \in \mathfrak{U}^{red}, \tag{6.40}$$

the equation

$$\sum_{\mathbf{u}\in\mathfrak{U}^{red}} \mathbf{w}_{\mathbf{u}}^T \mathbf{w}_{\mathbf{u}} \begin{bmatrix} \mathbf{r}_1^{MVS} \\ \vdots \\ \mathbf{r}_J^{MVS} \end{bmatrix} = \sum_{\mathbf{u}\in\mathfrak{U}^{red}} \mathbf{w}_{\mathbf{u}}^T \mathbf{u} \tag{6.41}$$

holds, and the vertices can be computed via equation (6.20).

For Step 3 *It is easy to see that* $\boldsymbol{v}_j \in \mathfrak{U}^{red}$ *is a point with* $(\mathbf{w}_{\boldsymbol{v}_j})_j = 1 - \delta_j$. *The left side of constraint can be written as*

$$\left(\mathbf{e}_j^J \otimes [\boldsymbol{v}_j \quad 1]\right) \mathbf{z} = \mathrm{vec}\left([\boldsymbol{v}_j \quad 1]\, \mathbf{Z}\mathbf{e}_j^J\right) = \mathrm{vec}\left(\mathbf{w}_{\boldsymbol{v}_j} \mathbf{e}_j^J\right) = (\mathbf{w}_{\boldsymbol{v}_j})_j \tag{6.42}$$

By describing the constraint $\delta_{j,resulted} \leq \delta_{j,described}$ *is guaranteed.* ■

Chapter 7

Generalization of Polytopic TP Model-based Controller Design

This chapter renews the concept of Polytopic Tensor Product (TP) Model-based control analysis and synthesis by generalizing the use of TP-structured variables in the definite conditions of the applied control criteria.

The variables used in the controller candidate, the Lyapunov-function or the slack variables can be defined in TP structure with arbitrarily chosen multiplicities, based on its extension in Section 5.2. This way, their parameter dependencies can be disabled or enabled (theoretically with arbitrarily high complexity) by choosing appropriate multiplicities. (See papers [69, 103, 104] for the relevance of Lyapunov-function and controller candidates on multiple summations.)

Based on Subsection 5.2.1, a definite condition constructed from polytopic TP forms can be written as a definite condition on a TP form and this chapter points out that via a recursive method, sufficient (and asymptotically necessary) Matrix Inequalities can be derived (see [52, 84, 150, 178]). These definite conditions can be Linear Matrix Inequalities (LMIs) or Bilinear Matrix Inequalities (BMIs), etc. according to the design method in consideration, showing the potential of the relaxed TP form during the control design.

The chapter is structured as follows: First Section 7.1 formalizes the use of variables in TP structure during the controller design. Then Section 7.2 shows how the definite conditions can be extracted to LMIs/BMIs, etc. Following that, Section 7.3 shows its application for state feedback controller design according to H_∞, H_2, pole placement constraints. Finally, Section **??** summarizes the results.

7.1 Polytopic TP forms in control analysis and synthesis criteria

First of all, let us recall that the relaxed definition of Polytopic TP form (see Definition 5.8) does not require the parameter sets to be disjoint, allowing the parameters to be considered multiple times. Thus, multiple summations can be described with a compact notation.

Now, consider a (q)LPV system described with polytopic TP structure where the parameters are grouped into $\mathbf{p}^{(l)}$ vectors ($l = 1, ..., L$) usually with single multiplicities:

$$\mathbf{S}(\mathbf{p}) = \mathcal{S} \mathop{\boxtimes}_{l=1}^{L} \mathbf{w}^{(l)}(\mathbf{p}^{(l)}). \tag{7.1}$$

Then consider a control criteria based on definite conditions (as Lemma 2.1, 2.2, 2.3 and 2.4), and define the unknown variables (in Lyapunov-function, controller/observer candidates and/or slack variables) in TP structures as

$$\mathbf{X}(\mathbf{p}) = \mathcal{X} \mathop{\boxtimes}_{k=1}^{K(\mathbf{x})} \mathbf{w}^{(l(l,\mathbf{x}))}(\mathbf{p}^{(l(k,\mathbf{x}))}), \tag{7.2}$$

where $\mathbf{x}$ denotes the multiplicities.

Then the definite criteria can be easily rewritten into definite conditions on a Polytopic TP form as

$$\mathcal{G} \mathop{\boxtimes}_{k=1}^{K(\mathbf{g})} \mathbf{w}^{(l(l,\mathbf{g}))}(\mathbf{p}^{(l(k,\mathbf{g}))}) \prec 0 \tag{7.3}$$

based on Subsection 5.2.1. The next section shows how they can be extracted to definite conditions on matrices, which can be LMIs or BMIs depending on the structure of the definite condition.

Remark 7.1. *Similar structures appear in [107], where the used variables in the controller candidate, the Lyapunov-function, and the slack variables can depend on the delayed values of the parameters with arbitrary multiplicities to relax the criteria.*

7.2 Definite conditions in Polytopic TP form

Papers [61, 84, 118, 150, 178, 183] introduced approaches to extract multiple polytopic summation. Because these approaches were published as part of control design methods, Appendix B summarizes them applying a common notation system. In the following, we will refer them to as extraction methods, see the following definition.

Definition 7.2 (Extraction of multiple polytopic sums)**.**
Consider the definite condition

$$\sum_{j_1=1}^{J} \cdots \sum_{j_M=1}^{J} w_{j_1}(\mathbf{x}) \dots w_{j_M}(\mathbf{x}) \mathbf{G}^{(j_1,\dots,j_M)}(\mathbf{q}) \succ 0 \tag{7.4}$$

is fulfilled for all $\mathbf{x} \in X$, $\mathbf{q} \in Q$, *where*

- $\mathbf{G}^{(j_1,\dots,j_M)}(\cdot)$ *are functions* $Q \to \mathbb{G}$ *and* $\mathbb{G}$ *is the space of symmetric matrices on real numbers with appropriate sizes,*

– *the* $w_j(\cdot)$ *functions denote convex combinations between* J *vertices as* $\sum_{j=1}^{J} w_j(\mathbf{x}) = 1, \quad w_j(\mathbf{x}) \geq 0 \ \forall j = 1, ..., J, \ \mathbf{x} \in X$,

– M *stands for multiplicity of the convex combinations.*

An extraction method to guarantee condition (7.4) *provides definite conditions*

$$\mathbf{F}^{(a)}(\mathbf{q}) \equiv \sum_{j_1=1}^{J} \cdots \sum_{j_M=1}^{J} \alpha^{(a,j_1,\dots,j_M)} \mathbf{G}^{(j_1,\dots,j_M)}(\mathbf{q}) + \mathbf{Y}^{(a)}(\mathbf{q}) \succ 0 \quad \forall \mathbf{q} \in Q \tag{7.5}$$

$a = 1, ..., A$, *and the* A *number of conditions depend on the applied method, values* J *and* M, *and* $\mathbf{Y}^{(a)}(\mathbf{q})$ *stands for the possibly introduced slack matrices and* $\alpha^{(a,j_1,\dots,j_M)}$ *are given constants.*

Now, the following recursive method can be formulated to derive Matrix Inequalities from definite conditions in Polytopic TP form as (7.3).

Algorithm 7.3. *(Extraction of Polytopic TP definite conditions). Denote the problem as*

$$\mathbf{X}(\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(X)}) = \mathcal{X} \mathop{\boxtimes}_{k=1}^{K(\mathbf{x})} \mathbf{w}^{(l(k,\mathbf{x}))}(\mathbf{p}^{(l(k,\mathbf{x}))}) \succ 0, \tag{7.6}$$

where

– X *denotes the number of parameter dependencies, and vector* $\mathbf{x} \in \mathbb{N}^X$ *the multiplicities,*

– $\mathbb{G}$ *is the space of symmetric matrices on real numbers with appropriate sizes,*

– *the core tensor is on* $\mathbb{G}$ *with appropriate sizes, as* $\mathcal{X} \in \mathbb{G}^{\overbrace{J_1 \times \cdots \times J_1}^{x_1} \times \cdots \times \overbrace{J_X \times \cdots \times J_X}^{x_X}}$.

Step 1 (Split). *Split the last parameter dependency* ($\mathbf{p}^{(X)}$) *syntactically from the TP form, by applying the following notations:*

– *The split multiplicity* $M \equiv x_X$, *and number of vertices* $J \equiv J_X$.

– *The remaining multiplicities* $\mathbf{y} \equiv \left[x_1 \dots x_{X-1}\right]$, $\beta \equiv \sum_i y_i$, *and their number* $Y \equiv X - 1$.

– *For the subtensors of* $\mathcal{X}$ *core tensor*

$$\mathcal{Y}^{(\alpha_1,\dots,\alpha_M)} \equiv \mathcal{X}_{j_{\beta+1}=\alpha_1,\dots,j_{\beta+M}=\alpha_M}, \tag{7.7}$$

and functions based on them

$$\mathbf{Y}^{(\alpha_1,\dots,\alpha_M)}(\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(Y)}) \equiv \mathcal{Y}^{(\alpha_1,\dots,\alpha_M)} \mathop{\boxtimes}_{k=1}^{K(\mathbf{y})} \mathbf{w}^{(l(k,\mathbf{y}))}(\mathbf{p}^{(l(k,\mathbf{y}))}). \tag{7.8}$$

Then the function can be written as

$$\mathbf{X}(\mathbf{p}) = \sum_{\alpha_1=1}^{J} \cdots \sum_{\alpha_M=1}^{J} w_{\alpha_1}^{(X)}(\mathbf{p}^{(X)}) \dots w_{\alpha_M}^{(X)}(\mathbf{p}^{(X)}) \mathbf{Y}^{(\alpha_1,\dots,\alpha_M)}(\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(Y)}). \qquad (7.9)$$

Step 2 (Extraction)**.** *Apply an extraction method (see the methods detailed in Appendix B that were first applied by [61, 84, 118, 150, 169, 183, 178]), which results in*

$$\mathbf{F}^{(a)}(\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(Y)}) \succ 0 \qquad\qquad a = 1, ..., A, \qquad (7.10)$$

where the value A depends on the applied method, the number of vertices J and the multiplicities M, and which are definite conditions in Polytopic TP forms again, but with $Y = X - 1$ parameter dependencies.

Step 3 (Recursive call)**.** *If $Y > 0$, Algorithm 7.3 can be applied on conditions* (7.10) *recursively. Otherwise they are already matrix inequalities (LMIs/BMIs according to the affine/biaffine structure of the elements of the original core tensor).*

Remark 7.4. *The parameter dependencies can be split and extracted in an arbitrary order. In Step 1 of Algorithm 7.3, the last ones are chosen for syntactical reasons.*

Remark 7.5 (Conservativeness)**.** *Although the conditions can be "asymptotically necessary" for the parameter sets, if there are parameters in more sets, from the viewpoint of the method, they are different ones, which can cause conservativeness. For example, if $\mathbf{p}^{(1)} = \begin{bmatrix} p_1 & p_2 \end{bmatrix}$ and $\mathbf{p}^{(2)} = \begin{bmatrix} p_1 & p_3 \end{bmatrix}$, the method sees $\mathbf{p}^{(1)} = \begin{bmatrix} p_{1a} & p_2 \end{bmatrix}$ and $\mathbf{p}^{(2)} = \begin{bmatrix} p_{1b} & p_3 \end{bmatrix}$ and it gives guarantee for all $p_{1a}, p_{1b} \in [\underline{p}_1, \overline{p}_1]$, $p_2 \in [\underline{p}_2, \overline{p}_2]$ and $p_3 \in [\underline{p}_3, \overline{p}_3]$.*

7.3 Application in state feedback controller design

Consider an LPV/qLPV model in the form (2.6) or (2.17). Construct a polytopic TP model in the form (7.1) separating the measured/unknown and constant/varying parameters by defining appropriate parameter groups.

Choose control criteria given as a set of definite conditions and define the unknown functions in polytopic TP form with appropriately chosen multiplicities. For example, for Lemma 2.1, 2.2, 2.3 and 2.4, define the variables as:

$$\mathbf{X}(\mathbf{p}) = \mathcal{X} \mathop{\boxtimes}_{k=1}^{K(\mathbf{M}^L)} \mathbf{w}^{(l(k,\mathbf{M}^L))}(\mathbf{p}^{(l(k,\mathbf{M}^L))}), \qquad (7.11)$$

$$\mathbf{M}(\mathbf{p}) = \mathcal{M} \mathop{\boxtimes}_{k=1}^{K(\mathbf{M}^C)} \mathbf{w}^{(l(k,\mathbf{M}^C))}(\mathbf{p}^{(l(k,\mathbf{M}^C))}), \qquad (7.12)$$

$$\mathbf{R}(\mathbf{p}) = \mathcal{R} \mathop{\boxtimes}_{k=1}^{K(\mathbf{M}^R)} \mathbf{w}^{(l(k,\mathbf{M}^R))}(\mathbf{p}^{(l(k,\mathbf{M}^R))}), \qquad (7.13)$$

where

– the $\mathcal{X}$ tensor has appropriate sizes and it is on symmetric matrices with sizes $n \times n$,

– the $\mathcal{M}$ tensor has appropriate sizes and it is on matrices with sizes $m_u \times n$,

– the controller depends on the parameter sets, which appears in the $\mathbf{X}(\mathbf{p})$ or $\mathbf{M}(\mathbf{p})$ matrices, which implies, these parameters must be measurable or estimable,

– furthermore the parameter sets that appear in the Lyapunov-function and so in the $\mathbf{X}(\mathbf{p})$ matrix must be constant or approximately constant, which can be concluded as

$$M_k^C = 0 \text{ if } \mathbf{p}^{(k)} \text{is not measured/estimated,} \tag{7.14}$$

$$M_k^C \geq 0 \text{ otherwise,} \tag{7.15}$$

and

$$M_k^L = 0 \text{ if } \mathbf{p}^{(k)} \text{ is not measured/estimated or it is not constant,} \tag{7.16}$$

$$M_k^L \geq 0 \text{ otherwise,} \tag{7.17}$$

– and let us recall the corresponding functions from Notation 5.9

$$K(\mathbf{M}) = \sum_i M_i, \tag{7.18}$$

$$l(k, \mathbf{M}) = i \quad \text{where} \quad \sum_{a=1}^{i-1} M_a < k \leq \sum_{a=1}^{i} M_a. \tag{7.19}$$

By substituting the TP forms into the definite conditions, they can be written as definite conditions of polytopic TP forms based on Subsection 5.2.1, which can be extracted to LMIs via Algorithm 7.3.

Theorem 7.6. *Consider an LPV/qLPV model* (2.6)-(2.17) *and a set of control criteria given as definite conditions consisting affine dependency from the unknown functions.*

By constructing polytopic TP forms for the LPV/qLPV, and defining the variables in polytopic TP forms with appropriate multiplicities, they can be written as LMIs, and in this way, the special properties of the parameters can be taken into account.

The polytopic descriptions can be iteratively improved taking into account the achievable control performance with the current description.

7.4 Summary

The presented methods show that the renewed Tensor Product Model-based control analysis and synthesis concept provides a general abstraction for various existing approaches. Using the proposed formalisms, they can be applied independently for each parameter set according to their practical properties. Based on the introduced tensor product notations, very complex structures can be represented in a compact form.

Part III

Practical merit

Chapter 8

Inverted pendulum

This chapter presents a detailed numerical example of control design for the inverted pendulum. It illustrates the role of the proposed methods and their efficiency. First, the qLPV model is derived, where the non-linearity and parameter dependency of the model is decreased by applying feedback linearization. By assuming that one parameter is not exactly known, it also becomes a parameter of the qLPV model. For the sake of brevity, we apply here only the H_∞ state feedback design without applying frequency filters.

Then the Affine TP model is determined, showing that different approaches to numerically reconstruct it lead to the same results. Then the opportunities of H_∞ control design are presented on the MVS polytopic model. Following that, the multiplicities of the TP structures within the controller is also investigated.

Finally, examples are given to Polytopic TP Model Manipulation to check if the achievable performance can be made better.

8.1 Mechanical model and qLPV modeling

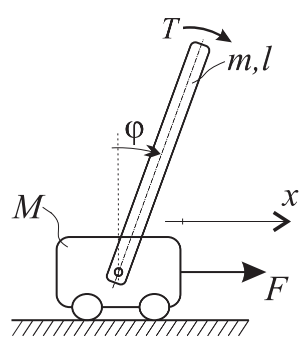


Figure 8.1: Inverted pendulum model

Consider the mechanical model of an inverted pendulum depicted in Figure 8.1. The corresponding equations of motions are written as

$$\ddot{\varphi} = \frac{6\frac{g}{l}(m+M) - 3m\dot{\varphi}^2\cos\varphi}{D}\sin\varphi - \frac{6\cos\varphi}{l\cdot D}F(t) + \frac{12(m+M)}{ml^2 D}T(t) \tag{8.1}$$

$$\ddot{x} = \frac{m(2\dot{\varphi}^2 l - 3g\cos\varphi)}{D}\sin\varphi + \frac{4}{D}F(t) - \frac{6\cos\varphi}{l\cdot D}T(t), \tag{8.2}$$

where $D = D(\varphi, m, M) = 4M + m(1 + 3\sin^2\varphi)$ and the following parameter values are considered: $m = 0.1[kg]$, $M_0 = 1.0[kg]$, $l = 0.3[m]$, $M \in [0.9, 1.3]\,[kg]$, $|\varphi| \in \left[0, \frac{5}{18}\pi\right]\,[deg]$.

By applying the feedback linearization

$$F(t) = \frac{l_0 D_0}{6\cos\varphi}\left[\frac{6\frac{g}{l_0}(M_0+m_0) - 3m_0\dot{\varphi}^2\cos\varphi}{D_0}\sin\varphi - u(t)\right] \tag{8.3}$$

with nominal m_0, l_0 and M_0 values and by assuming that $m_0 = m$ and $l_0 = l$ are exactly known, the equations of motions can be written as

$$\ddot{\varphi} = 6g\frac{M-M_0}{l}\frac{\sin\varphi}{D} + \frac{D_0}{D}u(t) + \frac{2}{ml\cos\varphi}w(t), \tag{8.4}$$

$$\ddot{x} = g\frac{D_0}{D}\tan\varphi - \frac{D_0}{D}\frac{2l}{3\cos\varphi}u(t) - \frac{1}{M+m}w(t), \tag{8.5}$$

where $D_0 = D_0(\varphi) = 4M_0 + m_0(1 + 3\sin^2\varphi)$ and $w(t) = 6\cos\varphi\dfrac{M+m}{l\cdot D(\varphi)}T(t)$.

Then the following qLPV model can be constructed

$$\dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} 0 & 6g\dfrac{M-M_0}{l\cdot D}\dfrac{\sin\varphi}{\varphi} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & g\dfrac{D_0}{D}\dfrac{\tan\varphi}{\varphi} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{A}(\varphi,M)}\mathbf{x}(t) + \underbrace{\begin{bmatrix} \dfrac{D_0}{D} \\ 0 \\ \dfrac{-2}{3\cos\varphi}\dfrac{D_0}{D} \\ 0 \end{bmatrix}}_{\mathbf{B}_u(\varphi,M)}u(t) + \underbrace{\begin{bmatrix} \dfrac{2}{ml\cos\varphi} \\ 0 \\ \dfrac{-1}{M+m} \\ 0 \end{bmatrix}}_{\mathbf{B}_w(\varphi,M)}w(t), \tag{8.6}$$

where the state variables are $\mathbf{x}(t) = \begin{bmatrix}\dot{\varphi} & \varphi & \dot{x} & x\end{bmatrix}^T$.

8.2 Control goals

Assume that the state variables are measured, and only state feedback design is considered. The primary objective of the controller is to minimize the effect of disturbance torque $T(t)$ to the pendulum and car. For this reason, the H_∞ norm is used to describe the system performance, and the goal of controller design is to minimize it via the Bounded Real Lemma by applying quadratic Lyapunov-function candidate.

For the sake of simplicity, we consider here only such a $\mathbf{z}(t)$ performance signal that can be expressed as $\mathbf{z}(t) = \mathbf{C}\mathbf{x}(t)$ without applying frequency filters or considering noises.

If the model is described in polytopic form, the following LMI criteria can be used for controller design.

Method 8.1. *Consider the following SDP:*

$$\begin{aligned}
&\min_{\mathbf{X}(\mathbf{p}),\mathbf{M}(\mathbf{p})} \gamma_\infty \\
&\quad s.t.\quad \mathbf{X}(\mathbf{p}) \succ 0, \quad \dot{\mathbf{X}}(\mathbf{p}) = 0, \\
&\qquad \begin{bmatrix} -\mathrm{Sym}(\mathbf{A}(\mathbf{p})\mathbf{X}(\mathbf{p}) + \mathbf{B}_u(\mathbf{p})\mathbf{M}(\mathbf{p})) & \mathbf{B}_w(\mathbf{p}) & (\mathbf{C}\mathbf{X}(\mathbf{p}))^T \\ * & \gamma_\infty & \mathbf{0} \\ * & * & \gamma_\infty \mathbf{E} \end{bmatrix} \succ 0,
\end{aligned}$$

where $\mathbf{X}(\mathbf{p})$ and $\mathbf{M}(\mathbf{p})$ are TP functions with the same polytopic structure and with given multiplicities. If the above SDP is solvable, then with the resulting $u(t) = \mathbf{M}(\mathbf{p})\mathbf{X}(\mathbf{p})^{-1}\mathbf{x}(t)$ controller, the condition $\left|\left|\frac{\mathbf{C}\mathbf{x}(t)}{w(t)}\right|\right|_\infty < \gamma_\infty$ (assuming that $||w(t)||_2$ does exist and it is bounded) holds for all trajectories

$$\mathbf{p}(\cdot) \in \{\mathbb{R}^+ \to \mathbb{R}^N | \mathbf{p}(t) \in \Omega,\ \dot{\mathbf{p}}(t) \in \Omega'\ \forall t \in \mathbb{R}^+\}.$$

8.3 Affine Tensor Product Model Transformation

Parameter sets to be separated

The qLPV model (8.6) depends on two parameters: φ is a state variable, which is measured and can be used in the controller. The parameter M is constant or (quasi) constant, and it may be known via measurement of the load or by estimating it from input-output characteristics.

Their different nature motivates to separate their dependencies, so $p^{(1)} = |\varphi|$ and $p^{(2)} = M$. The separation can be easily done except the $\frac{D_0}{D} = \frac{4M_0+m_0(1+3\sin^2(\varphi))}{4M+m(1+3\sin^2(\varphi))}$ term, because expressions like $1/(p_1+p_2)$ cannot be described into a separated form like $\sum_{n=1}^N f_n(p_1)g_n(p_2)$ with finite N. (Theoretically it would be infinite dimensional, practically a 4 or five dimensional description would be a good approximation in this case.) But it can be approximated by $\frac{M_0+m}{M+m}$, which depends only on parameter M.

