Lehrstuhl für Regelungstechnik

Technische Universität München

TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Regelungstechnik

Set Point and Trajectory Tracking of Constrained Systems in Takagi-Sugeno Form

Klaus J. Diepold

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Manfred Hajek
Prüfer der Dissertation:	Univ.-Prof. Dr.-Ing. habil. Boris Lohmann
	Univ.-Prof. Dr.-Ing. Andreas Kroll

Die Dissertation wurde am 11.11.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 11.04.2016 angenommen.

For my family.

Acknowledgments

At this point, I would like to express my sincere thanks to my supervisor Prof. Boris Lohmann for being part of his open-minded research group and to work in an interdisciplinary project. The enthusiasm of Prof. Lohmann concerning new research topics, his encouragement as well as his confidence in me have opened the door to this thesis. In this context, I gratefully acknowledge the German Research Foundation (DFG) for funding this thesis as part of the collaborative research "Managing cycles in innovation processes - Integrated development of product service systems based on technical products" (SFB 768).

I also would like to thank Prof. Andreas Kroll for his interest in my work – for the helpful hints he gave on the topic and for being my second examiner. Many thank to Prof. Manfred Hajek for chairing the board of examiners.

I am also thankful for being blessed to work together with such amazing students and colleagues at the Institute of Automatic Control. I very appreciate the helpfulness, encouragement and friendliness of all of you. Many of you became very good friends – thanks to all of you and a special thanks to André Albers, Hendrik Börner, Dr. Michael Buhl, Ronnie Dessort, Tobias Guggemos, Thomas Huber and Benjamin Stahl. I deeply thank my colleagues and friends Klaus Albert, Sergio Delgado, Dr. Tobias Kloiber, Dr. Enrico Pellegrini, Dr. Heiko Peuscher, Dr. Peter Philipp, Sebastian Pieczona and Nils Pletschen. The fruitful discussions on and off the job as well as the synergistic and cooperative way of working together valuable contribute to this work.

Last but not least, let me express a very special and heartfelt thanks to my parents Irmtraud and Dieter Diepold – for their sustained love, support and encouragement throughout my life. I owe unspeakable much to you.

Contents

Glossary

Frequently used Acronyms

CAD	computer-aided design
DA	domain of attraction
DOF	degree of freedom
GINA	Governor Integrated Nominal-Value Adaptation
LISA	Limits of Inputs and States are Allocated
LMI	linear matrix inequality
LO	local
MPC	model predictive control
PCU	position control unit
PDC	parallel distributed compensation
RFS	recurrent fuzzy system
RG	reference governor
SE-NL	sector nonlinearity
SOS	sum of squares
T-S	Takagi-Sugeno
TU	transformation unit
UoD	universe of discourse
VCU	velocity control unit
VTOL	vertical take-off and landing
YACU	yaw angle control unit

Frequently used Indices

$(.)^*$	upper index $*$ denotes that (.) is an equilibrium (set point)
$(.)^*_d$	indices $*$ and d denote that (.) is a desired equilibrium (set point)
$(.)^*_t$	indices $*$ and t denote that (.) is a temporary equilibrium (set point)
$(.)^*_r$	indices $*$ and r denote that (.) is a reference equilibrium (set point)

$(.)_e$	lower index e denotes that (.) belongs to the tracking error dynamics
$(.)_i$	lower index i denotes that (.) is the i-th element from a set
$(.)_j$	lower index j denotes that (.) is the j-th element from a set
$(.)_k$	lower index k denotes that (.) is the k-th element from a set
$(.)_{max}$	lower index max denotes the maximum absolute value of (.)
$(.)_T$	lower index T denotes that (.) belongs to a trajectory
(k)	iteration or discretization index
${}_{\gamma}(.)$	pre-index γ denotes that (.) belongs to a yaw angle control unit
${}_{p}(.)$	pre-index p denotes that (.) belongs to a position control unit
${}_{v}(.)$	pre-index v denotes that (.) belongs to a velocity control unit

Frequently used Latin Symbols

$\mathbf{A}$	dynamic matrix of a linear system
$\mathbf{A}(\mathbf{x},\mathbf{u})$	state affine dependent dynamic matrix of a nonlinear system
$\hat{\mathbf{A}}$	dynamic matrix of a linear system subject to actuator dynamics
$\mathbf{B}$	matrix of the inputs of a linear system
$\mathbf{B}(\mathbf{x},\mathbf{u})$	input affine dependent matrix of the inputs of a nonlinear system
$\hat{\mathbf{B}}$	matrix of the inputs of a linear system subject to actuator dynamics
c	scaling parameter
$c(k)$	discrete value of a scaling parameter c at an iteration (k)
$\hat{\mathbf{c}}(\mathbf{x},\mathbf{u}) < \mathbf{0}$	input and state depending constraints
$\mathbf{D}_{lin}$	linearized damping matrix
$\mathbf{E}$	boolean diagonal matrix
$\mathbf{E}^{-}$	denotes a matrix defined by $\mathbf{I} - \mathbf{E}$
E	edges of a automaton
$\mathbf{e}$	tracking error
$\mathbf{f}(\mathbf{x},\mathbf{u})$	nonlinear dynamical system
$\mathbf{f}(\mathbf{x})\mathbf{g}(\mathbf{x})\mathbf{u}$	input affine dynamical systems
$\hat{\mathbf{f}}(\mathbf{x})$	smooth piecewise analytical function of allowed equilibria
$\hat{\mathbf{f}}(\mathbf{x}(k))$	numerically calculated steady-state curve of discrete equilibria
$\mathbf{F}$	linear state feedback matrix
$\tilde{\mathbf{F}}$	linear state feedback matrix
$\mathbf{g}_i$	vector with zero elements, expert the i-th element which is one
h_i	scalar nonlinear blending function i
$\mathbf{H}$	linear state feedback matrix, auxiliary controller to $\mathbf{F}$

$\mathbf{I}$	identity matrix
$\mathbf{K}_{lin}$	linearized stiffness matrix
$\mathbf{M}_{lin}$	linearized mass matrix
$\mathbf{P}$	positive definite Lyapunov matrix
$\bar{\mathbf{p}}^{x}_{\mathbf{C}_w}$	optimization vector for recurrent fuzzy switching
$\mathbf{q}$	minimal coordinate vector
$\mathbf{Q}$	scaled inverse of $\mathbf{P}$
$\mathbf{Q}_s$	Kalman controllability matrix
$\mathbf{Q}_{LQR}$	weighting matrix of the state vector when designing a LQR
$\mathbf{R}_{LQR}$	weighting matrix of the input vector when designing a LQR
$\mathbf{s}$	core position vector
t	time variable
t_0	initial time of a trajectory
t_e	final time of a trajectory
$\mathbf{T}$	diagonal matrix of inverse actuator time constants
$\mathbf{u}$	input vector
$\tilde{\mathbf{u}}$	virtual (auxiliary) input vector
$\mathbf{u}_0$	initial input vector
$\dot{\mathbf{u}}$	time derivative of the input vector
$\mathbf{u}^*$	stationary control input associated with $\mathbf{x}^*$
$\mathbf{u}^*_r$	stationary control input associated with $\mathbf{x}^*_r$
$\mathbf{u}^*_t$	stationary control input associated with $\mathbf{x}^*_t$
$\mathbf{u}^*_d$	stationary control input associated with $\mathbf{x}^*_d$
$\mathbf{u}_T$	desired input trajectory associated with $\mathbf{x}^*_T$
V	control modes of a automaton
$V_{\mathbf{x}^*}$	quadratic Lyapunov function at $\mathbf{x}^*$
$V_{\mathbf{e}_t,\mathbf{e}_d}$	Lyapunov-like function for proofing stability of the GINA controller
$\dot{V}_{\mathbf{x}^*}$	time derivative of a quadratic Lyapunov function at $\mathbf{x}^*$
$\mathbf{v}$	input vector of an actuator
$\mathbf{x}$	state vector
$\bar{x}$	state variable for recurrent fuzzy switching
$\mathbf{x}_0$	initial state vector
$\mathbf{x}^*$	equilibrium point (set point)
$\mathbf{x}^*_r$	reference equilibrium point (set point)
$\mathbf{x}^*_t$	temporary equilibrium point (set point)
$\mathbf{x}^*_d$	desired equilibrium point (set point)

$\mathbf{x}_T$	desired state trajectory
$\mathbf{x}^*(k)$	discrete value of an equilibrium point (set point) at iteration (k)
$\mathbf{x}_r^*(k)$	value of a reference equilibrium point (set point) at iteration (k)
$\mathbf{x}_t^*(k)$	value of a temporary equilibrium point (set point) at iteration (k)
$\mathbf{x}_t^{*-}(k)$	value of an equilibrium point that is an infinitesimal step closer towards $\mathbf{x}_r^*(k)$ than $\mathbf{x}^*(k)$
$\mathbf{x}_t^{*+}(k)$	value of an equilibrium point that is an infinitesimal step closer towards $\mathbf{x}_d^*$ than $\mathbf{x}^*(k)$
$\dot{\mathbf{x}}$	time derivative of the state vector
$\hat{\mathbf{x}}$	state vector of a of a system subject to actuator dynamics
$\dot{\hat{\mathbf{x}}}$	time derivative of a state vector of a system subject to actuator dynamics
$\mathbf{x}_{gen}$	generalized state vector
$\mathbf{y}$	output of a dynamical system
$\mathbf{y}_f$	flat output of a dynamical system
$\mathbf{z}$	vector of independent directions within a subspace
$\mathbf{z}_f$	flat coordinates of a system
$\mathbf{z}_s$	premise vector of the system
$\mathbf{z}_c$	premise vector of the controller
$\mathbf{z}_w$	premise vector with variables that causes a system to switch

Frequently used Greek Symbols

α	decay rate of a quadratic Lyapunov function
$\alpha_{\mathbf{u}}$	required decay rate for tracking subject to input amplitude restrictions
$\alpha_{\dot{\mathbf{u}}}$	required decay rate for tracking subject to input rate restrictions
$\alpha_{\mathbf{x}}$	required decay rate for tracking subject to state restrictions
α_L	required decay rate for tracking subject to input amplitude, rate and state restrictions
$\boldsymbol{\Delta}$	closed-loop T-S controller including an over-saturating region
$\Delta(k)$	parameter to adapt the scaling variable $c(k)$
ϵ	region in the state space
$\tilde{\epsilon}$	region in the state space which is smaller than ϵ
Φ	defines a matrix tuple
$\boldsymbol{\Gamma}$	vector summarizing the position coordinates of the state vector

$\dot{\mathbf{\Gamma}}$	vector summarizing the velocity coordinates corresponding to $\mathbf{\Gamma}$
$\tilde{\mathbf{\Gamma}}$	vector summarizing the remaining state variables of a system (not part of $\mathbf{\Gamma}$, $\dot{\mathbf{\Gamma}}$)
$\eta_{\mathbf{u}^*}$	bounding level value of an ellipsoidal domain of attraction at $\mathbf{x}^*$
$\eta_{\mathbf{u}_r^*}$	bounding level value of an ellipsoidal domain of attraction at $\mathbf{x}_r^*$
$\eta_{\mathbf{u}_t^*}$	bounding level value of an ellipsoidal domain of attraction at $\mathbf{x}_t^*$
$\eta_{\mathbf{u}_d^*}$	bounding level value of an ellipsoidal domain of attraction at $\mathbf{x}_d^*$
$\eta_{\mathbf{u}_e^*}$	bounding level value of an ellipsoidal domain of attraction at $\mathbf{e}^*$
$\eta_{\mathbf{0}}(t)$	general time-variant bounding level value of an ellipsoidal domain of attraction at $\mathbf{e}^* = \mathbf{0}$
$\eta_{\mathbf{0}}(\mathbf{u}_T(t))$	time-variant bounding level value (depending on $\mathbf{u}_T$) of an ellipsoidal domain of attraction at $\mathbf{e}^* = \mathbf{0}$
$\eta_{\mathbf{0}}(\dot{\mathbf{u}}_T(t))$	time-variant bounding level value (depending on $\dot{\mathbf{u}}_T$) of an ellipsoidal domain of attraction at $\mathbf{e}^* = \mathbf{0}$
$\eta_{\mathbf{0}}(\mathbf{x}_T(t))$	time-variant bounding level value (depending on $\mathbf{x}_T$) of an ellipsoidal domain of attraction at $\mathbf{e}^* = \mathbf{0}$
$\tilde{\eta}_{\mathbf{u}^*}$	bounding level value with $\tilde{\eta}_{\mathbf{u}^*} \geq \eta_{\mathbf{u}^*}$
τ	actuator time constant
ϖ_b	distance where the braking process starts
ϖ_s	value of the Lyapunov function under which the PCU is activated
$\theta_k(\mathbf{z}_s)$	nonlinear function k depending on the premises vector of a system
$\sum_j$	sum over all elements j
μ	membership function
$\nu_t^-(k)$	ration of the Lyapunov function value and the corresponding bounding level value at $\mathbf{x}_t^{*-}(k)$
$\nu_t^+(k)$	ration of the Lyapunov function value and the corresponding bounding level value at $\mathbf{x}_t^{*+}(k)$
$\nu_t^\pm(k)$	ration of $\nu_t^-(k)$ and $\nu_t^+(k)$
$\nu_t(k)$	ration of the Lyapunov function value and the corresponding bounding level value at $\mathbf{x}_t^*(k)$
ϖ_{max}	constraint of the absolute value of a single state or input variable
$\varpi_{T,max}$	constraint $\varpi_{T,max} < \varpi_{max}$ allowed for a desired trajectory
$\varpi_{e,max}$	$\varpi_{e,max} = \varpi_{max} - \varpi_{T,max}$ allowed for tracking error compensation
$\boldsymbol{\sigma}$	saturation function
Ξ	sampling period
ξ	general switching parameter

$\boldsymbol{\psi}$	vector for mapping the state vector into the flat coordinates
$\boldsymbol{\Psi}$	transformation into flat coordinates
ζ	variable that represents a bound