Because the definition of TP form was extended, the third parameter set $\mathbf{p}^{(3)} = \begin{bmatrix}|\varphi| & M\end{bmatrix}$ can be applied to carry this non-separable dependency. By taking into account the cost in conservativeness (see Remark 7.5), its variance must be as low as possible. Because $\frac{D_0}{D}$ can be well approximated by $\frac{M_0+m}{M+m}$, it will be described as

$$\frac{D_0}{D} = \left(\frac{D_0}{D} \Big/ \frac{M_0+m}{M+m}\right) \cdot \frac{M_0+m}{M+m}, \tag{8.7}$$

where the $\frac{D_0}{D}/\frac{M_0+m}{M+m}$ term is a non-separable, but near constant function.

Initial TP model

The extension of the polytopic TP form allows us to derive exact TP description with low affine dimensions by defining a third parameter set for D_0/D terms. The initial TP form can be constructed manually: The initial weighting functions are chosen as

$$\begin{aligned}
&\alpha_1^{(1)} = \frac{\sin\varphi}{D_0\varphi},\ \alpha_2^{(1)} = \frac{\tan\varphi}{\varphi},\ \alpha_3^{(1)} = \frac{1}{\cos\varphi},\ \alpha_4^{(1)} = 1,\\
&\alpha_1^{(2)} = \frac{M_0+m}{M+m},\ \alpha_2^{(2)} = 1, \qquad \alpha_1^{(3)} = \frac{D_0}{D}\frac{M+m}{M_0+m},\ \alpha_2^{(3)} = 1.
\end{aligned}$$

Then the qLPV model can be written as

$$\dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} 0 & 6g\frac{M_0+m}{l}\alpha_1^{(1)}(\alpha_2^{(2)}-\alpha_1^{(2)})\alpha_1^{(3)} & 0 & 0 & \alpha_4^{(1)}\alpha_1^{(2)}\alpha_1^{(3)} & \frac{2}{ml}\alpha_3^{(1)}\alpha_2^{(2)}\alpha_2^{(3)} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & g\alpha_2^{(1)}\alpha_1^{(2)}\alpha_1^{(3)} & 0 & 0 & -\frac{2}{3}\alpha_3^{(1)}\alpha_1^{(2)}\alpha_1^{(3)} & -\frac{\alpha_4^{(1)}\alpha_1^{(2)}\alpha_2^{(3)}}{M_0+m} \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{S}(\mathbf{p})} \begin{bmatrix} \mathbf{x}(t) \\ u(t) \\ w(t) \end{bmatrix},$$

from which, the following Initial TP form can be constructed

$$\begin{bmatrix}\mathbf{A}(\mathbf{p}) & \mathbf{B}_u(\mathbf{p}) & \mathbf{B}_w(\mathbf{p})\end{bmatrix} = \mathcal{D} \boxtimes_{k=1}^{K} \alpha^{(k)}(\mathbf{p}^{(k)}) = \sum_{i=1}^{4}\sum_{j=1}^{2}\sum_{k=1}^{2} \mathbf{D}_{i,j,k}\alpha_i^{(1)}(p^{(1)})\alpha_j^{(2)}(p^{(2)})\alpha_k^{(3)}(\mathbf{p}^{(3)}), \tag{8.8}$$

where the non-zero elements of the initial core tensor are

$$\mathbf{D}_{1,1,1} = \begin{bmatrix} 0 & -6g\frac{M_0+m}{l} & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots \end{bmatrix}, \mathbf{D}_{1,2,1} = \begin{bmatrix} 0 & 6g\frac{M_0+m}{l} & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots \end{bmatrix}, \mathbf{D}_{2,1,1} = \begin{bmatrix} 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & g & 0 & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix},$$

$$\mathbf{D}_{3,1,1} = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&-\frac{2}{3}&0 \\ 0&0&0&0&0&0 \end{bmatrix}, \mathbf{D}_{3,2,2} = \begin{bmatrix} 0&0&0&0&0&\frac{2}{ml} \\ 0&0&0&0&0&0 \\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots \end{bmatrix}, \mathbf{D}_{4,1,1} = \begin{bmatrix} 0&0&0&0&1&0 \\ 0&0&0&0&0&0 \\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots \end{bmatrix},$$

$$\mathbf{D}_{4,1,2} = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&-M_0-m \\ 0&0&0&0&0&0 \end{bmatrix}, \mathbf{D}_{4,2,2} = \begin{bmatrix} 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&1&0&0&0 \end{bmatrix}.$$

Affine TP model

Algorithm 5.16 results in the Affine TP form from the Initial TP form. Its weighting functions are depicted in Figure 8.2. The elements of the core tensor are

$$\mathbf{G}_{111} = \begin{bmatrix} 0 & -1.196\cdot10^{-3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -4.8059\cdot10^{-4} & 0 & 0 & 4.9552\cdot10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{G}_{211} = \begin{bmatrix} 0 & 1.588\cdot10^{-4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.37\cdot10^{-6} & 0 & 0 & 6.34\cdot10^{-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{G}_{311} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{G}_{411} = \begin{bmatrix} 0 & 0.0236 & 0 & 0 & -4.7663\cdot10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -5.16\cdot10^{-3} & 0 & 0 & 3.68\cdot10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{G}_{121} = \begin{bmatrix} 0 & 9.922\cdot10^{-4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -4.938\cdot10^{-3} & 0 & 0 & 5.09\cdot10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{G}_{221} = \begin{bmatrix} 0 & -1.317\cdot10^{-4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.40\cdot10^{-5} & 0 & 0 & 6.517\cdot10^{-8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{G}_{321} = \begin{bmatrix} 0 & -3.11\cdot10^{-8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -6.79\cdot10^{-7} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{G}_{421} = \begin{bmatrix} 0 & -0.01965 & 0 & 0 & -4.897\cdot10^{-3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.0530 & 0 & 0 & 3.781\cdot10^{-3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{G}_{112} = \begin{bmatrix} 0 & 0.225 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.090414 & 0 & 0 & -9.3223\cdot10^{-3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{G}_{212} = \begin{bmatrix} 0 & -0.0298 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2.57\cdot10^{-4} & 0 & 0 & -1.193\cdot10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{G}_{312} = \begin{bmatrix} 0 & -7.072\cdot10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.2441\cdot10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{G}_{412} = \begin{bmatrix} 0 & -4.456 & 0 & 0 & 0.08966 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.971 & 0 & 0 & -0.0692 & -0.08185 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{G}_{122} = \begin{bmatrix} 0 & -0.186 & 0 & 0 & 0 & 10.39 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.929 & 0 & 0 & -0.0957 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{G}_{222} = \begin{bmatrix} 0 & 0.0247 & 0 & 0 & 0 & 1.330\cdot10^{-3} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2.648\cdot10^{-3} & 0 & 0 & -1.226\cdot10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{G}_{322} = \begin{bmatrix} 0 & 5.867\cdot10^{-6} & 0 & 0 & 0 & -1.127\cdot10^{-5} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.278\cdot10^{-4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{G}_{422} = \begin{bmatrix} 0 & 3.697 & 0 & 0 & 0.9214 & 77.21 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9.9846 & 0 & 0 & -0.711 & -0.8411 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

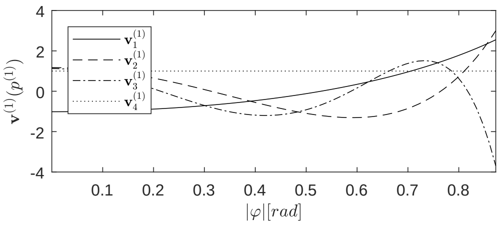
The singular values are respectively:

$$\sigma_1^{(1)} = 10.4427, \quad \sigma_2^{(1)} = 0.038941, \quad \sigma_3^{(1)} = 0.0001292, \quad \sigma_1^{(2)} = 4.5703, \quad \sigma_1^{(3)} = 0.06210.$$

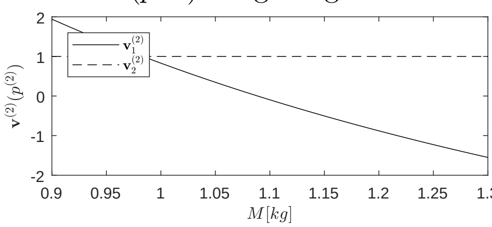
The discretisation based approach can be also applied to approximate the form by Algorithm 5.16 or Algorithm 5.20. By applying equidistant grid with sizes $M = \begin{bmatrix} 20 & 20 & 20 & 20 \end{bmatrix}$, the results give a good approximation of the Affine TP form. The weighting functions are depicted in Figure 8.3 and the singular values in this case are:

$$\sigma_1^{(1)} = 10.4577, \quad \sigma_2^{(1)} = 0.038964, \quad \sigma_3^{(1)} = 0.0001293, \quad \sigma_1^{(2)} = 4.5708, \quad \sigma_1^{(3)} = 0.06207.$$

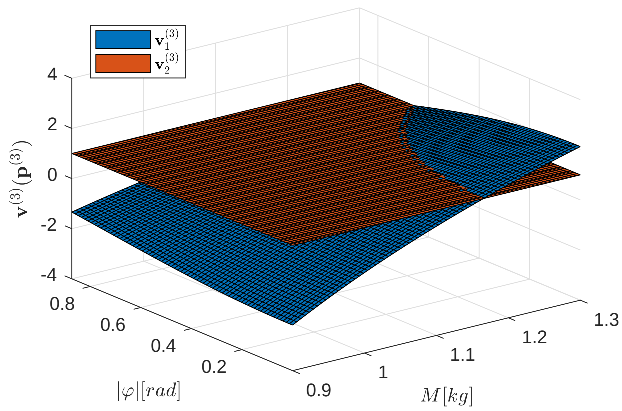
It shows well the uniqueness properties of Affine TP form described in Theorem 5.15: Because the singular values are different, only their signs can vary with the corresponding sub-tensors of the core tensor.



The $\mathbf{v}^{(1)}(p^{(1)})$ weighting functions

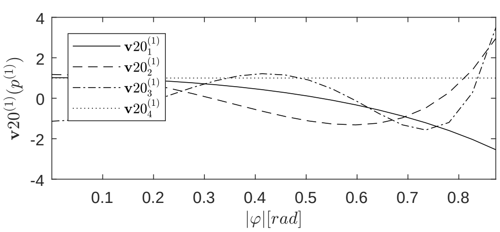


The $\mathbf{v}^{(2)}(p^{(2)})$ weighting functions

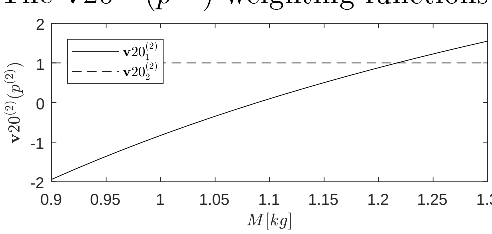


The $\mathbf{v}^{(3)}(\mathbf{p}^{(3)})$ weighting functions

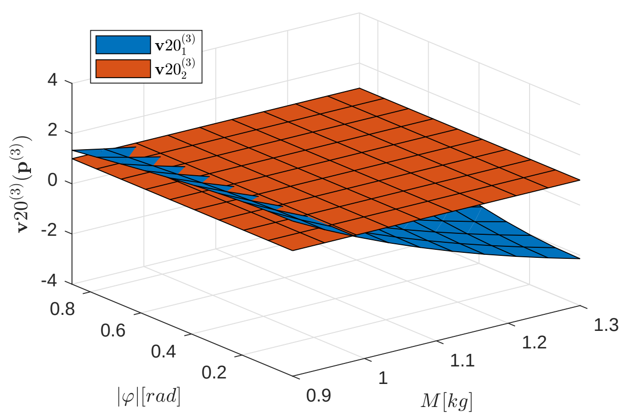
Figure 8.2: The weighting functions of the Affine TP model



The $\mathbf{v}20^{(1)}(p^{(1)})$ weighting functions



The $\mathbf{v}20^{(2)}(p^{(2)})$ weighting functions



The $\mathbf{v}20^{(3)}(\mathbf{p}^{(3)})$ weighting functions

Figure 8.3: The weighting functions of the Affine TP model via discretisation

8.4 MVS Polytopic TP Model-based controller design

The parameter dependency for the second and third parameter sets is one dimensional, for which, the determination of enclosing polytopes are trivial. The dependency for the 1st parameter set is three dimensional; then the MVS generation is applied to it (see Algorithm 6.4) to obtain the Minimal Volume Simplex Polytopic TP model

$$\mathbf{S}(\mathbf{p}) = \mathcal{S} \underset{k=1}{\overset{3}{\boxtimes}} \mathbf{w}^{(MVS,k)}(\mathbf{p}^{(k)}). \tag{8.9}$$

The resulting core tensor and the weighting functions can be written as $\mathcal{S} = \mathcal{D} \underset{k=1}{\overset{3}{\boxtimes}} \mathbf{T}^{(k)T}$ and $\mathbf{w}^{(k)}(\mathbf{p}^{(k)}) = \boldsymbol{\alpha}^{(k)}\mathbf{T}^{(k)}$, where

$$\mathbf{T}^{(1)} = \begin{bmatrix} 1.0566 \cdot 10^3 & -1.9537 \cdot 10^3 & 1.7198 \cdot 10^3 & -822.7828 \\ 4.6993 \cdot 10^3 & -9.2008 \cdot 10^3 & 8.9533 \cdot 10^3 & -4.4517 \cdot 10^3 \\ -3.0202 \cdot 10^3 & 5.9183 \cdot 10^3 & -5.7716 \cdot 10^3 & 2.8736 \cdot 10^3 \\ -1.9359 \cdot 10^3 & 3.7591 \cdot 10^3 & -3.601 \cdot 10^3 & 1.7788 \cdot 10^3 \end{bmatrix},$$
$$\mathbf{T}^{(2)} = \begin{bmatrix} 3.1818 & -3.1818 \\ -2.5 & 3.5 \end{bmatrix}, \mathbf{T}^{(3)} = \begin{bmatrix} 43.8435 & -43.8435 \\ -43.1667 & 44.1667 \end{bmatrix}.$$

The weighting functions are depicted in Figures 8.4, the resulting polytopic structure for the first parameter dependency in Figure 8.5, and the vertex system matrices within the $\mathcal{S}$ core tensor are

$$\mathbf{S}_{111} = \begin{bmatrix} 0 & -5.3703 & 0 & 0 & 1.1081 & 64.147 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10.5991 & 0 & 0 & -0.71082 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{S}_{211} = \begin{bmatrix} 0 & -5.0853 & 0 & 0 & 1.1081 & 71.5745 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 11.4072 & 0 & 0 & -0.79312 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

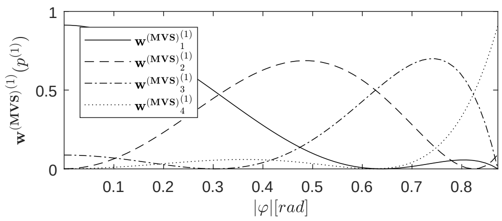
$$\mathbf{S}_{311} = \begin{bmatrix} 0 & -4.5956 & 0 & 0 & 1.1081 & 93.0425 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 13.712 & 0 & 0 & -1.031 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{S}_{411} = \begin{bmatrix} 0 & -4.4013 & 0 & 0 & 1.1081 & 106.9102 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15.1871 & 0 & 0 & -1.1847 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{S}_{121} = \begin{bmatrix} 0 & 11.5077 & 0 & 0 & 0.79151 & 64.147 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.5708 & 0 & 0 & -0.50773 & -0.71429 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{S}_{221} = \begin{bmatrix} 0 & 10.897 & 0 & 0 & 0.79151 & 71.5745 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8.148 & 0 & 0 & -0.56652 & -0.71429 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

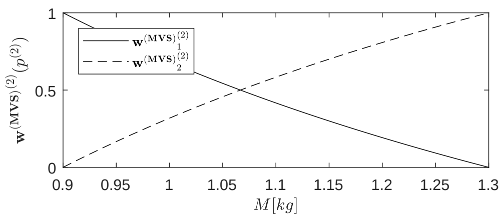
$$\mathbf{S}_{321} = \begin{bmatrix} 0 & 9.8477 & 0 & 0 & 0.79151 & 93.0425 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9.7943 & 0 & 0 & -0.73644 & -0.71429 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{S}_{421} = \begin{bmatrix} 0 & 9.4315 & 0 & 0 & 0.79151 & 106.9102 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10.848 & 0 & 0 & -0.8462 & -0.71429 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{S}_{112} = \begin{bmatrix} 0 & -5.2487 & 0 & 0 & 1.083 & 64.147 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10.3591 & 0 & 0 & -0.69472 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{S}_{212} = \begin{bmatrix} 0 & -4.9701 & 0 & 0 & 1.083 & 71.5745 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 11.1489 & 0 & 0 & -0.77517 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

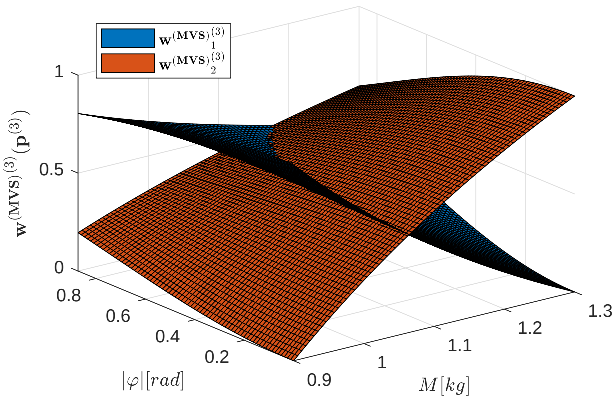
$$\mathbf{S}_{312} = \begin{bmatrix} 0 & -4.4916 & 0 & 0 & 1.083 & 93.0425 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 13.4016 & 0 & 0 & -1.0077 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{S}_{412} = \begin{bmatrix} 0 & -4.3017 & 0 & 0 & 1.083 & 106.9102 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 14.8433 & 0 & 0 & -1.1579 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$



The $\mathbf{w}^{(MVS,1)}(p^{(1)})$ weighting functions



The $\mathbf{w}^{(MVS,2)}(p^{(2)})$ weighting functions



The $\mathbf{w}^{(MVS,3)}(\mathbf{p}^{(3)})$ weighting functions

Figure 8.4: The weighting functions of the MVS polytopic TP model

$$\mathbf{S}_{122} = \begin{bmatrix} 0 & 11.2471 & 0 & 0 & 0.77358 & 64.147 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.3994 & 0 & 0 & -0.49623 & -0.71429 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{S}_{222} = \begin{bmatrix} 0 & 10.6502 & 0 & 0 & 0.77358 & 71.5745 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.9635 & 0 & 0 & -0.55369 & -0.71429 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{S}_{322} = \begin{bmatrix} 0 & 9.6248 & 0 & 0 & 0.77358 & 93.0425 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9.5725 & 0 & 0 & -0.71976 & -0.71429 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{S}_{422} = \begin{bmatrix} 0 & 9.2179 & 0 & 0 & 0.77358 & 106.9102 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10.6023 & 0 & 0 & -0.82704 & -0.71429 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

For sake of simplicity, consider simple static feedback first: $\mathbf{M}(\mathbf{p}) = \mathbf{M}$, $\mathbf{X}(\mathbf{p}) = \mathbf{X}$. Here three cases are investigated:

(A) $z(t) = \varphi(t)$ to balance the pendulum as fast as possible and to make it resistant to disturbances as much as possible. Figure 8.6 shows the results: The pendulum is balanced fast, but the car converges very slowly to the $x = 0$ coordinate. The Bode plots show that the x state variable highly depends on the w disturbance.

(B) $z(t) = x(t)$ in order to set back the car as fast as possible. Figure 8.7 shows the results with controller (B). The car is stabilized at zero position after two oscillations, but now the pendulum shows similar oscillations with slow frequency. The Bode plots show, that between $10^0 - 10^1 [rad/s]$ frequencies, it depends on the disturbance signal much more, which causes the oscillations.

(C) By taking into account these experiences, $\dot{\varphi}(t)$ is used as performance signal, which contains the higher frequencies with higher weights, and furthermore, $x(t)$ is used with smaller weight as $\mathbf{z}(t) = \begin{bmatrix} \dot{\varphi}(t) & 0.1x(t) \end{bmatrix}^T$ to take into account both goals. The properties of the resulting controller is depicted in Figure 8.8.

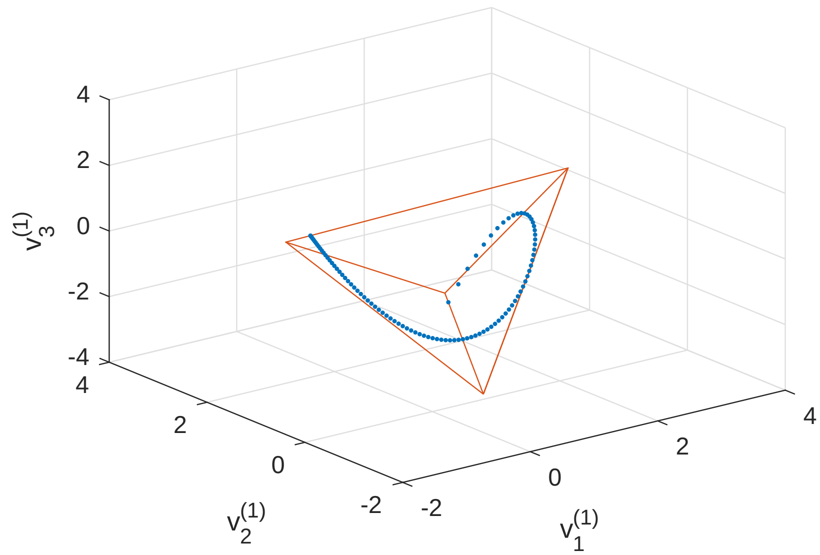
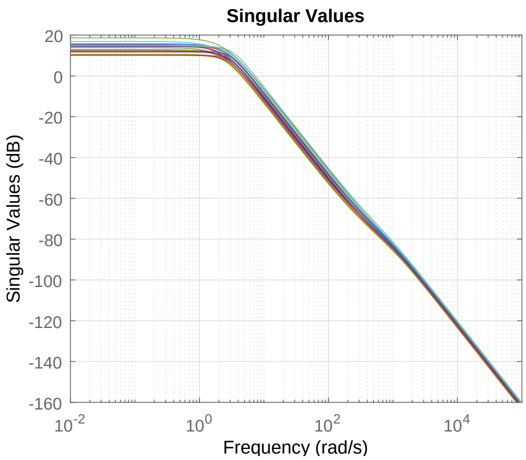
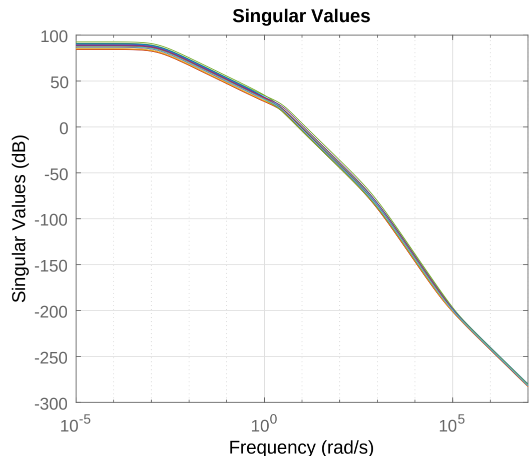


Figure 8.5: Minimal volume enclosing simplex for $\mathbf{v}^{(1)}(p_1)$ function

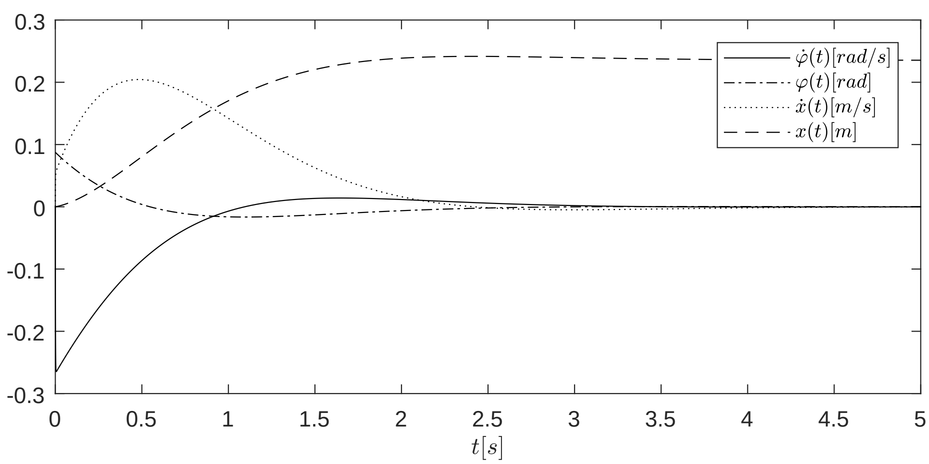
In this case, the pendulum is fast stabilized, and the convergence of $x(t)$ is not slower than with controller (B).



The transfer functions $W_{w\varphi}(j\omega)$

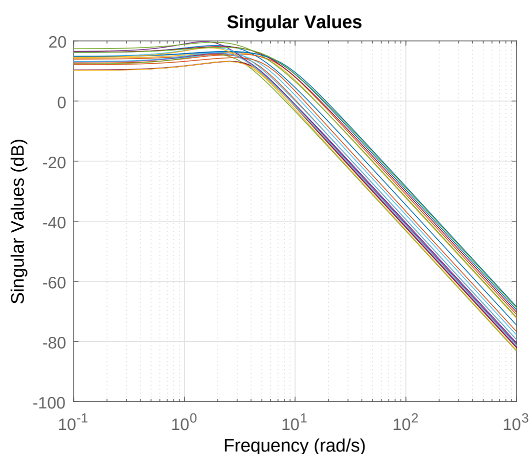


The transfer functions $W_{wx}(j\omega)$

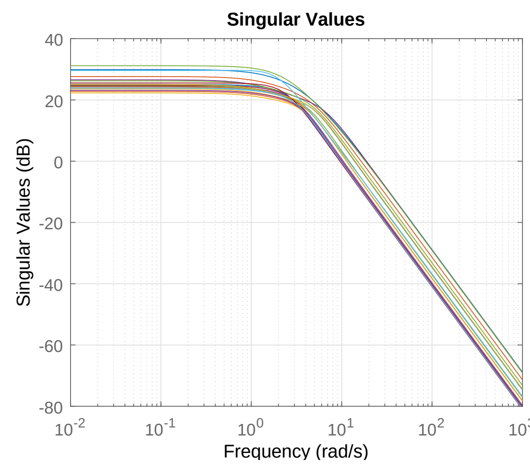


The transient of the controlled system

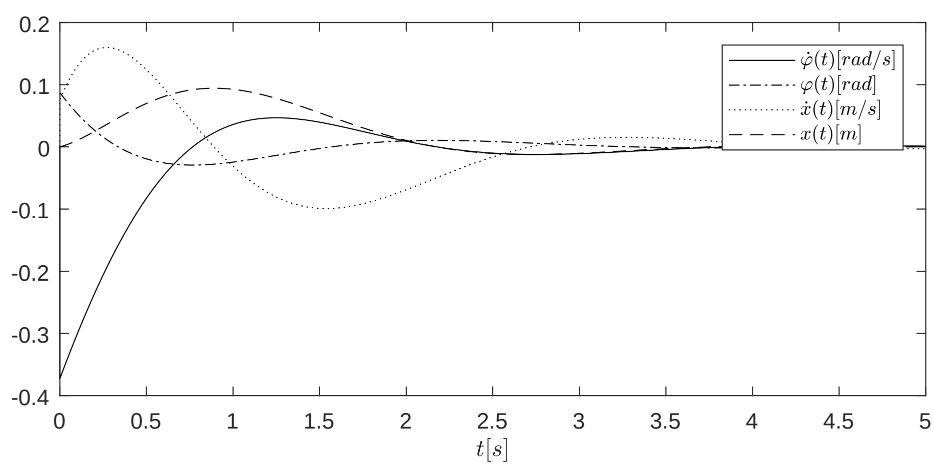
Figure 8.6: The results of controller design (A)



The transfer functions $W_{w\varphi}(j\omega)$

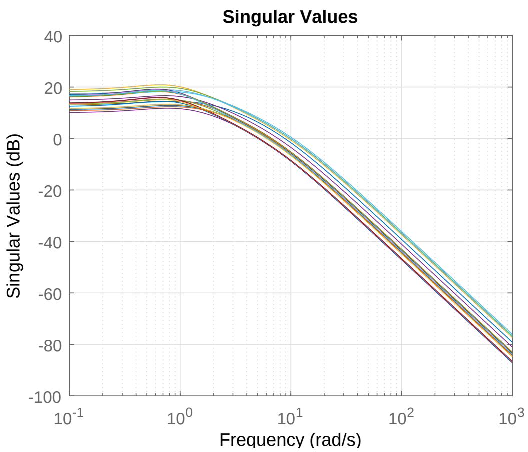


The transfer functions $W_{wx}(j\omega)$

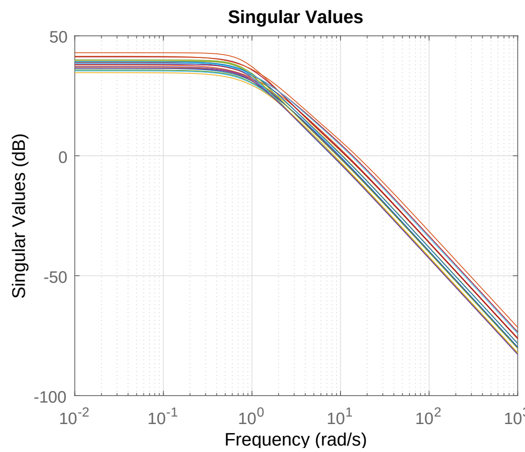


The transient of the controlled system

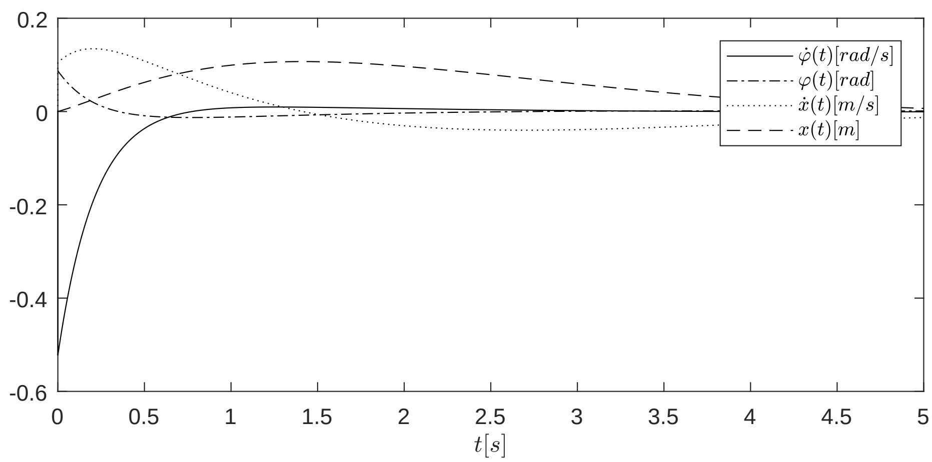
Figure 8.7: The results of controller design (B)



The transfer functions $W_{w\varphi}(j\omega)$



The transfer functions $W_{wx}(j\omega)$



The transient of the controlled system

Figure 8.8: The results of controller design (C)

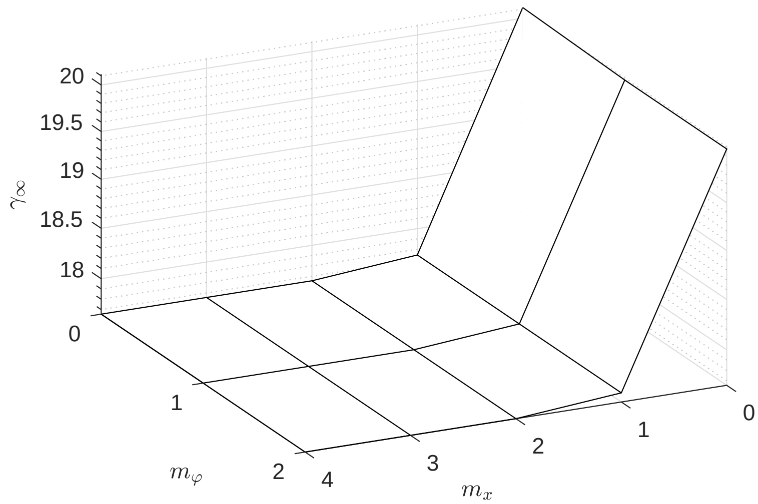


Figure 8.9: The achievable γ_∞ disturbance rejection with different variable complexities

Now, consider parameter dependant matrices $\mathbf{M}(\mathbf{p})$ and $\mathbf{X}(\mathbf{p})$ given in TP form on the same polytopic structures, as in (7.11) and (7.12). Assume that the M mass of the car may be measured or estimated. Thus the matrices can depend on it. In order to ensure $\dot{\mathbf{X}}(\mathbf{p}) = 0$, it cannot depend on the φ. For these reasons, denote the multiplicities of $\mathbf{X}(\mathbf{p})$ function with $[0, m_x, 0]$ and the multiplicities of $\mathbf{M}(\mathbf{p})$ function with $[m_\varphi, m_x, 0]$ and investigate the achievable γ_∞ disturbance rejection as function of m_φ and m_x. By applying controller design (C) with the Polya-theorem based extraction method, the results in Figure 8.9 are got. It shows that the m_φ multiplicity cannot increase the achievable performance of the control design, but if M can be measured/observed, it can be effectively applied in the controller with multiplicity $m_x = 1$ or $m_x = 2$.

8.5 Application of Polytopic TP Model Manipulation

Denote the 1-mode relative distances of the TP model by δ_1, δ_2, δ_3, δ_4, respectively. Via the MVS Manipulation, they can be optimized by applying Nelder-Mead optimization considering γ_∞ norm. It is performed by the following nested optimization

$$\gamma_\infty^* = \min_{\delta_1, \delta_3, \delta_4} \gamma_\infty \quad (Nelder - Mead\ optimisation\ from\ the\ MVS\ model)$$
$$\gamma_\infty = LMI\ optimisation\ on\ MVS\ Manipulation\ with\ (\delta_1, \delta_3, \delta_4)$$

The result is $\gamma_\infty^* = 18.794$ and the corresponding 1-mode polytopic structure is depicted in Figure 8.10. It is only slightly better than the results of the MVS model, which shows how good initial model the MVS based TP model can be.

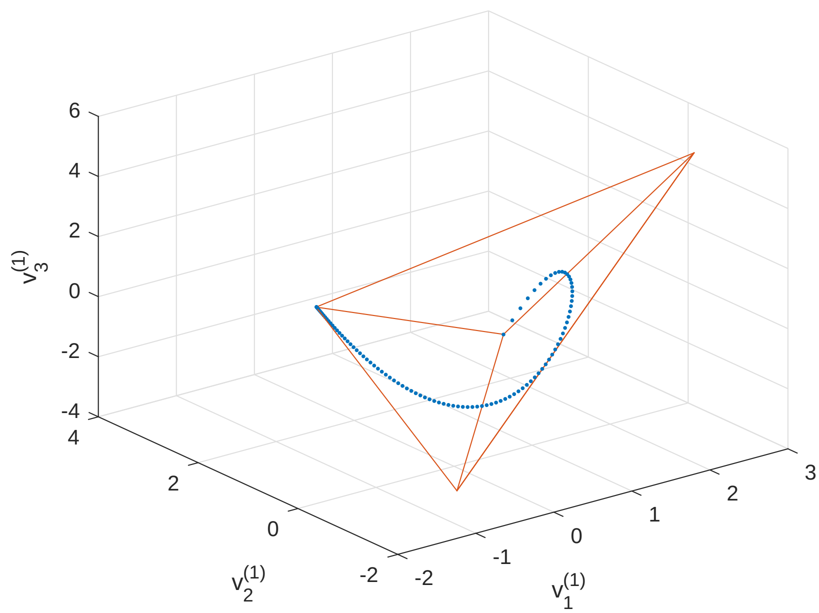


Figure 8.10: Minimal volume enclosing simplex for $\mathbf{v}^{(1)}(p_1)$ function

We can see that setting the δ_1 and δ_4 relative distances to zeros increases δ_2 and δ_3 in a large measure. Now consider the Non-Simplex manipulation opportunity and cut off the related 2nd and 3rd vertices by additional halfspaces to see if it can improve the achievable performance. This operation results in the 1-mode polytopic structure with 8 vertices, which is depicted in Figure 8.11. With this model, the achievable performance is the same.

The result shows that the performance can be hardly made better and that the increased number of vertices does not decrease the achievable performance. There are published cases where only the manipulation provides stabilizable polytopic TP models. For more details, see papers [96, 98, 100], where the benefits of these manipulation methods discussed in more details.

8.6 Conclusion

The numerical example illustrated the role of qLPV modeling to reduce the parameter dependencies of the model, the relevance of chosen parameter sets, the importance of used criteria and how it can be used to the derived MVS polytopic TP model. The results showed the effectiveness of the concept and the applied methods.

The numerical example showed that the more complex controller along the second parameter dependency can increase the achievable performance and that the MVS manipulation is enough in this case, while the MVNS could not give better results.

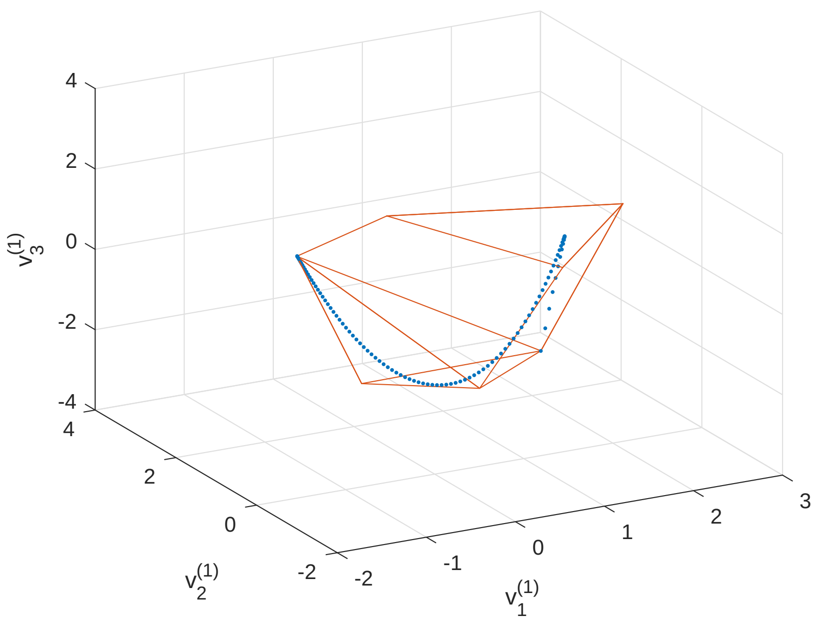


Figure 8.11: Minimal volume enclosing simplex for $\mathbf{v}^{(1)}(p_1)$ function

Chapter 9

Other published applications

The proposed concepts appear in more practical applications, and their results were applied in works of Árpád Takács, György Eigner and József Klespitz to derive a polytopic description for the developed soft tissue model [165, 163], to provide controllers for diabetes mellitus models [59, 58, 88] and to control fluid volume in blood purification therapies by József Klespitz [86].

Here the control of a dual-excenter vibration actuator is discussed in Section 9.1 and the results on the so-called Translational Oscillator with a Rotational Actuator (TORA) in Section 9.2.

9.1 Output feedback control of a dual-excenter vibration actuator

Vibration actuators are widely used in handheld devices to provide vibrotactile feedback or silent notification to the users. In most cases, miniature DC motors with eccentric rotors or the so-called coin-type shaftless vibration motors are utilized. The common disadvantage of the single rotor designs is that the frequency and the intensity of the generated vibration cannot be adjusted separately.

Ákos Miklós proposed a construction (illustrated in Figure 9.1) composed of two independently driven coaxial eccentric rotors, which makes a strongly coupled nonlinear system that allows the separate control of the frequency and amplitude by the adjustment of the angular speed and the total eccentricity [125].

In the considered model, an isotropic environment is assumed, where k denotes the stiffness and c is the damping coefficient in both directions. The two rotors are driven by torques T_1 and T_2 about the axis through the point C. The eccentricity is characterized by the mass of the rotors (m_0) and the distance e of the centre of mass from the rotation axis. Value J_0 denotes the rotor's moment of inertia, while $M = 2m_0 + m$ the total mass of the moving system. In the section φ is used to denote the mean position of the rotors, while δ represents half of the phase difference ($2\delta = \varphi_1 - \varphi_2$) between the rotors. The position of the vibrating system is described by Descartes coordinates x, y with respect to the balanced state. The mechanical model and the equations of motion can be written as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}, T_1, T_2, k, c), \tag{9.1}$$

where $\mathbf{q} = [\ x \quad y \quad \varphi \quad \delta\ ]^T$,

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} M & 0 & -2em_0\cos\delta\sin\varphi & -2em_0\sin\delta\cos\varphi \\ 0 & M & 2em_0\cos\delta\cos\varphi & -2em_0\sin\delta\sin\varphi \\ -2em_0\cos\delta\sin\varphi & 2em_0\cos\delta\cos\varphi & 2J_0 & 0 \\ -2em_0\sin\delta\cos\varphi & -2em_0\sin\delta\sin\varphi & 0 & 2J_0 \end{bmatrix},$$

$$\mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}, T_1, T_2, k, c) = \begin{bmatrix} -c\dot{x} - kx + 2(\dot{\delta}^2 + \dot{\varphi}^2)em_0\cos\delta\cos\varphi - 4\dot{\delta}\dot{\varphi}em_0\sin\delta\sin\varphi \\ -c\dot{y} - ky + 2(\dot{\delta}^2 + \dot{\varphi}^2)em_0\cos\delta\sin\varphi + 4\dot{\delta}\dot{\varphi}em_0\sin\delta\cos\varphi \\ T_1 + T_2 \\ T_1 - T_2 \end{bmatrix}.$$

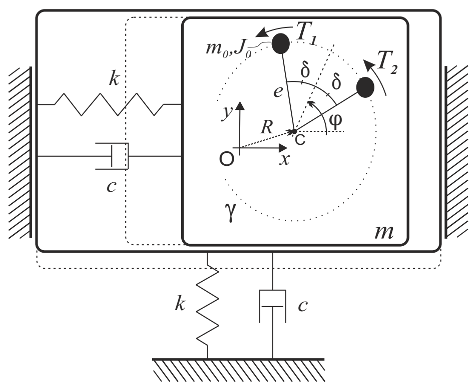


Figure 9.1: Mechanical model of the vibrating system

The key feature of this structure is that the vibration is generated by two actuators, that is, the amplitude and the frequency can be controlled independently because the total eccentricity of the system results from the offset. By the effect of T_1 and T_2, the system can oscillate about the equilibrium point O. As the nonlinear and parameter-dependent equations of motions (9.1) shows, the rotor motions are not independent due to the dynamic coupling effect of the suspension. The amplitude of the generated vibration depends on the phase difference, the angular velocity, and the suspension parameters.

The paper [101] presents a complete control design approach based on qLPV modeling and LMI-based synthesis utilizing the TP Model Transformation to determine the Polytopic TP representation of the parameter-dependent nonlinear system. The design approach is demonstrated via a concrete numerical example using the parameters of a real dual-excenter prototype device developed by the Research Group on Dynamics of Machines and Vehicles (MTA-BME) [124].

The time delay and the noise of the sensor system and the discrete-time characteristics of the control loop were not modeled in the design phase. However, the simulation showed that these unfortunate attributes significantly influence the control quality at low angular velocities. Namely, at small ω, the estimated signals have relatively large time-delay and noise because the relative rate of change of the angular velocity can also be high which renders the sensor system prone to instability and eventually causes the overall system to become unstable.

This negative effect was reduced through Kalman-filtering. Since the system parameters are only partly known, only the motor characteristics have been considered in the filter formulas. The overall system extended with Kalman-filter shows stable and favorable behavior concerning settling time and overshoot within the parameter domain $200 \leq \omega \leq 1000$ and $\leq \delta \leq \pi$ offset range.

The stability and performance of the overall control system are evaluated via numerical simulations in MATLAB Simulink environment. The simulation takes the real characteristics (delay, inaccuracy, stability issues) of the measurement (using $n = 3$ optical sensors) into consideration and also makes use of the quantized and bounded control signal ($U_1(t)$, $U_2(t)$) and considers inductance and commutation in the motor model. The controller sampling time is set to $T_s = 1[ms]$. The results are shown in Figure 9.2, where the system's behaviour is investigated at different circular frequencies $\omega = 200, 600, 1000[rad/s]$. The simulation shows that the controller is capable of stabilizing the $\delta = \pi/2$ (balanced) equilibrium state and can govern the offset into different $\delta \neq \pi/2$ values (vibration with $R \neq 0$ amplitude) with reasonable settling time, overshoot and oscillation.