Frequently used Notation

$\dot{(.)}$	time derivative of (.)
$\frac{\partial(.)}{\partial(..)}$	partial derivative of (.) in (..)
$(.)^k$	k-th time derivative
$(.)^-$, $(.)^+$	lower and upper bound of a variable (.)
$\mathrm{co}\{.,..\}$	convex hull spanned by the elements $\{.,..\}$
$\log(.)$	natural logarithm function of (.)
$\mathrm{sgn}(.)$	signum function of (.)
$\mathcal{C}^k$	at least k-times continuously differentiable function
$\mathcal{C}$	set of scaling variables c
$\delta(.)$	incremental piece of a variable (.)
$\mathcal{E}_{\mathbf{e}^*}(\mathbf{P}, \eta_{\mathbf{u}_e^*})$	ellipsoidal domain of attraction at $\mathbf{e}^*$ with the bounding level value $\eta_{\mathbf{u}_e^*}$
$\partial\mathcal{E}_{\mathbf{e}^*}(\mathbf{P}, \eta_{\mathbf{u}_e^*})$	bounding level set of an ellipsoidal domain of attraction $\mathcal{E}_{\mathbf{e}^*}(\mathbf{P}, \eta_{\mathbf{u}_e^*})$
$\mathcal{E}_{\mathbf{e}^*}(\mathbf{P}, \eta_{\mathbf{0}}(t))$	general time-variant ellipsoidal domain of attraction at $\mathbf{e}^* = \mathbf{0}$ with the bounding level value $\eta_{\mathbf{0}}(t)$
$\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\mathbf{u}_T(t)))$	time-variant ellipsoidal domain of attraction at $\mathbf{e}^* = \mathbf{0}$ where the time dependence is based on $\mathbf{u}_T$
$\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\dot{\mathbf{u}}_T(t)))$	time-variant ellipsoidal domain of attraction at $\mathbf{e}^* = \mathbf{0}$ where the time dependence is based on $\dot{\mathbf{u}}_T$
$\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\mathbf{x}_T(t)))$	time-variant ellipsoidal domain of attraction at $\mathbf{e}^* = \mathbf{0}$ where the time dependence is based on $\mathbf{x}_T$
$\mathcal{L}(\mathbf{F})$	polytope spanned based on a matrix $\mathbf{F}$ and the state vector $\mathbf{x}$
$\mathcal{M}_{s,x}$	set of state variables that are part of the premises vector
$\mathcal{M}_{s,u}$	set of input variables that are part of the premises vector
$\mathcal{M}_r^*$	set of reference equilibria
$\mathbb{N}$	set of natural numbers
$\mathbb{N}_{i:j}$	set of natural numbers from i to j with $j > i$
$\mathbb{R}$	set of real numbers
$\mathcal{S}_{\mathbf{x}}$	defines a symmetric polytope based on the state vector

$\mathcal{S}_{\mathbf{u}}$	defines a symmetric polytope based on the input vector
$\mathcal{S}_{\mathbf{x}_l^*}$	set which is spanned by a set point $\mathbf{x}_l^*$ and its surrounding set points along a discrete equilibria curve
$\mathcal{T}$	transition time of a trajectory
$\mathcal{V}$	set of boolean diagonal matrices
$\mathcal{X}_{\mathbf{x}^*}(\mathbf{P}, \eta_{\mathbf{u}^*})$	ellipsoidal domain of attraction at $\mathbf{x}^*$ with the bounding level value $\eta_{\mathbf{u}^*}$
$\partial\mathcal{X}_{\mathbf{x}^*}(\mathbf{P}, \eta_{\mathbf{u}^*})$	bounding level set of an ellipsoidal domain of attraction $\mathcal{X}_{\mathbf{x}^*}(\mathbf{P}, \eta_{\mathbf{u}^*})$
$\mathbf{A}_{(.)(..)}$	transformation matrix from the coordinate system (.) into (..)
$\mathbf{F}^T$	transpose of a matrix $\mathbf{F}$
$\mathbf{F}^{-1}$	inverse (or pseudoinverse) of a matrix $\mathbf{F}$
$\tilde{\mathbf{F}}$	defines the matrix $\mathbf{FQ}$ where $\mathbf{Q}$ is the inverse Lyapunov matrix
$\mathbf{F} > \mathbf{0}$	positive definite matrix $\mathbf{F}$
$\mathbf{F} \geq \mathbf{0}$	positive semi-definite matrix $\mathbf{F}$
$\mathbf{F} < \mathbf{0}$	negative definite matrix $\mathbf{F}$
$\mathbf{F} \leq \mathbf{0}$	negative semi-definite matrix $\mathbf{F}$
$\mathbf{f}^T$	transpose of a vector $\mathbf{f}$
f_{max}	maximal absolute value of a variable f
$\|f\|$	absolute value of a variable f
$\mathbf{L}_{(.)}^{(..)}$	vector of linguistic values (.) associated with (..)
$\mathbf{L}_{(.),(..)}^{\mathbf{x}}$	vector of linguistic values (.) associated with the state vector $\mathbf{x}$ after mapping (..)
$l+1 \rightarrow l$	switching a signal from $l+1$ towards l
l^+	value of the actual switching signal right before switching
l^-	value of the actual switching signal right after switching
s_t	vertex of a parabola function which depends on the scaling variable c
$\tilde{s}_t$	smallest vertex of parabola functions which depend on the scaling variable c
$\det(\mathbf{F})$	determinant of a matrix $\mathbf{F}$
$\operatorname{rank}(\mathbf{F})$	Rank of the matrix $\mathbf{F}$
$\operatorname{trace}(\mathbf{F})$	trace of a matrix $\mathbf{F}$
$\operatorname{eig}(\mathbf{F})$	eigenvalues of a matrix $\mathbf{F}$
$\|\mathbf{Fx}\|$	vector of absolute values of the vector $\mathbf{Fx}$

Chapter 1

Introduction

Most of the present control tasks are either *set point* or *trajectory tracking* problems. In the first category, a fixed or changing set point should be tracked, e.g. cruise control allows to keep a car at a constant speed which can be manually changed. The second, more challenging, category deals with problems where a system should follow a defined path through space as a function of time. For instance, mobile robots, robotic manipulators or flight systems like quadrotors that move along such a path through the terrain. Due to the rising complexity of mechatronics systems an efficient and performant tracking controller design requires often sophisticated methods which aggravates their practical applicability. In this context, attributes like numerically efficient and systematic design are of gaining importance. In addition, practical relevant system constraints, e.g. actuator saturation, presents a major challenge. Disregarding constraints might diminish the tracking performance, cause an unstable behavior or even damage the system.

In this thesis, we will develop methods for designing fast and asymptotically stable set point as well as trajectory tracking controllers. To put the material in this thesis in perspective, we start this introduction by reviewing current tracking control strategies and their drawbacks in Section 1.1. Based on that, we clarify our focus and point out the main contributions of this thesis in Section 1.2. Finally, we summarize the layout of the thesis Section 1.2.

1.1 Tracking Control Subject to Constraints

Control engineers are aware of the necessity to account system constraints while tracking a set point or a trajectory. Input saturation is by far one of the most important limitations. The magnitude and often also the rate that an actuator can deliver is practically bounded. For instance, saturation effects have been crucially involved in aircraft crashes and the nuclear disaster happened in 1986 at the Chernobyl nuclear plant [108]. A second practically relevant category to be aware of for performance and safety reasons are state constraints of a system. As reported by the NASA [74], an aircraft needs to be operated within a specified *flight envelope* which is usually given by a convex polyhedral set in the state space. The set is bounded by input and state constraints, e.g. pitch, roll and speed limitations. Departing from the flight envelope results often in a *loss of control* which is one of the main reasons for fatal aircraft crashes. Another example arising from the field of mobile robotics is the problem of motion planning. Here the allowed motion is often constrained due to obstacles and the robot's mobility (differential constraints) [96]. During the last decades five effective directions for facing constrained tracking control problems have crystallized out:

One route is to synthesize a *saturated controller* which is aware of the constraints. As implied by the name, these controllers have been originally established for tackling the input saturation problem. Concerning linear systems, the synthesis process is generally accepted to be formulated as a convex numerical optimization problem based on *linear matrix inequalities* (LMIs) [18, 118]. The major advance of their convexity property is that a local minimum is simultaneously a global one in the defined search space and hence an efficient optimization is expeditious. All requirements and system constraints that can be represented in terms of LMIs can be considered. Lots of effort has been made for set point tracking tasks by optimally estimating a domain of attraction (DA) of the desired equilibrium to be tracked subject to constraints. Thereby, the controller can be either predefined or simultaneously designed subject to desired performance requirements, e.g. defined pole region of the closed-loop system [57, 114]. The design of *nested DAs* by parametrized LMIs allows to derive a large stability region and a desired rate of convergence by switching between the invariant sets in their nested order [114]. Recently, the LMI-based design has been utilized for synthesizing a saturated controller for compensating a *trajectory tracking* error for linear systems subject to input limitations [62]. The problem is that both together, feedforward and feedback part, has to stay within the system limits. This is bypassed by the authors by subdividing the overall input signal into two parts: one reserved for the desired trajectory (feedforward) and one for error compensation (feedback). The static allocation of the input limits lead to a linear error dynamics with fixed saturation wherefore the LMI-framework is directly applicable. However, the control performance may be leaking as the input is only exploited when both, feedforward and feedback part, are saturating. For instance, if only a fraction of the reserved input signal is needed in some parts of the trajectory then the allocated saturation limit for error compensation is conservative.

Based on that, the following two questions arise naturally: First, can the input signal be adjusted for trajectory tracking, meaning to vary the saturation limits of the feedback part to the remaining values which are currently not required by the feedforward controller? Second, can the LMI-framework be general exploited also for nonlinear systems? Up the authors' knowledge, the first question remains largely unanswered. Only the variation of rate saturation has been investigated in [61] for linear systems. Contrary to that, progress has been made during the last decade concerning the second question at least for set point tracking. For instance, the method of sum of squares (SOS) has been introduced for polynomial systems [102] and in [49] exact feedback linearization has been considered to reformulate a nonlinear system to a linear one with a state-dependent input saturation. Also nested invariant sets has been estimated for nonlinear systems based on Fourier series [93]. However, the estimation of a DA for nonlinear systems is generally not an easy task and requires an iterative and numerically demanding computation.

During the last decade the *Takagi-Sugeno* (T-S) framework received great attention. It allows to represent a quite general class of nonlinear systems as a weighted sum of linear or affine systems which opens the door to apply convex optimization-based controller design [39, 113]. Hence, a common belief is that T-S model-based techniques offer a simple and effective way for controlling nonlinear systems. The estimation of a DA of a desired set point as well as a saturated controller design is intuitively possible for constrained T-S systems by finding a quadratic Lyapunov function [23, 67, 129]. During the last years piecewise, polynomial or fuzzy Lyapunov function approaches have been focused to reduce the inherent conservatism of the estimated DA based on a quadratic Lyapunov function, e.g. see [77, 95, 120]. However, most of this strategies seem to be of academic nature due to the following facts: First, they are not applicable to T-S system with affine terms [37]. Second, the applicability is limited to low-order systems due to the numerical complexity of the growing number of LMIs [104]. Third, the consideration of constraints notably increases the required computational effort or even leads to numerical problems. Consequently, quadratic Lyapunov function approaches are still of high practical relevance. A natural way of increasing the size of an ellipsoidal DA is by estimating several ones which pushes the topic of switched T-S control [121, 122, 91]. However, switched T-S controllers often require a high number of LMIs to be solved or result in complex switching conditions. For instance, relaxing the complexity by designing nested DAs (similar to linear systems) has not been investigated up the author's knowledge. Hence, we can state that although lots of progress has been made concerning the T-S framework, there are still some open research questions to be tackled - even for quadratic Lyapunov functions: First, an effective and practically applicable relaxation of the conservativeness of the estimate. Second, while input amplitude saturation has been often addressed within the T-S framework [23, 36, 129], input rate and state limits have not, up the authors' knowledge, been investigated in-depth. Third, the T-S framework is not exploited for trajectory tracking tasks. Most of the related literature assumes a linear reference dynamic [14, 15, 45].

A second route for tackling control issues subject to constraints is to iteratively solve a constrained open-loop optimal control problem. The optimization task is solved by a model-based prediction of the system behavior over a defined horizon which yield the method the name *Model Predictive Control* (MPC), see [11, 48, 75] for a broad method overview. Generally, the obtained solution will be executed until the next measurements become available for solving the optimization problem again. Due to its unified characteristics, the MPC framework is an universal approach for set point and trajectory tracking. However, practically the method possesses two contrary problems: *computational cost* and *stability.* The input function is usually discretized along the prediction horizon for enabling a real-time numerical computation of the optimization problem. Hence, it becomes obvious that a short horizon is desirable from a computational point of view. However, in [10] it has been shown that a finite prediction horizon leads to a difference between the predicted system behavior and the actual one. Consequently, a priori stability guarantees are hard to be made. In this contexts, stability is often augured by statements beginning with "if the prediction horizon is taken large enough then ..." [16, 48] whereby the question what large enough means remains often unanswered. *Finite horizon MPC schemes with guaranteed stability* bypass the *computational cost-stability-dilemma* by adding *terminal penalty terms* to the cost function (*terminal cost*) and/or consider *terminal constraints.* Here the key idea is to define an asymptotically stable *terminal region* which has to be reached within a single prediction horizon. Loosely speaking, the terminal region can be understood as an ensured domain of attraction of the desired set point or trajectory within which the asymptotic stability can be proven. However, enforcing the system to be inside the terminal region withing a single horizon length considerably reduces the operating range of MPC. In addition, the formulation of the terminal region and the terminal cost may be challenging in general [16] such that one can say, the *computational cost-stability-dilemma* is rather transformed into a *computational cost-operating range-dilemma* than solved. As in both dilemmas the computational cost is involved, it can be seen as the main drawback of MPC. Alternatively, one might think of solving the optimization problem before a real-time control execution starts. This method, called *explicit MPC*, allows to derive the optimal and asymptotically stable control actions as an explicit piecewise affine function of the state and reference vector [9, 88]. Concerning nonlinear systems explicit MPC methods provide only an approximation of the optimal solution (sub-optimality) under various strong and conservative assumptions, e.g. approximation of the Lipschitz constant [47, 110]. *Limits on the available storage space* restricts often the applicability of explicit MPC schemes to linear systems of low order [126].