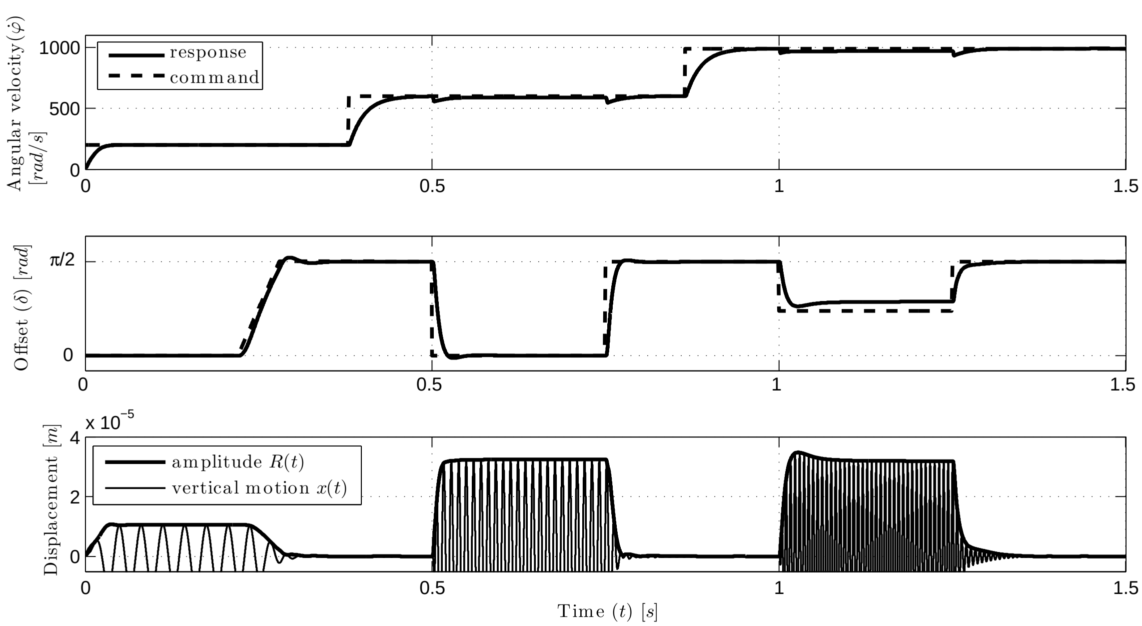


Figure 9.2: Simulation results: Three short vibration impulses show that the frequency and amplitude are adjustable independently.

Based on the simulation results, we can conclude that the overall control system enables the adjustment of the frequency and the intensity of the vibration independently while keeping the system stable, the maximum settling time is $70[ms]$ in $\dot{\varphi}$ and $30[ms]$

in δ. This statement shows the practical value of the applied theoretical achievements in a real engineering problem.

9.2 Translational Oscillator with a Rotational Actuator (TORA) system

Papers [98, 100] discuss the complete workflow of control design for a well-known nonlinear benchmark system, the so-called translational oscillator with an eccentric rotational mass actuator [37, 113] shown in Figure 9.3. The goal of the control effort is to stabilize its translational motion using a rotational actuator [35, 36, 38, 113, 138, 142, 181].

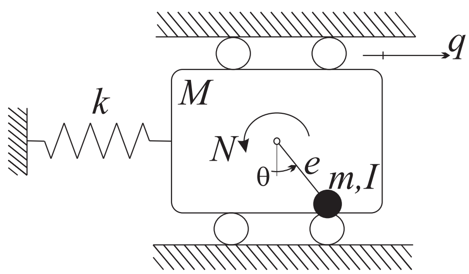


Figure 9.3: The mechanical model of the TORA system

The equation of motion is given as:

$$\begin{aligned}(M+m)\ddot{q}+kq &= -me\left(\ddot{\theta}\cos\theta-\dot{\theta}^2\sin\theta\right),\\ (I+me^2)\ddot{\theta} &= -me\ddot{q}\cos\theta+N,\end{aligned} \tag{9.2}$$

where q is the translational coordinate to be stabilized and θ is the position of the actuator driven by N torque. For deriving the qLPV model, it is reformulated in dimensionless form and coordinate tranformation is applied. By applying parameter separation, the resulting TP model has two parameter dependency ($p_1 = |\theta|$, and $p_2 = f(\theta, \dot{\theta})$), and it has the following general form

$$\begin{bmatrix}\mathbf{x}'\\ z\end{bmatrix} = \mathbf{S}(\mathbf{p})\begin{bmatrix}\mathbf{x}\\ u\end{bmatrix}, \tag{9.3}$$

where

$$\mathbf{S}(\mathbf{p}) = \mathcal{S}\underset{n=1}{\overset{2}{\boxtimes}}\mathbf{w}^{(n)}(p_n) = \sum_{j_1=1}^{J_1}\sum_{j_2=1}^{J_2} w_{j_1}^{(1)}(p_1)w_{j_2}^{(2)}(p_2)\begin{bmatrix}\mathbf{A}_{j_1,j_2} & \mathbf{B}_{j_1}\\ \mathbf{C}_{j_1} & 0\end{bmatrix}. \tag{9.4}$$

The goal of the control synthesis is the fast settling of the $\xi(\tau)$ position from ξ_0 to

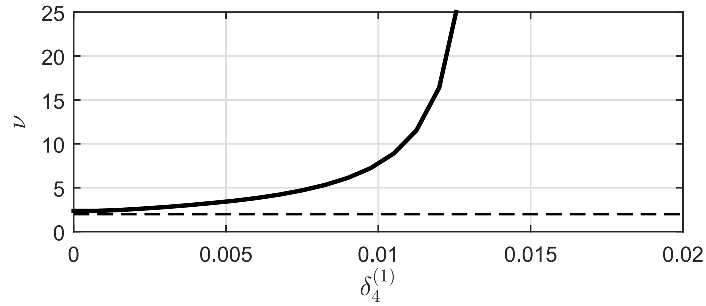


Figure 9.4: The results of optimisation LMI on manipulated models, the dash line shows the minimal feasible ν value for all $\mathbf{S}(\mathbf{p})$ system matrices for comparison

zero, which can be characterized by the cost function

$$J = \int_{\tau=0}^{\infty} \left(z^T(\tau) z(\tau) + R u^T(\tau) u(\tau) \right) d\tau \qquad \textit{where} \;\; R > 0. \tag{9.5}$$

The used control criterion ensures $J < \nu$ for all possible $\mathbf{p}(t)$ trajectory starting from the $\mathbf{x}_0$ state and ν is to be minimized. Furthermore, because the parameters are functions of the state variable, it is necessary to ensure that these variables do not leave the modelled $|\theta| < \theta_{max}$, $|\theta'| < \theta'_{max}$ domain during the transient motion.

First, the MVS-type convex TP model is determined, but it is not stabilizable. The problem can be overwhelmed by manipulating the polytope decreasing the relative distance $\delta_4^{MVS(1)} = 0.07$. The results on the manipulated models with different $\delta_4^{(1)} < \delta_4^{MVS(1)}$ values are depicted in Figure 9.4. At $\delta_4^{(1)} \approx 0.013$, the ν worst case value of the cost function tends to infinity. As the relative distance decreases, the cost converges to $\nu = 2.36$, which means the optimum in this case.

By applying cutting halfspaces, the problematic region can be removed after six cuts. The steps monotonously increase the number of vertices and (triangle) facets from $J_1 = F_1 = 4$ up to $J_1 = 16$ and $F_1 = 28$ as you can see in Figure 9.5: first Figure 9.5 shows the initial enclosing MVS polytope with the $\mathbf{u}^{(1)}(p_1)$ trajectory and the resulting non-simplex one. The images show well that the non-simplex polytope tends to the convex hull at a certain region.

The resulting polytopic model with 14×2 vertices is feasible. The corresponding controller was validated via numerical simulation in MATLAB Simulink environment. The system shows stable behavior and the prescribed performance within the investigated range of initial conditions. Figure 9.6 shows a simulation result.

The results show that the presented approach for polytope manipulation can be used effectively to increase the achievable performance in polytopic model-based controller design by excluding non-stabilizable regions located between the exact convex hull and the minimal volume enclosing simplex.

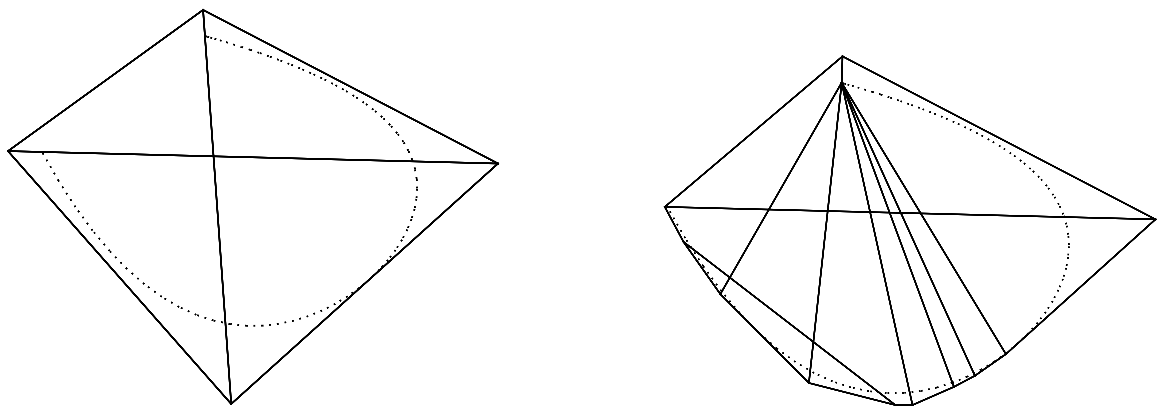


Figure 9.5: The MVS enclosing polytope and the cut one

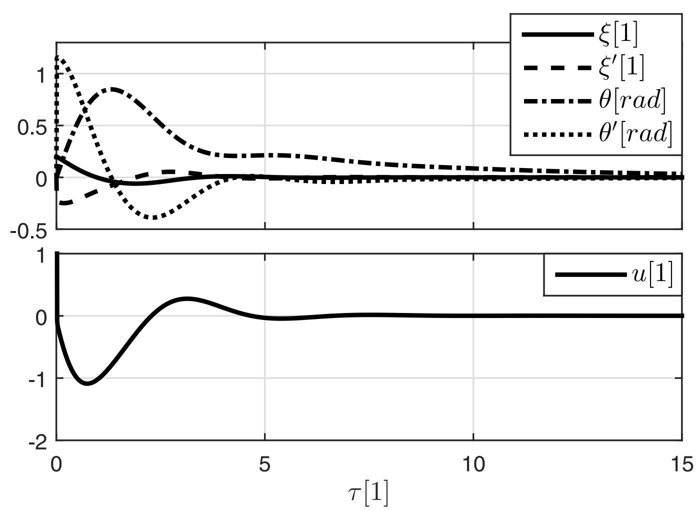


Figure 9.6: Behaviour of the controlled system

Part IV

Conclusion

Chapter 10

Summary of the scientific results

The chapter concludes the achieved results by composing the theses. Figure 10.1 recalls the new control design workflow and highlights the relevance to the theses.

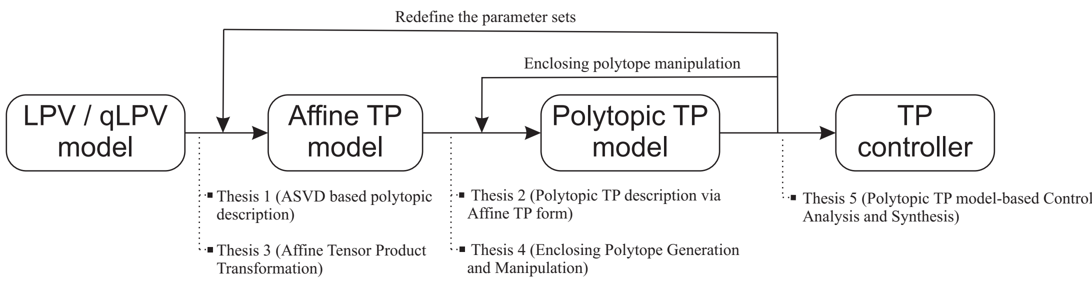


Figure 10.1: The new control design workflow and role of the elaborated theses

Thesis 1 (ASVD based polytopic description)

The Affine Singular Value Decomposition of multivariate functions

$$\mathfrak{f}(\mathbf{x}) = \sum_{d=1}^{D+1} \mathfrak{a}_d v_d(\mathbf{x}) \tag{10.1}$$

provides a factorization with the following properties:

- It represents the affine structure of the image set by describing the affine hull via the offset ($\mathfrak{a}_{D+1}$) and an orthogonal, ordered basis ($\mathfrak{a}_1, ..., \mathfrak{a}_D$) in such a way that the related homogeneous coordinate ($\mathbf{v}(\mathbf{x})$) are orthonormal functions as in the SVD.

- Through ASVD, the derivation of polytopic description can be transformed into a geometrical problem: Determination of enclosing polytope for a D dimensional point set (image of $\mathbf{v}(\mathbf{x})$).

- It is a canonical representation, because it shows uniqueness in terms of similarity as the SVD.

- The ASVD form is suitable for complexity reduction with minimal error (in terms of Frobenius norm) by omitting the last basis directions, where the complexity is understood as the D dimension of the affine hull.

The proposed algorithm is suitable for numerical reconstruction through analytical (exact) or discretisation based (approximating) initial forms.

Corresponding publications: [96, 98].

Thesis 2 (Polytopic TP description via Affine TP form)

The previous definition of Polytopic TP form can be relaxed along the following properties:

- arbitrary parameter sets can be used instead of complete separation of the scalar parameter dependencies,
- parameter dependencies can be used with arbitrarily high multiplicities to serve the further optimisation structures directly,
- the formalism is extended to Hilbert-spaces in general, by defining Lathauwer's tensor algebra to Hilbert spaces.

Consider the function $\mathfrak{f}(\mathbf{p}) : \Omega \to H$. According to the chosen parameter sets (denoted as $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots \subseteq \mathbf{p}$ and their domains accordingly as $\Omega_1, \Omega_2, \dots$), the TP form

$$\mathfrak{f}(\mathbf{p}) = \mathcal{F} \times_1 \mathbf{w}^{(1)}(\mathbf{p}^{(1)}) \times_2 \mathbf{w}^{(2)}(\mathbf{p}^{(2)}) \dots \tag{10.2}$$

$$\text{where} \quad \mathbf{w}^{(k)} : \Omega_k \to \mathbb{R}^{J_k}, \quad \text{and} \quad \mathcal{F} \in H^{J_1 \times J_2 \times \dots} \tag{10.3}$$

is called a Polytopic TP form if

$$\mathbf{w}^{(k)}(\mathbf{p}^{(k)}) \mathbf{1}^{J_k \times 1} = 1, \quad \mathbf{w}^{(k)}(\mathbf{p}^{(k)}) \geq \mathbf{0} \qquad \forall \mathbf{p}^{(k)} \in \Omega_k. \tag{10.4}$$

The extended polytopic TP form can be derived through Affine TP form

$$\mathbf{f}(\mathbf{p}) = \mathcal{F}^{aff} \times_1 \mathbf{v}^{(1)}(\mathbf{p}^{(1)}) \times_2 \mathbf{v}^{(2)}(\mathbf{p}^{(2)}) \dots, \tag{10.5}$$

in which the dependencies on the parameter sets show ASVD structures. The derivation requires enclosing polytopes for the weighting functions in the Euclidean space with the given dimension.

If the parameter sets are disjoint, the following statements hold:

- The complexity (geometric dimension) of the dependencies on the parameter sets can be reduced with minimal error (in terms of Frobenius-norm).
- The representation is canonical, since it inherits the uniqueness properties of ASVD.

Corresponding publications: [89, 95].

Thesis 3 (Affine Tensor Product Transformation)

The proposed Affine Tensor Product Transformation provides numerical algorithms to reconstruct the Affine TP form for multivariate functions considering arbitrary parameter sets:

- Exact TP forms can be obtained if the dependencies from the parameter sets can be separated analytically.

- In other cases, approximating TP forms can be obtained by constructing the initial TP form

$$\hat{\mathfrak{f}}(\mathbf{p}) = \mathcal{D} \underset{k=1}{\overset{K}{\boxtimes}} \boldsymbol{\alpha}^{(k)}(\mathbf{p}^{(k)}) = \sum_{m_1=1}^{M_1} \cdots \sum_{m_K=1}^{M_K} \mathfrak{d}_{m_1,\ldots,m_K} \prod_{k=1}^{K} \alpha_{m_k}^{(k)}(\mathbf{p}^{(k)}) \tag{10.6}$$

  via discretisation.

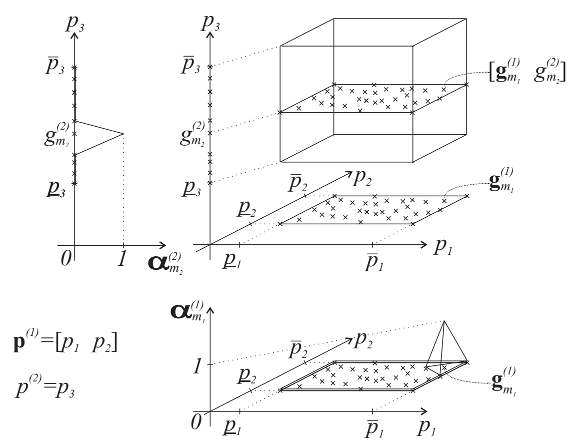


Figure 10.2: Illustration of discretisation via multivariate interpolatory functions

Furthermore, the exactness of the derived (discretised or complexity reduced) Affine TP form

$$\hat{\mathfrak{f}}(\mathbf{p}) = \hat{\mathcal{F}} \underset{k=1}{\overset{K}{\boxtimes}} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}) \tag{10.7}$$

can be restored by determining the ASVD of its error (without parameter separation)

$$\mathfrak{f}(\mathbf{p}) - \hat{\mathfrak{f}}(\mathbf{p}) = \tilde{\mathcal{F}} \times_{K+1} \mathbf{v}^{(K+1)}(\mathbf{p}) \tag{10.8}$$

and inserting it in the Affine TP form as a new parameter dependency:

$$\mathfrak{f}(\mathbf{p}) = \underbrace{\left(\hat{\mathcal{F}} \times_{K+1} \mathbf{1}^{(D_{K+1})} + \tilde{\mathcal{F}} \underset{k=1}{\overset{K}{\boxtimes}} \mathbf{1}^{(D_k)} \right)}_{\mathcal{F}} \underset{k=1}{\overset{K}{\boxtimes}} \mathbf{v}^{(k)}(\mathbf{p}^{(k)}) \times_{K+1} \mathbf{v}^{(K+1)}(\mathbf{p}), \tag{10.9}$$

where $\mathbf{1}^{(D)} = \begin{bmatrix} \mathbf{0}^{1\times D} & 1 \end{bmatrix}^T$. Corresponding publications: [89, 90, 95].

Thesis 4 (Enclosing Polytope Generation and Manipulation)

The envelope of polytopic models usually includes a larger set of LTI systems than the LPV/qLPV models highly increasing the conservativeness of the controller design. It is essential to avoid or at least minimize their presence of additional systems without significantly increasing the number of vertices.

This aspect can be taken into consideration through a two phase approach in order to maximize the achievable performance with the polytopic model:

1. First, the polytopic model is generated by determining the enclosing polytopes based on simple geometric aspects.
2. By analysing the actual polytopic model, geometric manipulations are performed on the enclosing polytopes to achieve satisfying control performance.

According to this concept, the following enclosing polytope generation and manipulation algorithms are proposed:

- Generation of Minimal Volume Simplex (MVS).
- Manipulation of the MVS by applying constraints to close some of the vertices to the convex hull.
- Deriving Non-Simplex enclosing polytopes by cutting regions off from the polytope by one or more halfspaces.
- Local Minimization of Volume of Enclosing Non-Simplex polytopes.

The algorithms are elaborated for higher dimensional spaces in general and the minimal volume has only approximating meaning because the volume minimization problem is highly non-convex.

Corresponding publications: [96, 97, 98, 100].

Thesis 5 (Polytopic TP model-based Control Analysis and Synthesis)

The concept of Polytopic Tensor Product (TP) Model based control analysis and synthesis has been revisited and renewed by proposing the use of TP-structured variables in the definite conditions derived from the applied control criteria. For TP forms that can depend on multivariate parameter sets (optionally with two times or higher multiplicities)

$$\mathbf{X}(\mathbf{p}) = \mathcal{X} \underbrace{\times_1 \mathbf{w}^{(1)}(\mathbf{p}^{(1)}) \times_2 \mathbf{w}^{(1)}(\mathbf{p}^{(1)}) \cdots \times_{M_1} \mathbf{w}^{(1)}(\mathbf{p}^{(1)})}_{M_1} \underbrace{\times_{M_1+1} \mathbf{w}^{(2)}(\mathbf{p}^{(2)}) \ldots}_{M_2} \ldots, \tag{10.10}$$

a compact TP formalism was proposed

$$\mathbf{X}(\mathbf{p}) = \mathcal{X} \mathop{\boxtimes}_{k=1}^{K(\mathbf{M})} \mathbf{w}^{(l(k,\mathbf{M}))}(\mathbf{p}^{(l(k,\mathbf{M}))}), \tag{10.11}$$

$$\text{where} \quad K(\mathbf{M}) = \sum_i M_i, \quad l(k, \mathbf{M}) = i \;\text{ if }\; \sum_{a=1}^{i-1} M_a < k < \sum_{a=1}^{i} M_a, \tag{10.12}$$

and the $\mathbf{M}$ multiplicity vector describes the structure. By setting the multiplicities, the parameter dependencies can be neglected or considered with arbitrary high complexity in the variables of controller-candidate, Lyapunov-function candidate and slack variables as well.

The definite conditions on the structures of these variables e.g.,

$$\mathbf{A}(\mathbf{p})\mathbf{X}(\mathbf{p}) + \mathbf{X}(\mathbf{p})\mathbf{A}^T(\mathbf{p}) < 0 \tag{10.13}$$

or the Bounded Real Lemma result in definite conditions on Polytopic TP forms. They can be handled in general by defining a recursive algorithm to reformulate them into Linear Matrix Inequalities (LMIs) or Bilinear Matrix Inequalities (BMIs), etc. according to the design method in consideration.

Corresponding publication: [92].