Another main direction for tracking control of constrained systems is the *anti-windup* scheme. The word wind-up describes the phenomenon noticed in the 1950's for controllers that contain an integrator state (like a PID controller). If the control signal reaches its limits then the integrator state would "wind up" to large values which in turn causes extensive overshoot, slow settling times and even unstable behavior. "Anti-windup" refers to the prevention of the "wind up" phenomenon. It is an augmentation to a controller which has been well-designed in the absence of constraints. The objective is to modify the controller's output in the case of constraint violation such that acceptable performance is achieved. Maybe due to their historical origin, most of the current available anti-windup schemes are proposed for linear systems subject to input saturation [53, 114, 125]. Thereby, similar to saturated controllers, LMI-based synthesis is of growing interest due to their convex optimization features. As stated in the recent papers [25, 115], anti-windup compensation for handling state constrained or nonlinear systems remain largely unsolved. Also the T-S framework seems to be barely exploited for tackling these open problems [128].

Reference governors (RG) approaches, which are also called *command governors*, form the fourth main approach for controlling constrained systems. Similar to an anti-windup a RG is an add-on to a well-designed closed-loop system which modifies the control action for preventing a constraint violation. As its name suggests, the reference command, which is traditionally a set point, is modified based on its current value and the actual state. Generally, constrained optimization problems have to be solved in real-time for this modification and thus a direct connection to the MPC framework is noticeable. The recent publication [69] gives a detailed overview of the state of the art and provides a comprehensive list of examples which range from automotive over wind and gas turbine systems to thermonuclear fusion. Most of the RG approaches are based on prediction by online simulation or Lyapunov level set calculations. A major advance compared to MPC is that stability guarantees are easier to make. This is due to the fact that stability is ensured for each set point that the RG commands to the system. The remaining question is how to vary the set point optimally such that stability is not lost, all constraints are kept and the desired equilibrium is reached as fast as possible.

Remember, in MPC the question is more like how to optimally tune the input action such that stability is achieved. Hence, the computational cost-stability-dilemma known from MPC reduces to a *computational cost problem* (solving the constrained optimization problem in real-time) which brings us to the current drawbacks of RGs [69]: First, the consideration of time-varying constraints remains largely to be developed from a theoretical point of view. This is for instance of highly interest for trajectory tracking where the available input amplitude for compensating the tracking error changes over time (depending on the current feedforward input signal). Second, compared to MPC the required computational effort for solving the optimization problem in real-time is not discussed in-depth. Third, in the RG theory it is frequently assumed that the set points and thus the modification range are given. However, adapting or changing the set point (due to external conditions) might cause an undesired feedback. Forth, a disturbance is often assumed to be set-bounded which leads to feasibility problems in case of large (external) disturbances, like large wind gusts in wind energy or aerospace systems [60]. While the first drawback (time-varying constraints) seem indeed, up to the author's knowledge, largely unsolved, the remaining ones has been tackled in a first approach in [21] at least for single-input linear systems. The authors propose a Lyapunov-based set point governor which automatically calculates a new asymptotically stable initial set point in case of disturbance or changed external conditions. The related optimization problem possesses an analytical solution. Concerning nonlinear systems only a few tries have been recently made to tackle the mentioned drawbacks: In [17] the nonlinear system is approximated in a piecewise affine manner whereby the computation of the reference signal will be less computationally demanding. In [116] an incremental step reference governor is proposed for load conditioning of hybrid fuel cell and gas turbine power plants. The generator load is discretized and the RG verifies if an incremental step change of the load is allowed, e.g. does not cause a plant shutdown. Contrary to the previously explained four tracking control strategies, the fifth and final one realizes the tracking of a desired set point which is not a set point of the system at hand. Lots of technical control systems operate in such a manner by switching between several subsystems. For instance, a heating system is automatically turned on and off with the objective to keep a desired temperature which can be manually changed. Such a control concept is called *hybrid automaton* [52]. Stability concepts for hybrid automata deviate from the classical definitions in control theory. *Practical* or *region stability* is considered where the objective is to determine an invariant region in the state space within which each trajectory remains after a finite time [99, 124].

1.2 Scope of the Thesis

To sum up the previous section, the research in the field of tracking control for constrained dynamical systems is multifarious and important results are available in the literature. However, there are still open problems to be addressed. For contributing to some of these problems, we are interested in exploiting the *Takagi-Sugeno (T-S) state space notation* which allows to represent a quite general class of (nonlinear and linear) systems by a combination of linear or affine mathematical models [39].

A T-S formulation allows to exploit convex optimization for estimating the domain of attraction (DA) and designing a controller for a wide range of systems. However, some problems have not or only barely been investigated yet: First, the (numerically) effective relaxation of the conservative estimate of the DA. Second, handling of system constraints beyond input amplitude saturation, e.g. input rate and state limits. Third, exploiting the T-S framework for trajectory tracking tasks.

Fig. 1.1 summarizes the scope of this thesis: As shown in the upper part of the figure, there are two approaches for generally constructing a constrained T-S model: First, identification of the input-output behavior of a system from experimental data and second, derivation from a given mathematical (theoretical) model. We focus on the second approach which directly leads to a T-S model in state space notation. The contribution (lower part of Fig. 1.1) of this thesis is subdivided into two areas of tracking control:

The first part is dedicated to the problem of set point tracking. We develop *novel linear matrix inequality* (LMI) conditions and *algorithms* for the *numerically efficient estimation of a large domain of attraction* (DA) of the closed-loop system *subject to* input amplitude, rate and state *constraints*. The control law can either be predefined or optimized simultaneously. Thereafter, the algorithms are further extended such that *multiple DAs* are obtained. The DAs *vary in their size and in their related control performance*, e.g. *nested invariant sets* can be designed. We derive *switched* and *smoothly switched controllers* such that the variety of the multiple estimations can be exploited. In addition, we enlarge the valid operation range of the closed-loop system by designing a *new reference governor*, called *GINA controller* (Governor Integrated Nominal-Value Adaptation). Reference governors have, up to the author's knowledge, not been investigated for T-S systems yet. The GINA controller modifies the reference signal of the closed-loop T-S system in real-time such that *asymptotic stability* and *robustness against (external) disturbances* is ensured while *preventing a constraint violation.*

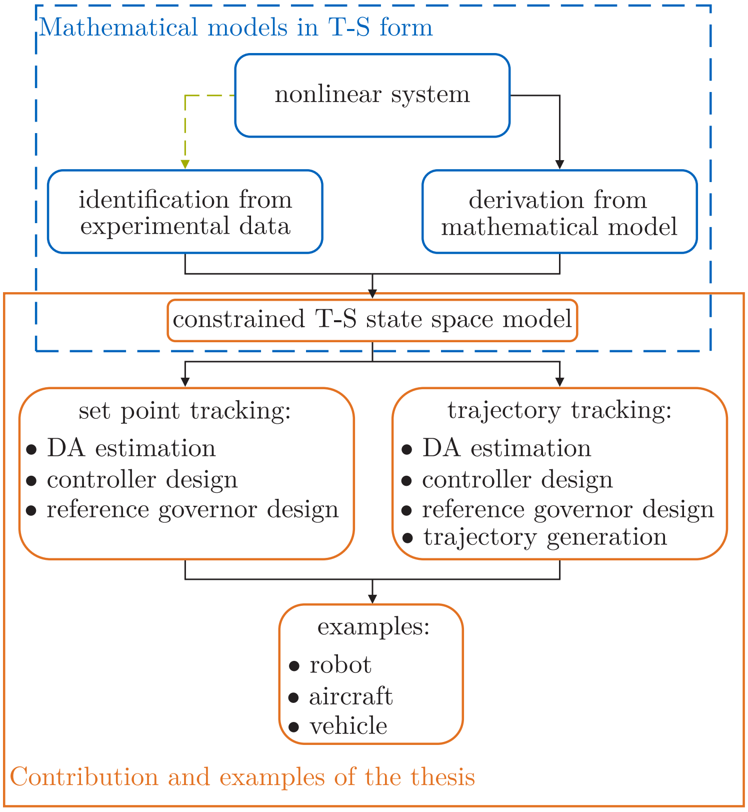

Mathematical models in T-S form
nonlinear system
identification from experimental data
derivation from mathematical model
constrained T-S state space model
set point tracking:
• DA estimation
• controller design
• reference governor design
trajectory tracking:
• DA estimation
• controller design
• reference governor design
• trajectory generation
examples:
• robot
• aircraft
• vehicle
Contribution and examples of the thesis

Figure 1.1: Scope of the thesis.