Appendix

Appendix A

Former methods of TP Model Transformation to generate simplex enclosing polytopes

The former literature calls the following algorithms to convex hull manipulation methods and applies them on the weighting functions of HOSVD based form after centralizing and reSVD them – that returns the related affine subspace. Behind the algorithm of these methods, there appears clear geometric meaning and it explains the practical properties of the algorithms. For these reasons, this chapter concludes the three most used algorithms allowing their theoretical comparison to the proposed MVS algorithms.

Their common first step is centralization and reSVD - that provides affine decompositions (see Sec. 4.2). Here we will denote its dimension by D and the orthogonal weighting functions to be enclosed by $\mathbf{u}(\mathbf{p})$.

A.1 SNNN enclosing simplex algorithm

Algorithm A.1 (SNNN enclosing polytope)**.** *In the D dimensional space, construct a simplex with vertices $\mathbf{r}_1 = \mathbf{e}_1^D, \mathbf{r}_2 = \mathbf{e}_2^D, ..., \mathbf{r}_D = \mathbf{e}_D^D, \mathbf{R}_{D+1} = \mathbf{0}^D$ and inflate it from its centre $\bar{\mathbf{r}} = \mathbf{1}^D/J$ until it becomes an enclosing polytope:*

$$\mathbf{r}_j^{SNNN} = \bar{\mathbf{r}} + (1+\theta)(\mathbf{r}_j^{(0)} - \bar{\mathbf{r}}), \tag{A.1}$$

where

$$\theta = -J \min_{1\le j\le J, \mathbf{p}\in\Omega} w_j(\mathbf{p}), \qquad \mathbf{w}(\mathbf{p}) = \begin{bmatrix} \mathbf{u}(\mathbf{p}) & 1 - \sum_d u_d(\mathbf{p}) \end{bmatrix}. \tag{A.2}$$

Fig. A.1 shows well, why it shows conservativeness during the control design: Usually only one facet touches the $\mathfrak{U}$ set and because the expansion is not started from the center of the $\mathfrak{U}$ set, the vertices except one will far away from the set. The weighting functions always show these properties: Only one weighting function closes the one value, and only one reaches the zero value. It is really fast because it does not include optimization.

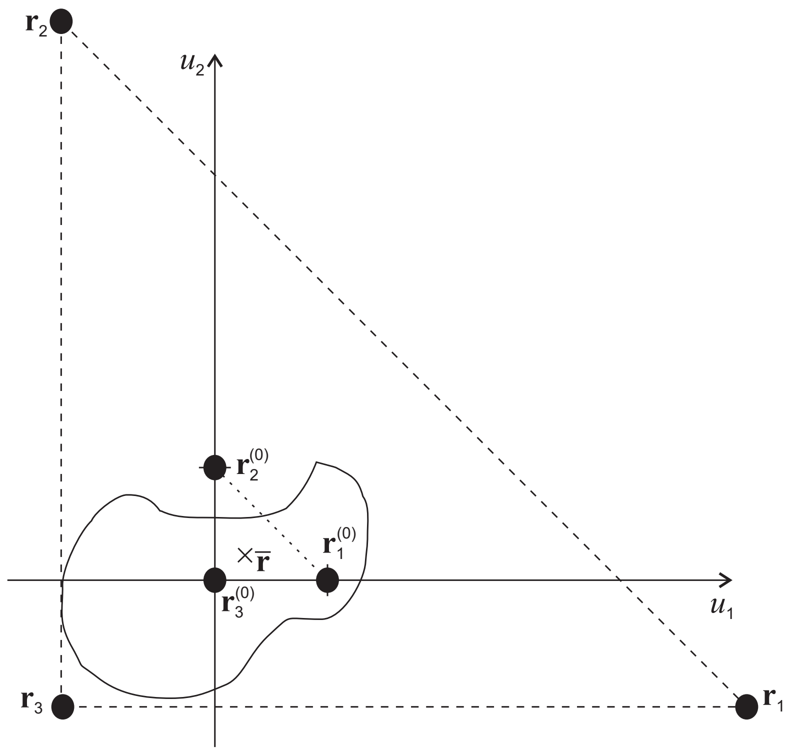


Figure A.1: Illustration of SNNN enclosing polytope generation

A.2 IRNO enclosing simplex algorithm

The goal of the algorithm is to obtain an enclosing simplex in the D dimensional space with the same relative distances in an iteration with D steps.

Algorithm A.2 (IRNO enclosing polytope)**.**

Step 1 (Initial polytope)**.** *Consider the initial polytope with 2 vertices*

$$\mathbf{r}_1^{[2]} = \begin{bmatrix}\min_{\mathbf{p}} u_1(\mathbf{p}) & 0 & \dots & 0\end{bmatrix}, \qquad \mathbf{r}_2^{[2]} = \begin{bmatrix}\max_{\mathbf{p}} u_1(\mathbf{p}) & 0 & \dots & 0\end{bmatrix}, \tag{A.3}$$

that is enclosing of function $\begin{bmatrix}u_1(\mathbf{p}) & 0 & \dots & 0\end{bmatrix}$ *with weighting functions*

$$w_1^{[2]} = \frac{\max_{\mathbf{p}}(u_1(\mathbf{p})) - u_1(\mathbf{p})}{\max_{\mathbf{p}}(u_1(\mathbf{p})) - \min_{\mathbf{p}}(u_1(\mathbf{p}))}, \qquad w_2^{[2]} = \frac{u_1(\mathbf{p}) - \min_{\mathbf{p}}(u_1(\mathbf{p}))}{\max_{\mathbf{p}}(u_1(\mathbf{p})) - \min_{\mathbf{p}}(u_1(\mathbf{p}))}. \tag{A.4}$$

Step 2 (Iteration)**.** *Consider the following polytope generation as a function of a scalar α parameter from a polytope with vertices $\mathbf{r}_1^{[a]}, \mathbf{r}_2^{[a]}, \dots, \mathbf{r}_a^{[a]}$.*

A Denote the previous vertices as $\mathbf{r}_i^A = \mathbf{r}_i^{[a]}$ for $i = 1..a$, and amend them with the vertex

$$\mathbf{r}_{a+1}^A = \alpha \mathbf{e}_{a+1}^D + \sum_{i=1}^{a} \mathbf{r}_i^A / a. \tag{A.5}$$

Then the weighting functions

$$w_{a+1}^{A}(\mathbf{p}) = u_{a+1}(\mathbf{p})/\alpha \quad and \quad w_i^{A}(\mathbf{p}) = w_i^{[a]}(\mathbf{p}) - w_{a+1}^{A}(\mathbf{p})/a \quad \forall i = 1..a. \quad \text{(A.6)}$$

B Move the polytope such that the facets in front of $i = 1..a$ *vertices until they touch the* $\{\begin{bmatrix}u_1(\mathbf{p}) & \dots & u_{a+1}(\mathbf{p}) & 0 & \dots & 0\end{bmatrix} \big| \mathbf{p} \in \Omega\}$ *set as*

$$\mathbf{r}_i^{B} = \mathbf{r}_i^{A} + \Delta, \quad \forall i = 1..(a+1),\ \Delta = \sum_{l=1}^{a} \left(\mathbf{r}_l^{A} - \mathbf{r}_{a+1}^{A}\right) \min_{\mathbf{p}} w_l^{A}(\mathbf{p}). \qquad \text{(A.7)}$$

Then the weighting functions can be written as

$$w_i^{B}(\mathbf{p}) = w_i^{A}(\mathbf{p}) - \min_{\mathbf{p}} w_i^{A}(\mathbf{p}) \qquad \forall i = 1..a, \qquad \text{(A.8)}$$

$$w_{a+1}^{B}(\mathbf{p}) = w_{a+1}^{A}(\mathbf{p}) + \sum_{i=1}^{a} \min_{\mathbf{p}} w_i^{A}(\mathbf{p}). \qquad \text{(A.9)}$$

C Expand the vertices $\mathbf{r}_1^{B}, ..., \mathbf{r}_a^{B}$ *from the vertex* $\mathbf{r}_{a+1}^{B}$ *until the maximal values of the corresponding weighting functions become one as*

$$\mathbf{r}_i^{C} = \mathbf{r}_i^{B} + \beta_i(\mathbf{r}_i^{B} - \mathbf{r}_{a+1}^{B}), \quad \beta_i = \max_{\mathbf{p}} w_i^{B}(\mathbf{p}) - 1 \quad \forall i = 1..a, \quad \mathbf{r}_{a+1}^{C} = \mathbf{r}_{a+1}^{B}. \qquad \text{(A.10)}$$

Then the corresponding weighting functions can be computed as $w_i^{C}(\mathbf{p}) = w_i^{B}(\mathbf{p})/(1+\beta_i)$ *for* $i = 1..a$ *and* $w_{a+1}^{C}(\mathbf{p}) = 1 - \sum_{i=1}^{a} w_i^{C}(\mathbf{p})$.

D Move the facet in front of the $(a+1)$*-th vertex until the polytope becomes enclosing as*

$$\mathbf{r}_i^{D} = \mathbf{r}_i^{C} + \gamma(\mathbf{r}_i^{C} - \mathbf{r}_{a+1}^{C}), \qquad \forall i = 1..a, \quad \mathbf{r}_{a+1}^{D} = \mathbf{r}_{a+1}^{C}. \quad \gamma = -\min_{\mathbf{p}} w_{a+1}^{C}(\mathbf{p}) \qquad \text{(A.11)}$$

Then the weighting functions $w_i^{D}(\mathbf{p}) = w_i^{C}(\mathbf{p})/(1+\gamma)$ *for* $i = 1..a$ *and* $w_{a+1}^{D}(\mathbf{p}) = (w_{a+1}^{C}(\mathbf{p}) + \gamma)/(1+\gamma)$.

For all $j = 3..(D+1)$ *value: Consider the vertices* $\mathbf{r}_1^{[j-1]}, \dots, \mathbf{r}_{j-1}^{[j-1]}$ *and the corresponding weighting functions* $\mathbf{w}^{[j-1]}(\mathbf{p})$ *and find an* α *such that the resulted weighting functions of the upper steps have the same maximal value. Denote the resulted vertices and weighting functions with* $\mathbf{r}_1^{[j]}, \dots, \mathbf{r}_j^{[j]}$ *and* $\mathbf{w}^{[j]}(\mathbf{p})$.

Fig. A.2 illustrates the initial polytope and steps A, B, C, D for a given α value.

The resulted weighting functions have the same maximal value, and the facets touch the image set of $\mathbf{u}(\mathbf{p})$. It must be remarked, that the algorithm does not guarantee that the common relative distance is as small as possible, because there is not a general optimization.

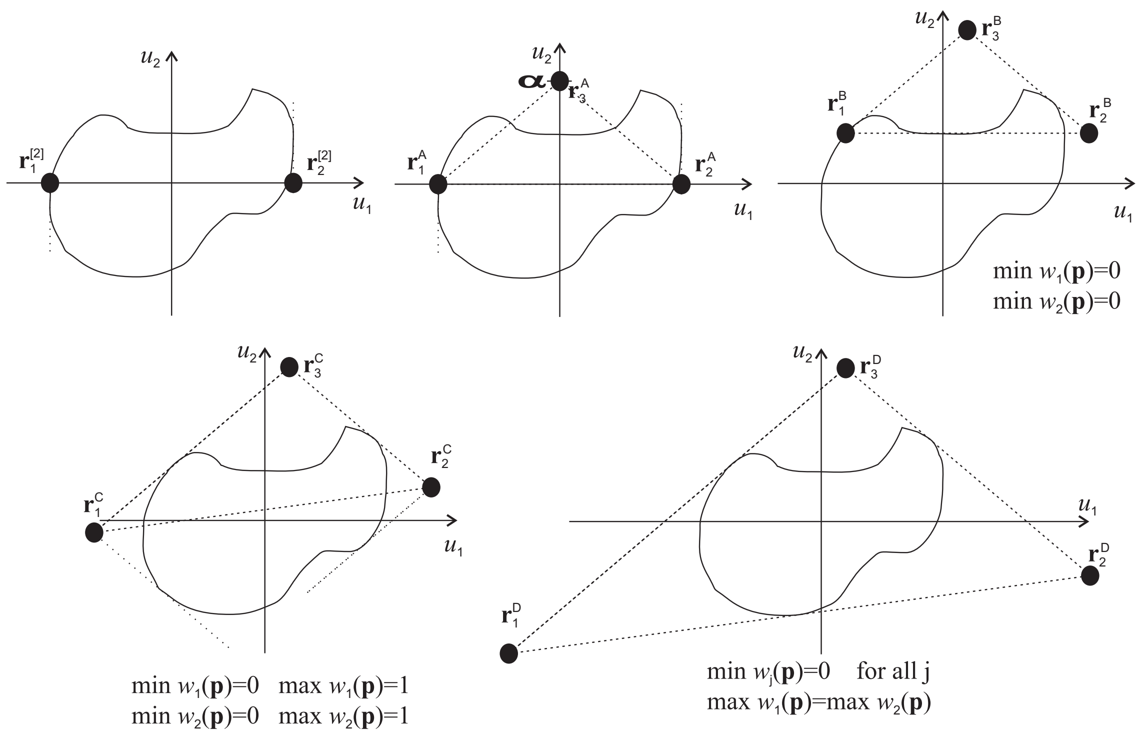


Figure A.2: Illustration of A, B, C, D steps for a given α

A.3 CNO enclosing simplex algorithm

The algorithm is to minimize a cost function derived from the $\delta_1, ..., \delta_J$ relative distances of the vertices, that is a non-convex optimization problem.

Algorithm A.3 (CNO enclosing simplex)**.** *Consider the following optimisation*

$$\begin{aligned}
&\min_{\{\boldsymbol{\varphi_1},\ldots,\boldsymbol{\varphi_J}\}} cost(\delta_1, \ldots, \delta_J) \\
&\text{subject to} \\
&\quad [\mathbf{r}_1, \ldots, \mathbf{r}_J, \mathbf{w}(\mathbf{p})] = polytope(\boldsymbol{\varphi_1}, \ldots, \boldsymbol{\varphi_J}), \\
&\quad \delta_j = 1 - \max_{\mathbf{p}} w_j(\mathbf{p}),
\end{aligned}$$

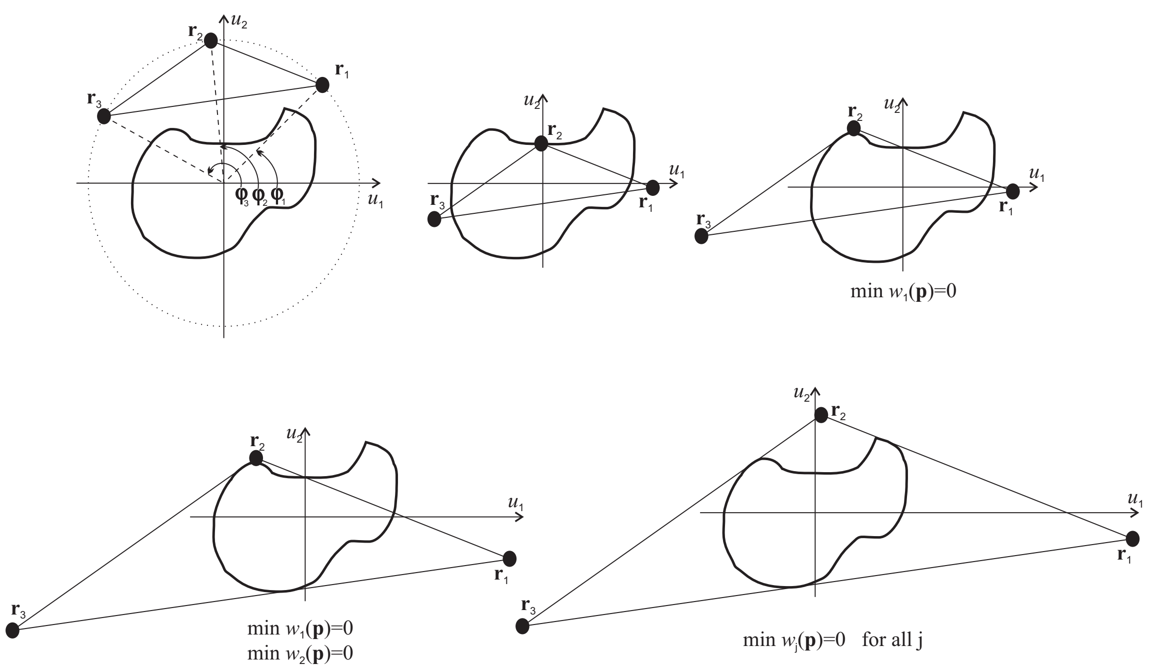


Figure A.3: Phases of CNO enclosing polytope generation for a $D = 2$ problem and given $\varphi_1, \varphi_2, \varphi_3$ values

where the polytope$(\boldsymbol{\varphi_1}, \ldots, \boldsymbol{\varphi_J})$ *function has the following pseudocode*

$$
\begin{aligned}
&\mathbf{r}_j = spherical2orthogonalcoord(\boldsymbol{\varphi_j}) \quad \forall j = 1..J \\
&\mathbf{r}_{mean} = \sum_{j=1}^{J} \mathbf{r}_j / J \\
&\mathbf{r}_j = \mathbf{r}_j - \mathbf{r}_{mean} \\
&\mathbf{w}(\mathbf{p}) = \begin{bmatrix}\mathbf{u}(\mathbf{p}) & 1\end{bmatrix} \begin{bmatrix}\mathbf{R} & \mathbf{1}\end{bmatrix}^{-1} \\
&\text{for } j = 1..J \\
&\qquad m = \min_{\mathbf{p}} w_j(\mathbf{p}) \\
&\qquad \mathbf{r}_i = \mathbf{r}_i - m\,(\mathbf{r}_i - \mathbf{r}_j) \quad \forall i = 1..J \\
&\qquad w_j(\mathbf{p}) = w_j(\mathbf{p}) - m \\
&\qquad \mathbf{w}(\mathbf{p}) = \mathbf{w}(\mathbf{p})/(1 - m)
\end{aligned}
$$

The problem is non-convex with a lot of local minima; the used algorithm combines the local minimization via simplex algorithm and random search from the best results.

The results are not repeatable because of the random search, and it has a high computational demand.

A.4 Proofs

Proof of Algorithm A.1. *The descriptions of SNNN algorithms begin with the centralization and reSVD of the HOSVD based weighting functions. It is easy to see that it results in a weighting function of an affine decomposition in the appropriate mode.*

Then the construction of $\mathbf{w}(\mathbf{p}) = \left[\mathbf{u}(\mathbf{p}) \quad 1 - \sum_d u_d(\mathbf{p})\right]$ *weighting functions correspond to the described initial simplex with vertices* $\mathbf{e}_1^D, \mathbf{e}_2^D, ..., \mathbf{e}_D^D, \mathbf{0}^D$ *that is not enclosing.*

Then the so-called SNNN enclosing is derived as

$$\mathbf{w}^{SNNN}(\mathbf{p}) = \mathbf{w}(\mathbf{p})\mathbf{T}, \quad \text{where} \quad \mathbf{T} = \frac{\mathbf{1}^{J\times J} + \zeta \mathbf{E}^J}{J+\zeta}, \ \zeta = -1/\min_{j,\mathbf{p}\in\Omega} w_j(\mathbf{p}). \tag{A.12}$$

It is easy to see, that $\zeta = J/\theta$*, and then the matrix can be written as*

$$\mathbf{T} = \frac{\mathbf{1}^{J\times J}\theta/J + \mathbf{E}^J}{\theta+1}, \tag{A.13}$$

and the weighting functions

$$\mathbf{w}^{SNNN}(\mathbf{p}) = \frac{\mathbf{1}^{1\times J}\theta/J + \mathbf{w}(\mathbf{p})}{\theta+1}. \tag{A.14}$$

That is resulted by central expansion see Step 2.■

Proof of Algorithm A.2. *For Step 1: trivial that in this case, the maximum of the weighting functions is the same, equal to one.*

For Step 2: First we prove, that the results of A,B,C,D steps result in polytopic description for $\mathbf{u}^{[a]}(\mathbf{p})$ *for all considered a value. After the A step:*

$$\begin{aligned}\sum_{i=1}^{a+1} \mathbf{r}_i^A w_i^A(\mathbf{p}) &= \sum_{i=1}^{a} \mathbf{r}_i^{[a]} \left(w_i^{[a]}(\mathbf{p}) - w_{a+1}^A(\mathbf{p})/a\right) + w_{a+1}^A(\mathbf{p}) \left(\bar{\mathbf{r}} - \alpha \mathbf{e}_{a+1}^D\right) = \sum_{i=1}^{a} \mathbf{r}_i^{[a]} w_i^{[a]}(\mathbf{p}) - \\ &- w_{a+1}^A(\mathbf{p}) \sum_{i=1}^{a} \mathbf{r}_i^{[a]}/a + w_{a+1}^A(\mathbf{p})\bar{\mathbf{r}} + \mathbf{e}_{a+1}^D u_{a+1}(\mathbf{p}) = \mathbf{u}^{[a]}(\mathbf{p}) + \mathbf{e}_{a+1}^D u_{a+1}(\mathbf{p}) = \mathbf{u}^{[a+1]}(\mathbf{p}).\end{aligned} \tag{A.15}$$

The weights are affine combinations trivially.