In the second part, we focus on trajectory tracking. We first show that all of our results developed so far can be also applied for following a desired trajectory. We additionally address the *trajectory generation* problem. Finally, we derive the *LISA condition* (Limits of Inputs and States are Allocated). It allows an *adjustment of the saturation limits in real-time* (e.g. input amplitude and rate) for compensating a trajectory tracking error depending on the desired trajectory. In other words, what is not required by the feedforward controller is allowed to be exploited for tracking error compensation. Consequently, both together (feedforward and error control) can better exploit the overall allowed constraints in every time step while still ensuring asymptotic stability. Based on that, we derive the *novel LISA-GINA control framework* which combines the advantages of the LISA condition and the GINA controller for tracking controller design.

The *benefit* and the *practical applicability* of the developed methods are illustrated by *numerical and technical examples* in both parts of the thesis. The examples arise from the area of robotics, aircraft and ground vehicle systems. We provide *several experimental results.*

1.3 Outline and Contributions of the Thesis

According to Fig. 1.1 the first part of this thesis is devoted to set point tracking. It comprises the Chapters 3 - 7. The second part is dedicated to the trajectory tracking problem and consists of the Chapters 8 - 12. The following overview reveals the focus of the individual chapters and puts their contributions in a nutshell:

Chapter 2: Preliminaries This chapter presents the required background. We start with reviewing dynamical fuzzy models focusing on *recurrent fuzzy* and *T-S systems*. Moreover, we give a short overview on estimating a DA based on *linear matrix inequalities* (LMIs). In this context, we briefly explain the characteristics of *LMI-based optimization problems* and show *related controller design* strategies. We also review the *hybrid automata theory* for modeling general switched or hybrid dynamical systems. Moreover, we recap the *two-degree of freedom control structure* as it will be considered for solving trajectory tracking problems. Finally, we summarize the theory of *flatness-based feedforward design*. The main contribution of this chapter is:

- presentation of the mathematical building blocks of this thesis.

Chapter 3: Set Invariance Conditions How to exploit the closed-loop structure of an *input and state constrained system* that has been formulated in T-S notation for estimating the domain of attraction (DA) of a desired equilibrium? More precisely, in this chapter we solve the question of how to determine an *as large as possible sublevel set* based on a quadratic Lyapunov function within which the asymptotic stability of the original nonlinear system is guaranteed. We develop related *set invariance conditions* in terms of LMIs and we propose a new *numerical algorithm* for *effectively computing the bounding level value*. The two key steps of the algorithm are: First, we determine the largest ellipsoidal DA derivable based on the T-S formulation of a system. Second, we iteratively enlarge the obtained bounding level value as long as the time derivative of the related Lyapunov function is negative concerning the original nonlinear system. The algorithm is straightforward to implement, which is advantageous in terms of its *broad practical applicability*. Finally, we extend the developed algorithm for *estimating multiple nested invariance sets* in order to ensure the *asymptotic stability within a large region of the state space* and, at the same time, realizing a *fast stabilization of the desired set point*. The asymptotic stabilization is directly guaranteed by nested sets if a set is activated as soon as the trajectory is within the set. We estimate each of the sets based on an *individually optimized closed-loop T-S model*. The main contributions of this chapter are:

- set invariance conditions for state, input amplitude and rate constrained systems in T-S notation.
- a numerical algorithm for the effective computation of a preferably large estimation of the DA.
- a LMI condition for estimating nested DAs.
- a numerical algorithm for the effective computation of nested DAs.

Parts of this chapter have been published in [29, 33].

Chapter 4: Switched Controller Design In this chapter, we address the problem of *designing switching controllers* for constrained T-S systems. We present two *switching strategies* for *relaxing the trade-off* between ensuring the *asymptotic stability* of a desired set point *within a large region of the state space and*, at the same time, realizing its *fast stabilization*. Both approaches are based on estimating several DAs according to Chapter 3. While the first strategy is restricted to *nested invariance sets* we bypass this limitation in the second one by invoking arguments of the *multiple Lyapunov functions* framework. The main contributions of this chapter are:

- two asymptotically stable switched controller design strategies for relaxing the trade-off between fast stabilization and a large DA of a set point.

Parts of this chapter are based on [32].

Chapter 5: Smooth Switching As a matter of fact a strict switching of a controller results in a jump of the input signal which entails a jolt to the system. Due to that, in this chapter we *extend* our *switching strategies* from Chapter 4 *to smooth switching*. The key element of the extension is to *guarantee a single defined change of the active Lyapunov function* while smoothly blending. In addition, we establish an *unified and linguistically interpretable smoothing* of a switched T-S system by formulating the switching conditions as a recurrent fuzzy system which we call *RFS-switching*. In this context we show that a hybrid automaton model (see Chapter 2) can be seen as a switched T-S system and thus also smoothed by RFS-switching. The main contributions of this chapter are:

- smoothing of the switching controller design strategies of the previous chapter.
- unified and linguistically interpretable smoothing of switching conditions.

This chapter is partly based on [30, 32, 35].

Chapter 6: Governor Integrated Nominal-Value Adaptation We show a *new reference governor* approach called *GINA* (Governor Integrated Nominal-Value Adaptation) controller to ensure the *stabilization* of a desired equilibrium if the state vector is *outside of a nominal estimated DA*, e.g. according to Chapter 3. The GINA controller allows *guarantees on stability* to be made in a *large region of the state space* for all systems that can be written in T-S form. We establish *different ways for implementing* the reference governor in order to deal with *available computational power* and for ensuring the *practical applicability* to *a broad range of systems*: First, we develop an algorithm providing an *optimal real-time computation* of the reference signal. Second, we show possible options for *relaxing its numerical complexity*. Third, we introduce an *iterative formulation of the algorithm* where most of the calculations to be made are done before a real-time execution starts. The main contributions of this chapter are:

- unified command governor approach for guarantees on stability to be made in a large region of the state space.
- five strategies and corresponding algorithms for the implementation of the command governor approach.

This chapter contains results shown in [31].

Chapter 7: Examples We *demonstrate the effectiveness and the broad practical applicability* of the methods presented in the Chapters 3 to 5 for set point tracking control in simulation and experiment. Three different technical applications are considered arising from the areas robotics and ground vehicle systems. We first consider a commonly known nonlinear benchmark example; the *inverted pendulum on cart system*. The second example is a *Ballbot* which is a promising *new variant of an unstable mobile robot* that balances upright on a sphere. Concerning the second application area, we investigate *active cruise control* (ACC) which is an advanced driver assistance system in modern cars. The main contribution of this chapter is:

- demonstration of the effectiveness and the practical applicability of the methods proposed in the Chapters 3 to 5.