After the B step: By using the following notation $\gamma_i = \min_{\mathbf{p}} w_i^A(\mathbf{p})$ *then*

$$\sum_{i=1}^{a+1} \mathbf{r}_i^B w_i^B(\mathbf{p}) = \sum_{i=1}^{a+1} (\mathbf{r}_i^A + \Delta) w_i^B(\mathbf{p}) = \Delta + \sum_{i=1}^{a} \mathbf{r}_i^A (w_i^A(\mathbf{p}) - \gamma_i) + \left(w_{a+1}^A(\mathbf{p}) + \sum_{i=1}^{a} \gamma_i \right) \mathbf{r}_{a+1}^A =$$
$$= \Delta + \sum_{i=1}^{a+1} \mathbf{r}_i^A w_i^A(\mathbf{p}) - \sum_{i=1}^{a} \gamma_i (\mathbf{r}_i^A - \mathbf{r}_{a+1}^A) = \mathbf{u}^{[a+1]}(\mathbf{p}) + 0. \quad \text{(A.16)}$$

It is easy to see that $\min_{\mathbf{p}} w_i^B(\mathbf{p}) = 0$ *for all* $i = 1..a$*, this way the corresponding facets touch the image set of* $\mathbf{u}^{[a+1]}(\mathbf{p})$*. And the weighting functions denote affine combinations trivially.*

After the C step: the description is exact

$$\sum_{i=1}^{a+1} \mathbf{r}_i^C w_i^C(\mathbf{p}) = \sum_{i=1}^{a+1} \left(\mathbf{r}_i^B + \beta_i (\mathbf{r}_i^B - \mathbf{r}_{a+1}^B) \right) \frac{w_i^B(\mathbf{p})}{1+\beta_i} + \mathbf{r}_{a+1}^B \left(1 - \sum_{i=1}^{a} w_i^C(\mathbf{p}) \right) =$$
$$= \sum_{i=1}^{a} \mathbf{r}_i^B w_i^B(\mathbf{p}) + \mathbf{r}_{a+1}^B \left(1 - \sum_{i=1}^{a} \frac{\beta_i w_i^B(\mathbf{p})}{1+\beta_i} - \sum_{i=1}^{a} \frac{w_i^B(\mathbf{p})}{1+\beta_i} \right) = \sum_{i=1}^{a+1} \mathbf{r}_i^B w_i^B(\mathbf{p}) = \mathbf{u}^{[a+1]}(\mathbf{p}).$$
$$\text{(A.17)}$$

Furthermore the maximal value of the $w_i^C(\mathbf{p})$ *functions for* $i = 1..a$ *are one and they denote affine combinations trivially.*

After the D step: the description is exact, because

$$\sum_{i=1}^{a+1} \mathbf{r}_i^D w_i^D(\mathbf{p}) = \sum_{i=1}^{a} \left(\mathbf{r}_i^C (1+\gamma) - \gamma \mathbf{r}_{a+1}^C \right) \frac{w_i^C(\mathbf{p})}{1+\gamma} + \mathbf{r}_{a+1}^C \frac{w_{a+1}^C(\mathbf{p}) + \gamma}{1+\gamma} = \sum_{i=1}^{a} \mathbf{r}_i^C w_i^C +$$
$$+ \mathbf{r}_{a+1}^C \left(\frac{w_{a+1}^C(\mathbf{p}) + \gamma}{1+\gamma} - \frac{\gamma}{1+\gamma} \sum_{i=1}^{a} w_i^C(\mathbf{p}) \right) = \sum_{i=1}^{a} \mathbf{r}_i^C w_i^C(\mathbf{p}) +$$
$$+ \frac{\mathbf{r}_{a+1}^C}{1+\gamma} \left(w_{a+1}^C(\mathbf{p}) + \gamma - \gamma (1 - w_{a+1}^C(\mathbf{p})) \right) = \sum_{i=1}^{a+1} \mathbf{r}_i^C w_i^C(\mathbf{p}) = \mathbf{u}^{[a+1]}(\mathbf{p}). \quad \text{(A.18)}$$

The weighting functions with the previous steps become convex combinations, furthermore that maximal values of $w_i^D(\mathbf{p})$ *functions are the same for* $i = 1..a$*. It is easy to see that the difference of this value and the maximal value of* $w_{a+1}^D(\mathbf{p})$ *is monotone function of* α*, this way, bisection method can be applied to find the polytopic description such that they are equal.*

Appendix B

Extraction of multiple polytopic summation

The importance of polytopic models in control comes from the fact that the definite conditions on them (that can describe a large variety of control criteria) can be rewritten as a convex semidefinite program if the definite conditions consist of only single polytopic summations.

However, the definite criteria on expressions with multiple polytopic summations cannot be rewritten into LMIs that describe sufficient and necessary conditions. Only sufficient (and asymptotically necessary) criteria can be given for their use in control analysis and synthesis. This chapter introduces these methods.

B.1 Methods to extract double polytopic summation

Consider the double polytopic summation problem in the following form

$$\sum_{i=1}^{r}\sum_{j=1}^{r} h_i(\mathbf{p})h_j(\mathbf{p})\Gamma_{ij} \prec 0. \tag{B.1}$$

(For example the quadratic stability criteria for continuous time systems with a PDC controller can be written as $\Gamma_{ij} = \mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_j + (\mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_j)^T$.)

In the followings, the notation $h_i = h_i(\mathbf{p})$ will be used for the sake of brevity, and in a few cases, the following matrix notation

$$\sum_i\sum_j h_ih_j\Gamma_{ij} = \begin{bmatrix} \vdots \\ h_i \\ \vdots \end{bmatrix}^T \begin{bmatrix} & \vdots & \\ \dots & \Gamma_{ij} & \dots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ h_i \\ \vdots \end{bmatrix} = \mathbf{h}^T\Gamma\mathbf{h} \prec 0. \tag{B.2}$$

The first publications of Tanaka and Wang from 1995 [183] proposed the following simple lemma:

Lemma B.1 (Tanaka & Wang (1)). *The definit condition* $\sum_i \sum_j h_i h_j \Gamma_{ij} \prec 0$ *holds if*

$$\Gamma_{ii} \prec 0 \quad and \quad \Gamma_{ij} + \Gamma_{ji} \preceq 0 \qquad for\ all\ i = 1..r,\ j < i. \tag{B.3}$$

Then in 1998 they found the following relaxed method by introducing a positive semidefinite matrix [169] and by exploiting the following fact:

Lemma B.2. *If* $\mathbf{Q} = \mathbf{Q}^T \succeq 0$ *and*

$$\mathbf{Y} = \begin{bmatrix} (r-1)\mathbf{Q} & -\mathbf{Q} & \dots \\ -\mathbf{Q} & (r-1)\mathbf{Q} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}, \tag{B.4}$$

then $\mathbf{h}^T\mathbf{Y}\mathbf{h} \succeq 0$ *holds.*

Then, the resulting criteria:

Lemma B.3 (Tanaka & Wang (2)). *The definit condition* $\sum_i \sum_j h_i h_j \Gamma_{ij} \prec 0$ *holds if*

$$\mathbf{Q} = \mathbf{Q}^T \succeq 0, \quad \Gamma_{ii}+(r-1)\mathbf{Q} \prec 0 \quad and \quad \Gamma_{ij}+\Gamma_{ji}-2\mathbf{Q} \preceq 0 \qquad for\ all\ i = 1..r,\ j < i. \tag{B.5}$$

The amount of additional scalar variables is $k = \frac{n^2+n}{2}$, *because* $\mathbf{Q}$ *is symmetric.*

In 2001, Tuan & P. Apkarian published a similarly relaxed method without introducing slack variables and two other methods in [178]. Their results are based on the following lemma.

Lemma B.4. *Consider the convex summation problem:*

$$\begin{bmatrix} h_1 & h_2 \end{bmatrix} \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \prec 0 \tag{B.6}$$

This inequality holds if and only if there exists $\mathbf{Q} \succeq \mathbf{M}_{12}$ *such as*

$$\begin{bmatrix} h_1 & h_2 \end{bmatrix} \begin{bmatrix} \mathbf{M}_{11} & \mathbf{Q} \\ \mathbf{Q} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \prec 0. \tag{B.7}$$

It holds if

$$\mathbf{M}_{ii} \prec 0, \quad \mathbf{M}_{ii} + \mathbf{M}_{ij} \prec 0 \quad \forall i, j. \tag{B.8}$$

Using the Lemma B.4 Tuan et al. formed the following methods:

Lemma B.5 (Tuan (1)). *The definit condition* $\sum_i \sum_j h_i h_j \Gamma_{ij} \prec 0$ *holds if*

$$\Gamma_{ii} \prec 0 \quad and \quad \frac{1}{r-1}\Gamma_{ii} + \frac{1}{2}(\Gamma_{ij} + \Gamma_{ji}) \prec 0 \qquad for\ all\ i, j = 1..r,\ i \neq j. \tag{B.9}$$

Remark B.6 (Tuan (1)). *It is important to remark, that Lemma B.5 is less conservative than Lemma B.3 without applying additional variables.*

Lemma B.7 (Tuan (2)). *The definit condition* $\sum_i \sum_j h_i h_j \Gamma_{ij} \prec 0$ *holds if*

$$\begin{bmatrix} \frac{1}{r-1}\Gamma_{ii} & Q_{ij} \\ Q_{ij} & \frac{1}{r-1}\Gamma_{jj} \end{bmatrix} \prec 0 \quad and \quad \frac{1}{2}(\Gamma_{ij}+\Gamma_{ji}) \preceq Q_{ij} = Q_{ij}^T = Q_{ji} \qquad for\ all\ i = 1..r,\ j < i. \tag{B.10}$$

The amount of additional scalar variables is $k = \frac{n^2+n}{2}\frac{r^2+r}{2}$.

Lemma B.8 (Tuan (3)). *The definit condition* $\sum_i \sum_j h_i h_j \Gamma_{ij} \prec 0$ *holds if*

$$\Gamma_{ii} + \sum_{j \neq i}^{r} \mathbf{Q}_{ij} \prec 0 \qquad for\ all\ i = 1..r, \tag{B.11}$$

$$\mathbf{Q}_{ij} \succeq 0, \quad \mathbf{Q}_{ij} \succeq (\Gamma_{ij} + \Gamma_{ji})/2 \qquad for\ all\ i = 1..r,\ j = 1..(i-1). \tag{B.12}$$

E. Kim and H. Lee constructed the following lemma by applying more additional variables [84].

Lemma B.9 (Kim). *The definit condition* $\sum_i \sum_j h_i h_j \Gamma_{ij} \prec 0$ *holds if*

$$\mathbf{Z} \prec 0, \quad \Gamma_{ii} - \mathbf{Z}_{ii} \prec 0 \quad and \quad \Gamma_{ij} + \Gamma_{ji} - 2\mathbf{Z}_{ij} \preceq 0 \qquad for\ all\ i = 1..r,\ j < i, \tag{B.13}$$

where $\mathbf{Z}_{ij} = \mathbf{Z}_{ij}^T = \mathbf{Z}_{ji}$ *and*

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \dots \\ \mathbf{Z}_{12} & \mathbf{Z}_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}. \tag{B.14}$$

The amount of additional scalar variables is $k = \frac{n^2+n}{2}\frac{r^2+r}{2}$.

X. Liu és Q. Zhang relaxed the method in 2002 by increasing the number of additional variables [118].

Lemma B.10. *The definit condition* $\sum_i \sum_j h_i h_j \Gamma_{ij} \prec 0$ *holds if*

$$\mathbf{Z} \prec 0, \quad \Gamma_{ii} - \mathbf{Z}_{ii} \prec 0 \quad and \quad \Gamma_{ij} + \Gamma_{ji} - \mathbf{Z}_{ij} - \mathbf{Z}_{ji} \preceq 0 \qquad for\ all\ i = 1..r,\ j < i, \tag{B.15}$$

where $\mathbf{Z}_{ij} = \mathbf{Z}_{ji}^T$ *and*

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \dots \\ \mathbf{Z}_{12} & \mathbf{Z}_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}. \tag{B.16}$$

The amount of additional scalar variables is $k = \frac{(nr)^2+nr}{2}$.

C. Fang, Y. Liu et al. showed [61] that three times polytopic summations can be handled as:

Lemma B.11. *Consider the following matrix given by triple convex sum*

$$\sum_{i=1}^{r}\sum_{j=1}^{r}\sum_{l=1}^{r} h_i h_j h_l \mathbf{Y}_{ijl} = \mathcal{Y} \overset{3}{\underset{1}{\boxtimes}} \mathbf{h}. \tag{B.17}$$

It can be rewritten as

$$\sum_{i=1}^{r} h_i^3 \mathbf{Y}_{iii} + \sum_{i=1}^{r}\sum_{j=1, i\neq j}^{r} h_i^2 h_j (\mathbf{Y}_{iij} + \mathbf{Y}_{iji} + \mathbf{Y}_{jii}) + \\ + \sum_{i=1}^{r}\sum_{j<i}\sum_{l<j} h_i h_j h_l (\mathbf{Y}_{ijl} + \mathbf{Y}_{jli} + \mathbf{Y}_{lij} + \mathbf{Y}_{lji} + \mathbf{Y}_{jil} + \mathbf{Y}_{ilj}). \tag{B.18}$$

Double polytopic summations can be relaxed, by adding a third summation and applying Lemma B.11.

Lemma B.12 (Fang (1))**.** *The definit condition* $\sum_i \sum_j h_i h_j \Gamma_{ij} \prec 0$ *holds if*

$$\Gamma_{ii} \prec 0, \qquad \Gamma_{ii} + \Gamma_{ij} + \Gamma_{ji} \preceq 0 \qquad \textit{for all } i = 1..r,\ j = 1..r,\ i \neq j, \textit{ and} \tag{B.19}$$

$$\Gamma_{ij} + \Gamma_{ji} + \Gamma_{il} + \Gamma_{li} + \Gamma_{jl} + \Gamma_{lj} \preceq 0 \qquad \textit{for all } i = 1..r,\ j < i,\ l < j. \tag{B.20}$$

The amount of additional scalar variables is $k = 0$*.*

The method can be further relaxed by applying additional variables.

Lemma B.13 (Fang (1))**.** *The definit condition* $\sum_i \sum_j h_i h_j \Gamma_{ij} \prec 0$ *holds if*

$$\Gamma_{ii} + \mathbf{Y}_{iii} \prec 0, \qquad \Gamma_{ii} + \Gamma_{ij} + \Gamma_{ji} + \mathbf{Y}_{jii} + \mathbf{Y}_{jii}^T + \mathbf{Y}_{iji} \preceq 0 \quad \forall\ i = 1..r,\ j = 1..r,\ i \neq j,$$
$$\Gamma_{ij} + \Gamma_{ji} + \Gamma_{il} + \Gamma_{li} + \Gamma_{jl} + \Gamma_{lj} + \mathrm{He}(\mathbf{Y}_{ijl} + \mathbf{Y}_{jli} + \mathbf{Y}_{lij}) \preceq 0,\ \forall i, j < i, l < j.$$
$$\mathbf{Y}_i = \left[\mathbf{Y}_{:,i,:}\right] = \mathbf{Y}_i^T \succeq 0.$$

The amount of additional scalar variables is $k = R\frac{(Rn)^2 + Rn}{2}$*.*

B.2 Pólya-theorem based method to extract multiple polytopic summations

First, let recall the Pólya-theorem.

Theorem B.14 (Pólya-theorem)**.** *Let consider the real, homogeneous*

$$f(\mathbf{h}) = \sum_{r_1} \cdots \sum_{r_L} f_{r_1,\dots,r_L} \prod_{l=1}^{L} h_{r_l} \tag{B.21}$$

polynomial scalar function, which is positive $\forall h_r \geq 0, \sum_r h_r = 1$. *Then for a sufficiently large* $M \in \mathbb{Z}^+$, *the product*

$$\left(\sum_r h_r\right)^M \sum_{r_1} \cdots \sum_{r_L} f_{r_1,\ldots,r_L} \prod_{l=1}^{L} h_{r_l} \tag{B.22}$$

has only strictly positive coefficients.

The definiteness of matrix coefficients behaves similarly, this way, by increasing the multiplicity of polytopic summations and investigate the matrix coefficients the conservativeness can be decreased [150].

Lemma B.15. *Consider the following* L *times definite condition*

$$\sum_{r_1} \cdots \sum_{r_L} \Gamma_{r_1,\ldots,r_L} \prod_{l=1}^{L} h_{r_l} \prec 0 \tag{B.23}$$

in general and a $K \geq L$ *integer. The condition holds if* $\forall r_1 \leq r_2 \leq \cdots \leq r_K$

$$\begin{aligned}
&\mathbf{Y}_{r_1,\ldots,r_K} \prec 0 \quad if \ r_1 = r_2 = \cdots = r_K, \\
&\mathbf{Y}_{r_1,\ldots,r_K} \preceq 0 \quad otherwise,
\end{aligned}$$

where

$$\begin{aligned}
&\mathbf{Y}_{r_1,\ldots,r_K} = {\Gamma'}_{a_1^{(1)},a_2^{(1)},\ldots,a_K^{(1)}} + {\Gamma'}_{a_1^{(2)},a_2^{(2)},\ldots,a_K^{(2)}} + \cdots + {\Gamma'}_{a_1^{(M^{(r_1,\ldots,r_K)})},a_{r_2}^{(M^{(r_1,\ldots,r_K)})},\ldots,a_K^{(M^{(r_1,\ldots,r_K)})}}, \\
&{\Gamma'}_{a_1,a_2,\ldots,a_K} = \Gamma_{a_1,a_2,\ldots,a_L}, \\
&(a_1^{(i)}, a_2^{(i)}, \ldots, a_K^{(i)}) = \mathrm{reorder}(r_1, \ldots, r_K), \\
&(a_1^{(i)}, a_2^{(i)}, \ldots, a_K^{(i)}) \neq (a_1^{(r)}, a_2^{(r)}, \ldots, a_K^{(r)}) \ if \ i \neq r, \\
&M^{(r_1,\ldots,r_K)} = \frac{K!}{m_1!m_2!\ldots} (where \ m_i - s \ are \ the \ multiplicities \ of \ (r_1, \ldots, r_K) indices),
\end{aligned}$$

furthermore, as $K \to \infty$ *the condition tends to be necessary (theoretically).*

B.3 Proofs

Proof of Lemma B.1. *Because the summmation can be written as*

$$\sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j \Gamma_{ij} = \sum_{i=1}^{r} h_i^2 \Gamma_{ii} + \sum_{i=1}^{r} \sum_{i<j} h_i h_j (\Gamma_{ij} + \Gamma_{ji}), \tag{B.24}$$

the lemma is proofed. ■

Proof of Lemma B.2. *Consider the expression*

$$\sum_{i=1}^{r}\sum_{j<i}(h_i - h_j)^2 \geq 0. \tag{B.25}$$

Because it can be written as

$$\sum_{i=1}^{r}\sum_{j<i}(h_i - h_j)^2 = \sum_{i=1}^{r}\sum_{j<i}(h_i^2 + h_j^2) - \sum_{i=1}^{r}\sum_{j<i}2h_ih_j,$$

where

$$\sum_{i=1}^{r}\sum_{j<i}(h_i^2 + h_j^2) = (r-1)\sum_{i=1}^{r}h_i^2, \tag{B.26}$$

so

$$\sum_{i=1}^{r}\sum_{j<i}(h_i - h_j)^2 = (r-1)\sum_{i=1}^{r}h_i^2 - \sum_{i=1}^{r}\sum_{j<i}2h_ih_j.$$

Then for all $\mathbf{Q} \succeq 0$ *matrix*

$$(r-1)\sum_{i=1}^{r}h_i^2\mathbf{Q} - \sum_{i=1}^{r}\sum_{j<i}2h_ih_j\mathbf{Q} \succeq 0.$$

that is same as $\mathbf{h}^T\mathbf{Y}\mathbf{h} \succeq 0$*.* ■

Proof of Lemma B.3. *If the conditions holds, then* $\mathbf{h}^T\Gamma\mathbf{h} + \mathbf{h}^T\mathbf{Y}\mathbf{h} \prec 0$*, where* $\mathbf{Y}$ *is constructed as in Lemma B.2. and* $\mathbf{h}^T\mathbf{Y}\mathbf{h} \succeq 0$*. Then it is trivial, that* $\mathbf{h}^T\Gamma\mathbf{h} \prec 0$*.* ■

Proof of Lemma B.4. *I.* (B.6)$\to$(B.7)*: it is trivial, applying* $\mathbf{Q} = \mathbf{M}_{12}$*.*

II. (B.7)$\to$ (B.6)*: if* (B.7) *holds:*

$$h_1^2\mathbf{M}_{11} + h_2^2\mathbf{M}_{22} + 2h_1h_2\mathbf{Q} \prec 0. \tag{B.27}$$

If $\mathbf{M}_{12} \preceq \mathbf{Q}$ *it is trivial that* (B.6) *holds.*

III. (B.8)$\to$ (B.6) *If* $\mathbf{M}_{12} \preceq 0$ *(and* $\mathbf{M}_{ii} \prec 0$*) it is trivial that* $h_1^2\mathbf{M}_{11} + h_2^2\mathbf{M}_{22} + 2h_1h_2\mathbf{M}_{12} \prec 0$*. If* $\mathbf{M}_{12} \succ 0$ *the following eq. holds:*

$$h_1^2(\mathbf{M}_{11} + \mathbf{M}_{12}) + h_2^2(\mathbf{M}_{22} + \mathbf{M}_{12}) \prec 0 \tag{B.28}$$

and then

$$h_1^2\mathbf{M}_{11} + h_2^2\mathbf{M}_{22} + 2h_1h_2\mathbf{M}_{12} \preceq h_1^2(\mathbf{M}_{11} + \mathbf{M}_{12}) + h_2^2(\mathbf{M}_{22} + \mathbf{M}_{12}) \prec 0 \tag{B.29}$$

because it can be rewritten as $0 \preceq (h_1 - h_2)^2\mathbf{M}_{12}$*.* ■

Proof of Lemma B.5. *If the conditions holds then*

$$\begin{bmatrix} h_i & h_j \end{bmatrix} \begin{bmatrix} \Gamma_{ii}/(r-1) & \Gamma_{ij} \\ \Gamma_{ji} & \Gamma_{jj}/(r-1) \end{bmatrix} \begin{bmatrix} h_i \\ h_j \end{bmatrix} \prec 0 \quad \forall i,j \tag{B.30}$$

based on lemma B.4. Summed the expressions:

$$\sum_{i=1}^{r} \sum_{j<i} ((h_i^2 \Gamma_{ii} + h_j^2 \Gamma_{jj})/(r-1) + h_i h_j (\Gamma_{ij} + \Gamma_{ji})) \prec 0. \tag{B.31}$$

Because based on equation (B.26)

$$\sum_{i=1}^{r} \sum_{j<i} (h_i^2 \Gamma_{ii} + h_j^2 \Gamma_{jj}) = (r-1) \sum_{i=1}^{r} h_i^2 \Gamma_{ii}, \tag{B.32}$$

the condition holds:

$$\sum_{i=1}^{r} h_i^2 \Gamma_{ii} + \sum_{i=1}^{r} \sum_{j<i} h_i h_j (\Gamma_{ij} + \Gamma_{ji}) = \mathbf{h}^T \Gamma \mathbf{h} \prec 0. \blacksquare \tag{B.33}$$

Proof of Lemma B.7. *The same as the previous one.* $\blacksquare$

Proof of Lemma B.8. *If* $\mathbf{Q}_{ij} \succeq 0$

$$0 \preceq \sum_{i=1}^{r} \sum_{j=1}^{r} (h_i - h_j)^2 \mathbf{Q}_{ij}, \tag{B.34}$$

(where $\mathbf{Q}_{ii} = 0$.*) Then because*

$$\begin{aligned} \sum_{i=1}^{r} \sum_{j=1}^{r} (h_i - h_j)^2 \mathbf{Q}_{ij} &= \sum_{i=1}^{r} \sum_{j=1}^{r} (h_i^2 + h_j^2) \mathbf{Q}_{ij} - \sum_{i=1}^{r} \sum_{j=1}^{r} 2 h_i h_j \mathbf{Q}_{ij} = \\ &= \sum_{i=1}^{r} \sum_{j=1}^{r} 2 h_i^2 \mathbf{Q}_{ij} - \sum_{i=1}^{r} \sum_{j=1}^{r} 2 h_i h_j \mathbf{Q}_{ij}, \end{aligned} \tag{B.35}$$

so

$$2 \sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j \mathbf{Q}_{ij} \preceq 2 \sum_{i=1}^{r} \sum_{j=1}^{r} h_i^2 \mathbf{Q}_{ij}. \tag{B.36}$$