The equations of motion of some of the examples in this chapter as well as parts of the results are based on [32, 92]. The methods devolved within this thesis have been also successfully applied to non-technical examples arising from the field of production planning, team processes and product development [71, 83, 84, 97, 107].

Chapter 8: Flatness-based Trajectory Generation With this chapter we start the second part of the thesis which is devoted to *trajectory tracking* based on the two-degree of freedom (2-DOF) control structure. We propose a *novel flatness-based approach* for *generating performant trajectories along predefined waypoints* which is a practically relevant navigation task, e.g. in robotics. In this context performant means a *minimal trajectory transition time while precisely hitting the waypoints subject to the relevant constraints and the possibility of a replanning in real-time.* Thus, we are concerned with *performance and real-time execution* requirements. The essential part of our approach is that we formulate the trajectory generation process as an *piecewise optimization problem* such that a maximum stationary velocity (stationary movement without acceleration) is assigned to the system at each waypoint. The main contribution of this chapter is:

- a novel flatness-based approach for generating performant trajectories along predefined waypoints subject to constraints.

This chapter is mainly based on [28].

Chapter 9: Form Set Point to Trajectory Tracking In this chapter, we deal with the question when it is possible to *directly apply* the developed *set point tracking methods* (Chapters 3 to 5) *to trajectory tracking* using the well known two-degree of freedom structure. We establish two approaches to derive the tracking *error dynamics in T-S notation.* In the first one, local linear systems, which approximate the error dynamics, are convexly interpolated. In the second approach, the tracking error dynamics is embedded in a convex hull of linear systems. Based on that, we show that *all of our results derived for set point tracking are applicable for asymptotically stabilizing the tracking error dynamics* if the system constraints are a priori subdivided into two parts: one reserved for the desired trajectory (feedforward) and one for error compensation (feedback). The main contributions of this chapter are:

- two approaches to derive the tracking error dynamics in T-S notation.
- transfer of all of our results from set point to trajectory tracking.

Chapter 10: Limits of Input and State are Allocated We introduce a *new condition, called LISA* (Limits of Inputs and States are Allocated), that *avoids the a priori subdivision of the system constraints* made in the previous chapter. More precisely, we are able to *adjust the limits for compensating a trajectory tracking error* (input amplitude and rate) depending on the feedforward controller signals while still *ensuring asymptotic stability.* To this aim, we derive an *exponential decay rate* required for the

Lyapunov function of the tracking error dynamics by *analyzing the desired trajectory.* A Lyapunov function that meets this requirement is efficiently searched. We prove the asymptotic stabilization of the desired trajectory if the *DA of the tracking error dynamics varies over time* due to the adaption of the saturating signals. The main contribution of this chapter is:

- a condition for exploiting the system constraints (input amplitude, rate and state saturation) for a fast but asymptotically stable trajectory tracking.

This chapter extents the results from [34] to nonlinear systems subject to input amplitude, rate and state saturation.

Chapter 11: The LISA-GINA Control Framework In this chapter, we present an *unified framework for designing a fast and saturating trajectory tracking controller* which allows *guarantees on stability* to be made *in a large region of the error state space.* The framework is called *LISA-GINA* (Limits of Inputs and States are Allocated-Governor Integrated Nominal-Value Adaptation) which expresses that the results of Chapter 9 and Chapter 10 are brought together. In other words, a trajectory tracking controller which has been designed according to the LISA-GINA control framework allows: First, to *adjust the input and state constraints* according to the LISA condition (Chapter 10). Second, to guarantee the *asymptotic stabilization* of the desired trajectory even *if the tracking error is outside of the estimated DA* due to the GINA controller. The main contribution of this chapter is:

- unified framework for designing an asymptotically stable, fast and saturating trajectory tracking controller.

This chapter extents the results from [34] to nonlinear systems subject to input amplitude, rate and state saturation.

Chapter 12: Examples We *demonstrate the effectiveness and the broad practical applicability* of the methods presented in the Chapters 8 to 11 for trajectory tracking control in simulation and experiment. We consider two robotic systems out of Chapter 7: *the inverted pendulum on cart* and the *Ballbot* system. In addition, we investigate a *vertical take-off and landing aircraft* (VTOL). The main contribution of this chapter is:

- demonstration of the effectiveness and the practical applicability of the methods proposed in the Chapters 8 to 11.

Some experimental results of this chapter are based on [34].

Chapter 2

Preliminaries

In this chapter, we summarize the theoretically required background of the thesis. In Section 2.1, we review two strategies based on *fuzzy theory* for modeling and controlling a nonlinear dynamical system. First, *recurrent fuzzy systems* are described which allow an intuitive and linguistical interpretation of the dynamics. The second one, the *Takagi-Sugeno system* approach allows an approximation or even an exact representation of the originally nonlinear dynamics by combining linear or affine models. Especially, the Takagi-Sugeno system structure can be exploited for estimating the domain of attraction (DA) of a system's set point using *linear matrix inequalities* (LMIs). Such estimates form the basic for the stability investigations concerning set point and trajectory tracking within this thesis and will be thus presented in Section 2.2. A short introduction to *hybrid automaton* is given in Section 2.3. Concerning trajectory tracking problems, the *two-degree-of-freedom* (2-DOF) control structure (Section 2.4) and the concept of *flatness-based feedforward design* (Section 2.5) are very popular method. Both will be taken up in this thesis for effectively planning and tracking a desired trajectory.

2.1 Dynamic Fuzzy Systems

In this section, we outline two key concepts for handling nonlinear dynamical systems based on fuzzy logic: In Section 2.1.1 we start with introducing *recurrent fuzzy systems* (RFS) that enable a linguistic description of the system dynamics. Although originally arisen from fuzzy theory *Takagi-Sugeno systems*, which are presented in Section 2.1.2, are nowadays an independent and mathematical substantiated research area. For a general introduction to fuzzy systems the reader is referred to [7, 39, 70].

2.1.1 Recurrent Fuzzy Systems

Since we consider only discrete-time recurrent fuzzy systems (RFS) in this thesis, we review only the corresponding RFS formulation. Practical relevant examples for applying RFS are for instance pattern recognition, traffic simulations [63] and as shown in our [30] papers traffic control systems. A more detailed description of RFS as well as its continuous-time application can be found in [6, 43].