From the other conditions

$$2\sum_i \sum_j h_i^2 \mathbf{Q}_{ij} \prec -\sum_i 2h_i^2 \Gamma_{ii} \tag{B.37}$$

$$\sum_{i=1}^{r} \sum_{j=1,j\neq i}^{r} h_i h_j (\Gamma_{ij} + \Gamma_{ji}) \preceq 2\sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j \mathbf{Q}_{ij} \tag{B.38}$$

so

$$\sum_{i=1}^{r} \sum_{j=1,j\neq i}^{r} h_i h_j (\Gamma_{ij} + \Gamma_{ji}) \preceq \ldots \preceq \ldots \prec -\sum_i 2h_i^2 \Gamma_{ii}. \tag{B.39}$$

this way $\mathbf{h}^T \Gamma \mathbf{h} \prec 0$ *is proved.* ■

Proof of Lemma B.9. *If the condition holds*

$$\sum_i h_i^2 (\Gamma_{ii} - \mathbf{Z}_{ii}) \prec 0, \quad \sum_i \sum_{j<i} h_i h_j (\Gamma_{ij} + \Gamma_{ji} - 2\mathbf{Z}_{ij}) \preceq 0 \tag{B.40}$$

and

$$\sum_i h_i^2 (\Gamma_{ii} - \mathbf{Z}_{ii}) + \quad \sum_i \sum_{j<i} h_i h_j (\Gamma_{ij} + \Gamma_{ji} - 2\mathbf{Z}_{ij}) = \mathbf{h}^T \Gamma \mathbf{h} - \mathbf{h}^T \mathbf{Z} \mathbf{h} \prec 0. \tag{B.41}$$

Because $-\mathbf{h}^T \mathbf{Z} \mathbf{h} \succ 0$, $\mathbf{h}^T \Gamma \mathbf{h} \prec 0$. ■

Proof of Lemma B.10. *The same as one of lemma B.9.* ■

Proof of Lemma B.11. *Trivial.* ■

Proof of Lemma B.12. *If the conditions hold*

$$\sum_{i=1}^{r} h_i^3 \Gamma_{ii} + \sum_{i=1}^{r} \sum_{j=1,i\neq j}^{r} h_i^2 h_j (\Gamma_{ii} + \Gamma_{ij} + \Gamma_{ji}) +$$
$$+ \sum_{i=1}^{r} \sum_{j<i} \sum_{l<j} h_i h_j h_l (\Gamma_{ij} + \Gamma_{jl} + \Gamma_{li} + \Gamma_{lj} + \Gamma_{ji} + \Gamma_{il}) \prec 0. \tag{B.42}$$

Then based on Lemma B.11,

$$\sum_{i=1}^{r} \sum_{j=1}^{r} \sum_{l=1}^{r} h_i h_j h_l \Gamma_{ij} = \sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j \Gamma_{ij} \prec 0. ■ \tag{B.43}$$

Proof of Lemma B.13. *If the conditions hold*

$$\sum_{i=1}^{r} h_i^3(\Gamma_{ii} + \mathbf{Y}_{iii}) + \sum_{i=1}^{r} \sum_{j=1, i\neq j}^{r} h_i^2 h_j(\Gamma_{ii} + \Gamma_{ij} + \Gamma_{ji} + \mathbf{Y}_{jii} + \mathbf{Y}_{iij} + \mathbf{Y}_{iji}) +$$
$$+ \sum_{i=1}^{r} \sum_{j<i} \sum_{l<j} h_i h_j h_l (\Gamma_{ij} + \Gamma_{jl} + \Gamma_{li} + \Gamma_{lj} + \Gamma_{ji} + \Gamma_{il} + \mathbf{Y}_{ijl} + \mathbf{Y}_{jli} + \mathbf{Y}_{lij} + \mathbf{Y}_{jil} + \mathbf{Y}_{lji} + \mathbf{Y}_{ilj}) \prec 0. \tag{B.44}$$

Then based on Lemma B.11,

$$\sum_{i=1}^{r} \sum_{j=1}^{r} \sum_{l=1}^{r} h_i h_j h_l (\Gamma_{ij} + \mathbf{Y}_{ijl}) = \sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j \Gamma_{ij} + \sum_{i=1}^{r} h_i \mathbf{h}^T \mathbf{Y}_i \mathbf{h} \prec 0. \tag{B.45}$$

Because $\mathbf{Y}_i$ *are positive semi-definite*

$$\sum_{i=1}^{r} h_i \mathbf{h}^T \mathbf{Y}_i \mathbf{h} \succeq 0, \tag{B.46}$$

so

$$\sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j \Gamma_{ij} \prec 0. \blacksquare \tag{B.47}$$

Proof of Lemma B.15. *It is the generalisation of Lemma B.1, Lemma B.11 and Lemma B.12.* ■

The author's publications

Journal papers

[89] József Kuti and Péter Galambos. "Affine Tensor Product Model Transformation". en. In: *Complexity* (2018), pp. 1–12.

[96] József Kuti, Péter Galambos, and Péter Baranyi. "Minimal Simplex (MVS) Polytopic Model Generation and Manipulation Methodology for TP Model Transformation". en. In: *Asian Journal of Control* 19.1 (Jan. 2017), pp. 289–301.

[101] József Kuti, Péter Galambos, and Ákos Miklós. "Output Feedback Control of a Dual-Excenter Vibration Actuator via qLPV Model and TP Model Transformation". en. In: *Asian Journal of Control* 17.2 (Mar. 2015), pp. 432–442.

Book chapter

[94] József Kuti, Péter Galambos, and Péter Baranyi. "Delay and Stiffness Dependent Polytopic LPV Modelling of Impedance Controlled Robot Interaction". In: *Issues and Challenges of Intelligent Systems and Computational Intelligence*. Ed. by László T. Kóczy, Claudiu R. Pozna, and Janusz Kacprzyk. Vol. 530. Cham: Springer International Publishing, 2014, pp. 163–174.

Conference papers

[64] Péter Galambos, József Kuti, and Péter Baranyi. "On the Tensor Product based qLPV modeling of bio-inspired dynamic processes". In: *IEEE 17th Int. Conf. Intell. Eng. Syst. (INES), 2013*. 2013, pp. 79–84.

[65] Peter Galambos, Jozsef Kuti, Peter Baranyi, Gabor Szogi, and Imre J. Rudas. "Tensor Product Based Convex Polytopic Modeling of Nonlinear Insulin-Glucose Dynamics". In: *IEEE Int. Conf. Syst. Man and Cybernatics (SMC)*. Oct. 2015, pp. 2597–2602.

[82] A. I. Károly, J. Kuti, and P. Galambos. "Unsupervised real-time classification of cycle stages in collaborative robot applications". In: *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. Feb. 2018, pp. 000097–000102.

[90] József Kuti and Péter Galambos. “Computational Relaxations for Affine Tensor Product Model Transformation”. In: *IEEE 15th Int. Symp. Intell. Syst. and Inf. (SISY)*. 2017, pp. 79–84.

[91] József Kuti and Péter Galambos. “Control Design-oriented {qLPV} Modelling of Virtual Impedance-based Telemanipulation”. In: *IEEE 30th Jubilee Neumann Colloquium*. IEEE, 2017, pp. 131–134.

[92] József Kuti, Péter Galambos, and Péter Baranyi. “Control Analysis and Synthesis through Polytopic Tensor Product Model: a General Concept”. In: *Proc. of the Int. Fed. of Aut. Contr. (IFAC)*. 2017, pp. 6742–6747.

[93] József Kuti, Péter Galambos, and Péter Baranyi. “Delay and Stiffness Dependent Polytopic LPV Model of Impedance Controlled Robot Interaction”. In: *Fifth Győr Symp. and First Hung.-Pol. Joint Conf. Comput. Intell.* 2012, pp. 174–175.

[95] József Kuti, Péter Galambos, and Péter Baranyi. “Generalization of Tensor Product Model Transformation based Control Design Concept”. In: *Proceedings of the Int. Fed. of Automatic Contr. (IFAC)*. 2017, pp. 5769–5774.

[97] József Kuti, Péter Galambos, and Péter Baranyi. “Minimal Volume Simplex (MVS) approach for convex hull generation in TP Model Transformation”. In: *18th Int. Conf. Intell. Eng. Syst. (INES)*. July 2014, pp. 187–192.

[98] József Kuti, Péter Galambos, and Péter Baranyi. “Non-simplex Enclosing Polytope Generation Concept for Tensor Product Model Transformation based Controller Design”. In: *IEEE Int. Conf. Syst. Man and Cybernatics (SMC)*. 2016, pp. 3368–3373.

[99] József Kuti, Péter Galambos, and Péter Baranyi. “Time-delay and stiffness dependant polytopic (LPV) modelling of impedance model based robot interaction”. In: *Erdélyi Múzeum-Egyesület*. In Hungarian. 2013.

[100] József Kuti, Péter Galambos, Péter Baranyi, and Péter Várlaki. “A Hands-On Demonstration of Control Performance Optimization Using Tensor Product Model Transformation and Convex Hull Manipulation”. In: *IEEE Int. Conf. Syst. Man and Cybernatics (SMC)*. Oct. 2015, pp. 2609–2614.

[164] Árpad Takács, Tamás Haidegger, Péter Galambos, József Kuti, and Imre J. Rudas. “Nonlinear soft tissue mechanics based on polytopic Tensor Product modeling”. In: *IEEE 14th Int. Symp. Appl. Mach. Intell. and Inf. (SAMI)*. 2016, pp. 211–215.

[165] Árpád Takács, József Kuti, Tamas Haidegger, Péter Galambos, and Imre Rudas. “Polytopic Model Based Interaction Control for Soft Tissue Manipulation”. In: *IEEE Int. Conf. Syst. Man and Cybernatics (SMC)*. 2016, pp. 3899–3905.

References

[1] Jurgen Ackermann. "Der entwurf linearer regelungssysteme im zustandsraum". In: *at-Automatisierungstechnik* 20.1-12 (1972), pp. 297–300.

[2] A. Agathos, Jun Li, D. Petcu, and A. Plaza. "Multi-GPU Implementation of the Minimum Volume Simplex Analysis Algorithm for Hyperspectral Unmixing". In: *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* 7.6 (June 2014), pp. 2281–2296.

[3] Alexander Agathos, Jun Li, José M Bioucas-Dias, and Antonio Plaza. "Robust minimum volume simplex analysis for hyperspectral unmixing". In: *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 22nd European.* IEEE. 2014, pp. 1582–1586.

[4] Farid Alizadeh. "Interior point methods in semidefinite programming with applications to combinatorial optimization". In: *SIAM Journal on Optimization* 5.1 (1995), pp. 13–51.

[5] Erling D Andersen, Cornelis Roos, and Tamas Terlaky. "On implementing a primal-dual interior-point method for conic quadratic optimization". In: *Mathematical Programming* 95.2 (2003), pp. 249–277.

[6] Martin Andersen, Joachim Dahl, Zhang Liu, and Lieven Vandenberghe. "Interior-point methods for large-scale cone programming". In: *Optimization for machine learning* (2011), pp. 55–83.

[7] Carlos Ariño and Antonio Sala. "Relaxed LMI conditions for closed-loop fuzzy systems with tensor-product structure". In: *Eng. App. of Artificial Intell.* 20.8 (2007), pp. 1036–1046.

[8] Michael Athans, David Castanon, K-P Dunn, C Greene, Wing Lee, N Sandell, and A Willsky. "The stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method–Part I: Equilibrium flight". In: *IEEE Transactions on Automatic Control* 22.5 (1977), pp. 768–780.

[9] David Avis and David Bremner. "How Good Are Convex Hull Algorithms?" In: *Proceedings of the Eleventh Annual Symposium on Computational Geometry.* SCG '95. Vancouver, British Columbia, Canada: ACM, 1995, pp. 20–28.

[10] Gary Balas, József Bokor, and Zoltán Szabó. "Invariant subspaces for LPV systems and their applications". In: *IEEE Transactions on Automatic Control* 48.11 (2003), pp. 2065–2069.

[11] Gary Balas, Richard Chiang, Andy Packard, and Michael Safonov. "Robust control toolbox". In: *For Use with Matlab. User's Guide, Version* 3 (2005).

[12] P. Baranyi. “Convex hull generation methods for polytopic representations of LPV models”. In: *Applied Machine Intelligence and Informatics, 2009. SAMI 2009. 7th International Symposium on*. Jan. 2009, pp. 69–74.

[13] P. Baranyi. “TP Model Transformation as a Way to LMI-Based Controller Design”. In: *IEEE Transactions on Industrial Electronics* 51.2 (Apr. 2004), pp. 387–400.

[14] P. Baranyi, L. Szeidl, and P. Várlaki. “Numerical reconstruction of the HOSVD based canonical form of polytopic dynamic models”. In: *Intelligent Engineering Systems, 2006. INES'06. Proceedings. International Conference on*. 2006, pp. 196–201.

[15] P. Baranyi, Y. Yam, and P. Várlaki. *Tensor Product Model Transformation in Polytopic Model-Based Control*. Automation and Control Engineering. Taylor & Francis Group, 2013.

[16] Peter Baranyi. “The Generalized TP model transformation for TS fuzzy model manipulation and generalized stability verification”. In: *IEEE Transactions on Fuzzy Systems* (2013), pp. 934–948.

[17] Péter Baranyi. “Output Feedback Control of 2-D Aeroelastic System”. In: *Journal of Guidance, Control, and Dynamics* 29.3 (2006), pp. 762–767.

[18] Péter Baranyi. *TP-model Transformation-based-control Design Frameworks*. Springer, 2016.

[19] Peter Baranyi, Domonkos Tikk, Yeung Yam, and Ron J. Patton. “From differential equations to PDC controller design via numerical transformation”. In: *Computers in Industry* 51.3 (Aug. 2003), pp. 281–297.

[20] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. “The Quickhull algorithm for convex hulls”. In: *ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE* 22.4 (1996), pp. 469–483.

[21] Zachary Battles. “Numerical linear algebra for continuous functions”. PhD thesis. University of Oxford, 2005.

[22] Richard Bellman. “Dynamic programming and Lagrange multipliers”. In: *Proceedings of the National Academy of Sciences* 42.10 (1956), pp. 767–769.

[23] Fernando D Bianchi, Ricardo J Mantz, and Carlos F Christiansen. “Multivariable PID control with set-point weighting via BMI optimisation”. In: *Automatica* 44.2 (2008), pp. 472–478.

[24] David Bindel, James Demmel, William Kahan, and Osni Marques. “On computing Givens rotations reliably and efficiently”. In: *ACM Trans. on Math. Software (TOMS)* 28.2 (2002), pp. 206–238.

[25] José M. Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches”. In: *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* 5.2 (2012), pp. 354–379.

[26] Chris C Bissell. “Stodola, Hurwitz and the genesis of the stability criterion”. In: *International Journal of Control* 50.6 (1989), pp. 2313–2332.

[27] Ake Björck. “Numerics of gram-schmidt orthogonalization”. In: *Lin. Alg. and Its Appl.* 197 (1994), pp. 297–316.

[28] Hendrik W Bode et al. “Network analysis and feedback amplifier design”. In: (1945).

[29] Paul T. Boggs and Jon W. Tolle. “Sequential quadratic programming”. In: *Acta numerica* 4 (1995), pp. 1–51.

[30] József Bokor and Gary Balas. “Linear parameter varying systems: A geometric theory and applications”. In: *IFAC Proceedings Volumes* 38.1 (2005), pp. 12–22.

[31] T. Bouarar, K. Guelton, and N. Manamanni. “Robust fuzzy Lyapunov stabilization for uncertain and disturbed Takagi–Sugeno descriptors”. en. In: *ISA Transactions* 49.4 (Oct. 2010), pp. 447–461.

[32] Stephen P Boyd. *Linear matrix inequalities in system and control theory.* Philadelphia: Society for Industrial and Applied Mathematics, 1994.

[33] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization.* English. Cambridge, UK; New York: Cambridge University Press, 2004.

[34] Arne Brøndsted. *An introduction to convex polytopes*. Springer, 1983.

[35] R. T. Bupp, D. S. Bernstein, and V. T. Coppola. “A benchmark problem for nonlinear control design: problem statement, experimental testbed, and passive nonlinear compensation”. In: *Proceedings of 1995 American Control Conference - ACC’95*. Institute of Electrical & Electronics Engineers (IEEE), 1995, pp. 4363–4367.

[36] R. T. Bupp, D. S. Bernstein, and V. T. Coppola. “Vibration suppression of multi-modal translational motion using a rotational actuator”. In: *Proceedings of 1994 33rd IEEE Conference on Decision and Control.* Institute of Electrical & Electronics Engineers (IEEE), 1994.

[37] Robert T Bupp, Dennis S Bernstein, and T VINCENT. “A benchmark problem for nonlinear control design”. In: *Int. J. Robust Nonlinear Control* 307 (1998), pp. 307–310.

[38] Robert T Bupp, Chih-Jian Wan, Vincent T Coppola, and Dennis S Bernstein. “Design of a rotational actuator for global stabilization of translational motion”. In: *Proc. ASME.* 1994, pp. 449–456.

[39] Renan Cabrera, Traci Strohecker, and Herschel Rabitz. “The canonical coset decomposition of unitary matrices through Householder transformations”. In: *J. Math. Phys.* 51.8 (2010), pp. 82–101.

[40] Víctor Costa da Silva Campos, Leonardo Antônio Borges Tôrres, and Reinaldo Martinez Palhares. “Revisiting the TP Model Transformation: Interpolation and Rule Reduction”. In: *Asian Journal of Control* (2014).

[41] Francesco Casella and Marco Lovera. “LPV/LFT modelling and identification: overview, synergies and a case study”. In: *Computer-Aided Control Systems, 2008. CACSD 2008. IEEE International Conference on*. IEEE. 2008, pp. 852–857.

[42] Tsung-Han Chan, Chong-Yung Chi, Yu-Min Huang, and Wing-Kin Ma. “A Convex Analysis-Based Minimum-Volume Enclosing Simplex Algorithm for Hyperspectral Unmixing”. In: *IEEE Transactions on Signal Processing* 57.11 (Nov. 2009), pp. 4418–4432.

[43] NG Chetaev. “Stability of Motion [in Russian], Gostekhizdat, Moscow (1955)”. In: (1959).

[44] Chong-Yung Chi. “Non-negative least-correlated component analysis for separation of dependent sources”. In: *invited talk at the Workshop on Optimization and Signal Processing, The Chinese University Hong Kong, Hong Kong*. 2007.

[45] Mahmoud Chilali and Pascal Gahinet. “H_∞ design with pole placement constraints: an LMI approach”. In: *IEEE Transactions on Automatic Control* 41.3 (1996), pp. 358–367.

[46] Mahmoud Chilali, Pascal Gahinet, and Pierre Apkarian. “Robust pole placement in LMI regions”. In: *IEEE Transactions on Automatic Control* 44.12 (1999), pp. 2257–2270.

[47] Cesar AR Crusius and Alexandre Trofino. “Sufficient LMI conditions for output feedback control problems”. In: *IEEE Transactions on Automatic Control* 44.5 (1999), pp. 1053–1057.

[48] N. Daraoui, M. Nachidi, F. Tadeo, and A. Benzaouia. “LMI approach for exponential stabilization of continuous-time delayed Takagi-Sugeno systems”. In: *World Congress*. Vol. 18. 2011, pp. 3873–3878.

[49] L. De Lathauwer, B. De Moor, and J. Vandewalle. “A multilinear singular value decomposition”. In: *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1253–1278.

[50] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. “On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors”. In: *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1324–1342.

[51] Baocang Ding. “New Formulation of Dynamic Output Feedback Robust Model Predictive Control with Guaranteed Quadratic Boundedness: Output Feedback MPC”. In: *Asian Journal of Control* 15.1 (Jan. 2013), pp. 302–309.

[52] BaoCang Ding, HeXu Sun, and Peng Yang. “Further studies on LMI-based relaxed stabilization conditions for nonlinear systems in Takagi–Sugeno’s form”. In: *Automatica* 42.3 (Mar. 2006), pp. 503–508.

[53] Jiuxiang Dong and Guang-Hong Yang. “Robust static output feedback control synthesis for linear continuous systems with polytopic uncertainties”. en. In: *Automatica* 49.6 (June 2013), pp. 1821–1829.

[54] John C Doyle, Keith Glover, Pramod P Khargonekar, and Bruce A Francis. "State-space solutions to standard H_2 and H_∞ control problems". In: *IEEE Trans. Aut. Control* 34.8 (1989), pp. 831–847.

[55] John C Doyle and Gunter Stein. "Multivariable feedback design: Concepts for a classical/modern synthesis". In: *IEEE Trans. on Aut. Control*. 1981.

[56] John Doyle et al. "Analysis of feedback systems with structured uncertainties". In: *IEE proceedings*. Vol. 129. 6. 1982, pp. 242–250.

[57] Carl Eckart and Gale Young. "The approximation of one matrix by another of lower rank". In: *Psychometrika* 1.3 (1936), pp. 211–218.

[58] György Eigner, István Böjthe, Péter Pausits, and Levente Kovács. "Investigation of the TP modeling possibilities of the Hovorka T1DM model". In: *Applied Machine Intelligence and Informatics (SAMI), 2017 IEEE 15th International Symposium on*. IEEE. 2017, pp. 259–264.

[59] György Eigner, Imre J Rudas, and Levente Kovács. "Investigation of the TP-based modeling possibility of a nonlinear ICU diabetes model". In: *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE. 2016, pp. 3405–3410.

[60] Laurent El Ghaoui and Silviu-lulian Niculescu. *Advances in linear matrix inequality methods in control*. SIAM, 2000.

[61] Chun-Hsiung Fang, Yung-Sheng Liu, Shih-Wei Kau, Lin Hong, and Ching-Hsiang Lee. "A new LMI-based approach to relaxed quadratic stabilization of T-S fuzzy control systems". In: *IEEE Transactions on Fuzzy Systems* 14.3 (June 2006), pp. 386–397.

[62] Pascal Gahinet, Pierre Apkarian, and Mahmoud Chilali. "Affine parameter-dependent Lyapunov functions and real parametric uncertainty". In: *IEEE Transactions on Automatic control* 41.3 (1996), pp. 436–442.

[63] Péter Galambos and Péter Baranyi. "TP$^\tau$ Model Transformation: A Systematic Modelling Framework to Handle Internal Time Delays in Control Systems". In: *Asian Journal of Control* (2015).

[66] Huijun Gao and Xianwei Li. *Robust Filtering for Uncertain Systems*. Springer, 2016.

[67] *Convex hull manipulation does matter in LMI based observer design, a TP model transformation based optimisation*. Vol. 11. 2010, p. 12.

[68] Yongru Gu, Hua O. Wang, Kazuo Tanaka, and Linda G. Bushnell. "Fuzzy control of nonlinear time-delay systems: stability and design issues". In: *American Control Conference, 2001. Proceedings of the 2001*. Vol. 6. 2001, pp. 4771–4776.

[69] Thierry Marie Guerra, Alexandre Kruszewski, and Jimmy Lauber. "Discrete Tagaki—Sugeno models for control: Where are we?" In: *Annual Reviews in Control* 33.1 (Apr. 2009), pp. 37–47.

[70] Arash Hassibi, Jonathan How, and Stephen Boyd. "A path-following method for solving BMI problems in control". In: *American Control Conference, 1999. Proceedings of the 1999*. Vol. 2. IEEE. 1999, pp. 1385–1389.

[71] Brandon Hencey and Andrew G. Alleyne. "A KYP lemma for LMI regions". In: *IEEE Transactions on Automatic Control* 52.10 (2007), pp. 1926–1930.

[72] Frank L Hitchcock. "The expression of a tensor or a polyadic as a sum of products". In: *Studies in Applied Mathematics* 6.1-4 (1927), pp. 164–189.

[73] Christian Hoffmann and Herbert Werner. "Compact lft-lpv modeling with automated parameterization for efficient lpv controller synthesis". In: *American Control Conference (ACC), 2015*. IEEE. 2015, pp. 119–124.

[74] Alston S Householder. "Unitary triangularization of a nonsymmetric matrix". In: *J. ACM* 5.4 (1958), pp. 339–342.

[75] Mariya Ishteva, Lieven De Lathauwer, P Absil, and Sabine Van Huffel. "Dimensionality reduction for higher-order tensors: algorithms and applications". In: *International Journal of Pure and Applied Mathematics* 42.3 (2008), p. 337.

[76] Tetsuya Iwasaki and Shinji Hara. "Generalized KYP lemma: Unified frequency domain inequalities with design applications". In: *IEEE Transactions on Automatic Control* 50.1 (2005), pp. 41–59.

[77] Abdelhafidh Jaadari, Thierry Marie Guerra, Antonio Sala, and Miguel Bernal. "Finsler's relaxation for local H_∞ controller design of continuous-time Takagi-Sugeno models via non-quadratic Lyapunov functions". In: *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 5648–5653.

[78] Luc Jaulin. *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*. Vol. 1. Springer Science & Business Media, 2001.

[79] Rudolf Emil Kalman et al. "Contributions to the theory of optimal control". In: *Bol. Soc. Mat. Mexicana* 5.2 (1960), pp. 102–119.

[80] Narendra Karmarkar. "A new polynomial-time algorithm for linear programming". In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM. 1984, pp. 302–311.

[81] Narendra Karmarkar. "A new polynomial-time algorithm for linear programming". In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM, 1984, pp. 302–311.

[83] Vladimir Leonidovich Kharitonov. "The asymptotic stability of the equilibrium state of a family of systems of linear differential equations". In: *Differentsial'nye Uravneniya* 14.11 (1978), pp. 2086–2088.

[84] Euntai Kim and Heejin Lee. "New approaches to relaxed quadratic stability condition of fuzzy control systems". In: *Fuzzy Systems, IEEE Transactions on* 8.5 (2000), pp. 523–534.

[85] Sung Hyun Kim. “Relaxation Technique for a T-S Fuzzy Control Design Based on a Continuous-Time Fuzzy Weighting-Dependent Lyapunov Function”. In: *Fuzzy Systems, IEEE Transactions on* 21.4 (2013), pp. 761–766.

[86] József Klespitz, Imre J Rudas, and Levente Kovács. “LMI-based feedback regulator design via TP transformation for fluid volume control in blood purification therapies”. In: *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. IEEE. 2015, pp. 2615–2619.

[87] GT Klincsek. “Minimal triangulations of polygonal domains”. In: *Ann. Discrete Math* 9 (1980), pp. 121–123.

[88] Levente Kovács and György Eigner. “Convex polytopic modeling of diabetes mellitus: A Tensor Product based approach”. In: *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE. 2016, pp. 3393–3398.

[102] Andreas Kwiatkowski, Marie-Theres Boll, and Herbert Werner. “Automated generation and assessment of affine LPV models”. In: *Decision and Control, 2006 45th IEEE Conference on*. IEEE. 2006, pp. 6690–6695.

[103] Dong Hwan Lee, Young Hoon Joo, and Myung Hwan Tak. “Local stability analysis of continuous-time Takagi–Sugeno fuzzy systems: a fuzzy Lyapunov function approach”. In: *Inf. Sci.* 257 (2014), pp. 163–175.

[104] Dong Hwan Lee, Jin Bae Park, and Young Hoon Joo. “A fuzzy Lyapunov function approach to estimating the domain of attraction for continuous-time Takagi–Sugeno fuzzy systems”. In: *Inf. Sci.* 185.1 (2012), pp. 230–248.

[105] DJ Leith and WE Leithead. “On formulating nonlinear dynamics in LPV form”. In: *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*. Vol. 4. 2000, pp. 3526–3527.

[106] Douglas J Leith and William E Leithead. “Survey of gain-scheduling analysis and design”. In: *International journal of control* 73.11 (2000), pp. 1001–1025.

[107] Zsófia Lendek, Thierry-Marie Guerra, and Jimmy Lauber. “Controller design for TS models using delayed nonquadratic Lyapunov functions”. In: *IEEE Trans. Cybern.* 45.3 (2015), pp. 439–450.

[108] Jun Li and José M. Bioucas-Dias. “Minimum volume simplex analysis: a fast algorithm to unmix hyperspectral data”. In: *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*. Vol. 3. IEEE, 2008, pp. III–250.

[109] Xianwei Li and Huijun Gao. “A heuristic approach to static output-feedback controller synthesis with restricted frequency-domain specifications”. In: *IEEE Transactions on Automatic Control* 59.4 (2014), pp. 1008–1014.

[110] Xianwei Li and Huijun Gao. “Robust frequency-domain constrained feedback design via a two-stage heuristic approach”. In: *IEEE transactions on cybernetics* 45.10 (2015), pp. 2065–2075.

[111] Xianwei Li, Huijun Gao, and Changhong Wang. "Generalized Kalman–Yakubovich–Popov lemma for 2-D FM LSS model". In: *IEEE Transactions on Automatic Control* 57.12 (2012), pp. 3090–3103.

[112] Xianwei Li, James Lam, Huijun Gao, and Junlin Xiong. "H_∞ and H_2 filtering for linear systems with uncertain Markov transitions". In: *Automatica* 67 (2016), pp. 252–266.

[113] D. K. Lindner, T. P. Celano, and E. N. Ide. "Vibration Suppression Using a Proofmass Actuator Operating in Stroke/Force Saturation". In: *J. Vib. Acoust.* 113.4 (1991), p. 423.

[114] Peter Liu, Wen-Tsung Yang, and Chang-En Yang. "Output feedback control of fuzzy descriptor systems with interval time-varying delay". In: *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*. 2012, pp. 1–8.

[115] R Liu, R Saeks, and RJ Leake. "On Global Linearization1". In: *Mathematical Aspects of Electrical Network Analysis* 3 (1971), p. 93.

[116] Ruey-Wen Liu. "Convergent systems". In: *Automatic Control, IEEE Transactions on* 13.4 (1968), pp. 384–391.

[117] RW Liu. "Stabilities of nonlinear networks". In: *J. Elec. Eng* (1970), pp. 1–21.

[118] Xiaodong Liu and Qingling Zhang. "New approaches to H_∞ controller designs based on fuzzy observers for T-S fuzzy systems via LMI". In: *Automatica* 39.9 (Sept. 2003), pp. 1571–1582.

[119] AI Lur'e. *Some Nonlinear Problems in the Theory of Automatic Control*. 1951.

[120] Ao Mo Lyapunov. "The general problem of motion stability". In: *Kharkovskoye Matematicheskoe Obshchestvo* 11 (1892).

[121] Andrés Marcos and Gary J Balas. "Development of linear-parameter-varying models for aircraft". In: *Journal of Guidance, Control, and Dynamics* 27.2 (2004), pp. 218–228.

[122] J Clerk Maxwell. "On governors". In: *Proceedings of the Royal Society of London* 16 (1867), pp. 270–283.

[123] W El Messoussi, O Pagès, and A El Hajjaji. "Robust Pole Placement for Fuzzy Models with Parametric Uncertainties: An LMI Approach." In: *EUSFLAT Conf.* 2005, pp. 810–815.

[124] ÁKOS Miklós and Zsolt Szabó. "Simulation and experimental validation of the dynamical model of a dual-rotor vibrotactor". In: *Journal of Sound and Vibration* (2015), pp. 98–107.

[125] Ákos Miklós and Zsolt Szabó. "Vibrator with DC motor driven eccentric rotors". In: *Periodica Polytechnica. Engineering. Mechanical Engineering* 56.1 (2012), pp. 49–53.

[126] Javad Mohammadpour and Carsten W Scherer. *Control of linear parameter varying systems with applications*. Springer Science & Business Media, 2012.

[127] Leonardo A Mozelli, Reinaldo M Palhares, and Gustavo SC Avellar. "A systematic approach to improve multiple Lyapunov function stability and stabilization conditions for fuzzy systems". In: *Inf. Sci.* 179.8 (2009), pp. 1149–1162.

[128] S. Nagy, Z. Petres, and P. Baranyi. "TP Tool-a Matlab toolbox for TP model transformation". In: *Proc. of 8th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*. 2007, pp. 483–495.

[129] Szabolcs Nagy, Zoltán Petres, Peter Baranyi, and Hideki Hashimoto. "Computational relaxed TP model transformation: restricting the computation to subspaces of the dynamic model". In: *Asian Journal of Control* 11.5 (2009), pp. 461–475.

[130] J.M.P. Nascimento and J.M.B. Dias. "Vertex component analysis: a fast algorithm to unmix hyperspectral data". In: *IEEE Transactions on Geoscience and Remote Sensing* 43.4 (Apr. 2005), pp. 898–910.

[131] Yu Nesterov and A Nemirovsky. "A general approach to polynomial-time algorithms design for convex programming". In: *Report, Central Economical and Mathematical Institute, USSR Academy of Sciences, Moscow* (1988).

[132] Yu Nesterov and A Nemirovsky. "A general approach to polynomial-time algorithms design for convex programming". In: *Report, Central Economical and Mathematical Institute, USSR Academy of Sciences, Moscow* (1988).

[133] Yu Nesterov and A Nemirovsky. "Interior-point polynomial methods in convex programming, volume 13 of Studies in Applied Mathematics". In: *SIAM, Philadelphia, PA* (1994).

[134] Harry Nyquist. "Regeneration theory". In: *Bell system technical journal* 11.1 (1932), pp. 126–147.

[135] Jun-Tao Pan, Thierry Marie Guerra, Shu-Min Fei, and Abdelhafidh Jaadari. "Nonquadratic stabilization of continuous T–S fuzzy models: LMI solution for a local approach". In: *Fuzzy Systems, IEEE Transactions on* 20.3 (2012), pp. 594–602.

[136] Junjun Pan and Linzhang Lu. "TP Model Transformation Via Sequentially Truncated Higher-Order Singular Value Decomposition". In: *Asian J. Control* 17.2 (Mar. 2015), pp. 467–475.

[137] Patrick C. Parks. "AM Lyapunov's stability theory—100 years on". In: *IMA journal of Mathematical Control and Information* 9.4 (1992), pp. 275–303.

[138] MICHAEL E. POLITES. "New method for scanning spacecraft and balloon-borne/space-based experiments". In: *Journal of Guidance, Control, and Dynamics* 14.3 (May 1991), pp. 548–553.

[139] ES Pyatnitskii and VI Skorodinskii. "Numerical method of construction of Lyapunov functions and absolute stability criteria in the form of numerical procedures." In: *AUTOM. REMOTE CONTROL.* 44.11 (1984), pp. 1427–1437.

[140] Ye S Pyatnitskiy and VI Skorodinskiy. “Numerical methods of Lyapunov function construction and their application to the absolute stability problem”. In: *Systems & Control Letters* 2.2 (1982), pp. 130–135.

[141] Jianbin Qiu, Gang Feng, and Huijun Gao. “Static-Output-Feedback H_∞ Control of Continuous-Time T-S Fuzzy Affine Systems Via Piecewise Lyapunov Functions”. In: *IEEE Transactions on Fuzzy Systems* 21.2 (2013), pp. 245–261.

[142] R. H. Rand, R. J. Kinsey, and D. L. Mingori. “Dynamics of spinup through resonance”. In: *International Journal of Non-Linear Mechanics* 27.3 (May 1992), pp. 489–502.

[143] Bong-Jae Rhee and Sangchul Won. “A new fuzzy Lyapunov function approach for a TS fuzzy control system design”. In: *Fuzzy sets and systems* 157.9 (2006), pp. 1211–1228.

[144] EJ Routh. *Essay on the stability of Motion*. 1876.

[145] Michael George Safonov. *Stability and robustness of multivariable feedback systems*. MIT press, 1980.

[146] Michael Safonov and Richard Chiang. “Robust control toolbox”. In: *The Mathworks Inc* (1988).

[147] A. Sala and C. Arino. “Relaxed Stability and Performance LMI Conditions for Takagi–Sugeno Fuzzy Systems With Polynomial Constraints on Membership Function Shapes”. In: *IEEE Transactions on Fuzzy Systems* 16.5 (Oct. 2008), pp. 1328–1336.

[148] Antonio Sala. “On the conservativeness of fuzzy and fuzzy-polynomial control of nonlinear systems”. In: *Annual Reviews in Control* 33.1 (2009), pp. 48–58.

[149] Antonio Sala and Carlos Arino. “Relaxed stability and performance conditions for Takagi–Sugeno fuzzy systems with knowledge on membership function overlap”. In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 37.3 (2007), pp. 727–732.

[150] Antonio Sala and Carlos Ariño. “Asymptotically necessary and sufficient conditions for stability and performance in fuzzy control: Applications of Pólya’s theorem”. In: *Fuzzy Sets and Systems* 158.24 (Dec. 2007), pp. 2671–2686.

[151] Antonio Sala and Thierry M. Guerra. “Stability analysis of fuzzy systems: membership-shape and polynomial approaches”. In: *Proc. IFAC World Congress*. 2008, pp. 5605–5610.

[152] Antonio Sala, Thierry Marie Guerra, and Robert Babuška. “Perspectives of fuzzy systems and control”. In: *Fuzzy Sets and Systems* 156.3 (Dec. 2005), pp. 432–444.

[153] Carsten W Scherer. “LPV control and full block multipliers”. In: *Automatica* 37.3 (2001), pp. 361–375.

[154] Bernard N Sheehan and Yousef Saad. “Higher Order Orthogonal Iteration of Tensors (HOOI) and its Relation to PCA and GLRAM.” In: *SDM*. SIAM. 2007, pp. 355–365.

[155] Victor Costa da Silva Campos, Leonardo Antonio Borges Torres, and Reinaldo Martinez Palhares. “Revisiting the TP Model Transformation: Interpolation and Rule Reduction”. In: *Asian J. Control* 17.2 (2015), pp. 392–401.

[156] P Stein. “A note on the volume of a simplex”. In: *American Mathematical Monthly* (1966), pp. 299–301.

[157] Gilbert W Stewart. “An updating algorithm for subspace tracking”. In: *IEEE Transactions on Signal Processing* 40.6 (1992), pp. 1535–1541.

[158] JF Sturm. “Using SeDuMi 1. 02, a MATLAB toolbox for optimization over symmetric cones. 1999”. In: *MR1778433* (2004).

[159] Michio Sugeno and Tomohiro Takagi. “Multi-dimensional fuzzy reasoning”. In: *Fuzzy sets and systems* 9.1-3 (1983), pp. 313–325.

[160] László Szeidl, Peter Baranyi, Zoltán Petres, and Peter Varlaki. “Numerical reconstruction of the HOSVD based canonical form of polytopic dynamic models”. In: *Computational Intelligence and Intelligent Informatics, 2007. ISCIII'07. International Symposium on*. IEEE. 2007, pp. 111–116.

[161] Alexandra Szollosi and Peter Baranyi. “Improved control performance of the 3-DoF aeroelastic wing section: a TP model based 2D parametric control performance optimization”. In: *Asian Journal of Control* 19.2 (2017), pp. 450–466.

[162] Alexandra Szollosi, Péter Baranyi, and Péter Várlaki. “Example for Convex Hull Tightening increasing the feasible parameter region at Linear Matrix Inequality based Control Design”. In: *Intelligent Engineering Systems (INES), 2014 IEEE 18th International Conference on*. 2014.

[163] Árpád Takács, Tamás Haidegger, Péter Galambos, József Kuti, and Imre J Rudas. “Nonlinear soft tissue mechanics based on polytopic Tensor Product modeling”. In: *Applied Machine Intelligence and Informatics (SAMI), 2016 IEEE 14th International Symposium on*. IEEE. 2016, pp. 211–215.

[166] Béla Takarics. “TP model transformation based sliding mode control and friction compensation”. 2011.

[167] Béla Takarics and Yeung Yam. “Robust Grid Point-Based Control Design for LPV Systems via Unified TP Transformation”. In: *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. IEEE. 2015, pp. 2626–2631.

[168] K. Tanaka, H. Ohtake, and H.O. Wang. “A Practical Design Approach to Stabilization of a 3-DOF RC Helicopter”. In: *IEEE Transactions on Control Systems Technology* 12.2 (Mar. 2004), pp. 315–325.

[169] Kazuo Tanaka, Takayuki Ikeda, and Hua O. Wang. “Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs”. In: *Fuzzy Systems, IEEE Transactions on* 6.2 (1998), pp. 250–265.

[170] Kazuo Tanaka, Hiroshi Ohtake, and Hua O. Wang. “Shared nonlinear control in internet-based remote stabilization”. In: *Fuzzy Systems, 2005. FUZZ’05. The 14th IEEE International Conference on*. 2005, pp. 714–719.

[171] Kazuo Tanaka and Hua O Wang. *Fuzzy control systems design and analysis a linear matrix inequality approach*. English. New York: Wiley, 2001.

[172] Xuetao Tao, Bin Wang, Liming Zhang, and Jian Qiu Zhang. “A new scheme for decomposition of mixed pixels based on nonnegative matrix factorization”. In: *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*. July 2007, pp. 1759–1762.

[173] Jean-baptiste Thevenet, Dominikus Noll, and Pierre Apkarian. “Nonlinear spectral SDP method for BMI-constrained problems: Applications to control design”. In: *in Proceedings ICINCO*. 2004.

[174] Roland Tóth. *Modeling and identification of linear parameter-varying systems*. Vol. 403. Springer, 2010.

[175] Alex Townsend and Lloyd N Trefethen. “An extension of Chebfun to two dimensions”. In: *SIAM J. Sci. Comput.* 35.6 (2013), pp. C495–C518.

[176] Alex Townsend and Lloyd N Trefethen. “Continuous analogues of matrix factorizations”. In: *Proc. R. Soc. A*. Vol. 471. 2173. The Royal Society. 2015.

[177] L. N. Trefethen. “Householder triangularization of a quasimatrix”. en. In: *IMA J. Num. Anal.* 30.4 (Oct. 2010), pp. 887–897.

[178] Hoang Duong Tuan, Pierre Apkarian, Tatsuo Narikiyo, and Yasuhiro Yamamoto. “Parameterized linear matrix inequality techniques in fuzzy control system design”. In: *Fuzzy Systems, IEEE Transactions on* 9.2 (2001), pp. 324–332.

[179] Reha H Tütüncü, Kim C Toh, and Michael J Todd. “Solving semidefinite-quadratic-linear programs using SDPT3”. In: *Mathematical programming* 95.2 (2003), pp. 189–217.

[180] Péter Várkonyi, Domonkos Tikk, Péter Korondi, and Péter Baranyi. “A new algorithm for RNO-INO type tensor product model representation”. In: *2005 IEEE International Conference on Intelligent Engineering Systems, INES*. Vol. 5. 2005, pp. 263–266.

[181] Chih-Jian Wan, D. S. Bernstein, and V. T. Coppola. “Global stabilization of the oscillating eccentric rotor”. In: *Proceedings of 1994 33rd IEEE Conference on Decision and Control*. Institute of Electrical & Electronics Engineers (IEEE), 1994.

[182] H.O. Wang, K. Tanaka, and M. Griffin. “Parallel distributed compensation of nonlinear systems by Takagi-Sugeno fuzzy model”. In: *Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE Int.* Vol. 2. Mar. 1995, pp. 531–538.

[183] H.O. Wang, K. Tanaka, and M. Griffin. "Parallel distributed compensation of nonlinear systems by Takagi-Sugeno fuzzy model". In: vol. 2. Mar. 1995, 531–538 vol.2.

[184] Jan C Willems. "Least squares stationary optimal control and the algebraic Riccati equation". In: *Automatic Control, IEEE Transactions on* 16.6 (1971), pp. 621–634.

[185] Shengyuan Xu and J. Lam. "Robust H_∞ control for uncertain discrete-time-delay fuzzy systems via output feedback controllers". In: *IEEE Transactions on Fuzzy Systems* 13.1 (Feb. 2005), pp. 82–93.

[186] Vladimir A Yakubovich. "The method of matrix inequalities in the stability theory of nonlinear control systems. I-II-III". In: *Automation and Remote control* 25-26 (1967), pp. 577–592, 753–763, 905–917.

[187] Vladimir A Yakubovich. "The solution of certain matrix inequalities in automatic control theory". In: *Soviet Math. Dokl.* Vol. 3. 1962, pp. 620–623.

[188] George Zames and Bruce Francis. "Feedback, minimax sensitivity, and optimal robustness". In: *IEEE Transactions on Automatic Control* 28.5 (1983), pp. 585–601.

[189] Qian Zheng. *Sum of squares based nonlinear control design techniques.* North Carolina State University, 2009.

[190] Kemin Zhou and John Comstock Doyle. *Essentials of robust control.* Vol. 104. Prentice hall Upper Saddle River, NJ, 1998.

[191] A. Zymnis, S.J. Kim, J. Skaf, M. Parente, and S. Boyd. "Hyperspectral Image Unmixing via Alternating Projected Subgradients". In: *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on.* Nov. 2007, pp. 1164–1168.