Conventionally, a RFS represents the dynamics of a discrete-time nonlinear system in form of linguistic difference equations

$$\text{If } \mathbf{x}(k) \text{ is } \mathbf{L}^{\mathbf{x}}_{\mathbf{j}} \text{ and } \mathbf{u}(k) \text{ is } \mathbf{L}^{\mathbf{u}}_{\mathbf{q}} \text{ then } \mathbf{x}(k+1) \text{ is } \mathbf{L}^{\mathbf{x}}_{\mathbf{w}(\mathbf{j},\mathbf{q})} \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the input vector and k is the iteration index. The vectors $\mathbf{L}^{\mathbf{x}}_{\mathbf{j}} = \left[L^{x_1}_j, \ldots, L^{x_n}_j\right]^T$ and $\mathbf{L}^{\mathbf{u}}_{\mathbf{q}} = \left[L^{u_1}_q, \ldots, L^{u_m}_q\right]^T$ summarize linguistic values for each element x_i, $i \in \mathbb{N}_{1:n}$, and u_p, $p \in \mathbb{N}_{1:m}$, respectively. The amount of linguistical values have to be defined, meaning $L^{x_i}_j$ and $L^{u_p}_q$ with $j \in \mathbb{N}_{1:w}$ and $q \in \mathbb{N}_{1:z}$, respectively. The index vectors $\mathbf{j}$, $\mathbf{q}$ summarize the appropriate linguistical characteristics of a rule and the linguistic vector of the rules' conclusions in (2.1) is $\mathbf{L}^{\mathbf{x}}_{\mathbf{w}(\mathbf{j},\mathbf{q})}$. The index vector $\mathbf{w}$ is defined analogous to $\mathbf{j}$ and denotes the mapping $(\mathbf{j}, \mathbf{q}) \rightarrow \mathbf{w}(\mathbf{j}, \mathbf{q})$.

Example 2.1.1. Consider two state variables both having the same characteristics $L^{x_i}_1 = \text{small}$, $L^{x_i}_2 = \text{large}$, $i \in \mathbb{N}_{1:2}$ and a single input with two linguistic values $L^u_1 = \text{on}$, $L^u_1 = \text{off}$. Then a single rule (2.1) might be

$$\text{If } \mathbf{x}(k) \text{ is } [\text{small}, \text{large}]^T \text{ and } \mathbf{u}(k) \text{ is } [\text{on}] \text{ then } \mathbf{x}(k+1) \text{ is } [\text{large}, \text{small}]^T. \tag{2.2}$$

The linguistic difference equations (2.1) can be also interpreted as a deterministic linguistic automaton. The linguistic state vectors $\mathbf{L}^{\mathbf{x}}_{\mathbf{j}}$ represent the automaton states and the inputs $\mathbf{L}^{\mathbf{u}}_{\mathbf{q}}$ are the events which lead to a state transition. Fig. 2.1 shows the automaton description of the Example 2.1.1. The rule (2.2) is highlighted.

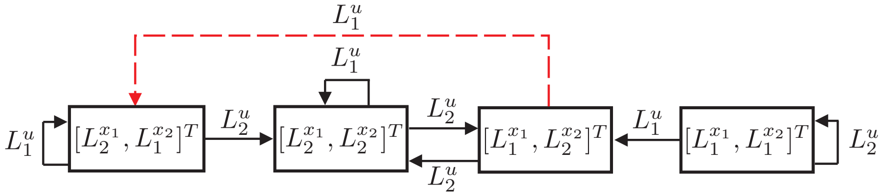
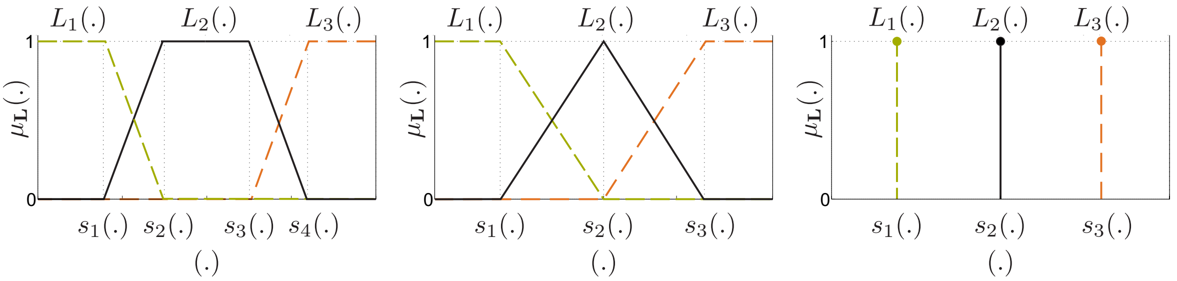

L^u_1
L^u_1
L^u_1
$[L^{x_1}_2, L^{x_2}_1]^T$
L^u_2
$[L^{x_1}_2, L^{x_2}_2]^T$
L^u_2
L^u_2
$[L^{x_1}_1, L^{x_2}_2]^T$
L^u_1
$[L^{x_1}_1, L^{x_2}_1]^T$
L^u_2

Figure 2.1: Deterministic automaton schematic of a recurrent fuzzy system.



(a) Trapezoids

(b) Triangular

(c) Singletons

Figure 2.2: Membership functions.

Finally, the RFS can be also formulated as a nonlinear difference equation. Therefore, the linguistic values have to be characterized by fuzzy compact sets using *membership functions* $\mu_{L_j}^{x_i}$ and $\mu_{L_q}^{u_p}$. These functions assign the membership of crisp valued signals x_i, u_p to the linguistic values. Conventionally, normalized membership functions are considered, meaning that the convex sum property

$$\sum_j \mu_{L_j}^{x_i} = \sum_q \mu_{L_q}^{u_p} = 1, \quad \mu_{L_j}^{x_i} \geq 0,\ \mu_{L_q}^{u_p} \geq 0 \tag{2.3}$$

is satisfied. Simple but often used are trapezoid-shaped and triangular functions as well as singletons as shown in Fig. 2.2(a), Fig. 2.2(b) and 2.2(c), respectively. The "(.)" is in place of the corresponding variable x_i or u_p. The membership functions are completely defined by *core position* vectors $\boldsymbol{s}(.) = [s_1(.), s_2(.), \ldots]$ which declare the crisp values where a membership function reaches its maximum. Singleton membership functions are often denoted by $\boldsymbol{s}_{\mathbf{L}}(.)$ as they are only active at its core position. The rule base (2.1) is inferred by classic fuzzy logic and the conclusion is *defuzzified* in order to obtain a crisp valued $\mathbf{x}(k+1)$. The nonlinear difference equation is given by

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) = \sum_{\mathbf{j},\mathbf{q}} \mathbf{s}^{\mathbf{x}}_{\mathbf{L}_{\mathbf{w}(\mathbf{j},\mathbf{q})}} \prod_i \mu_{L_j}^{x_i}(x_i) \prod_p \mu_{L_q}^{u_p}(u_p). \tag{2.4}$$

2.1.2 Takagi-Sugeno Systems

A Takagi-Sugeno (T-S) system consists of a family of linear or affine dynamic systems which are smoothly blended by fuzzy membership functions. A T-S model represents a nonlinear system either exactly or approximatively with definable accuracy. In the following, we will summarize the relevant modeling procedures for deriving a continuous-time T-S system as well as related T-S controllers. The closely related discrete-time case can be reviewed in [39, 113].

Definition 2.1.1. A *sector nonlinearity* (SE-NL) T-S system

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{z}_s)\left(\mathbf{A}_i\mathbf{x} + \mathbf{B}_i\mathbf{u}\right) \tag{2.5}$$

consists of r linear subsystems $(\mathbf{A}_i, \mathbf{B}_i)$ that are blended together by scalar, nonlinear functions $h_i(\mathbf{z}_s)$, with $\mathbf{z}_s(\mathbf{x}, \mathbf{u})$. The SE-NL T-S system, meaning the linear subsystems $(\mathbf{A}_i, \mathbf{B}_i)$ as well as the blending functions $h_i(\mathbf{z}_s)$, is obtained by transforming a continuous-time nonlinear system model

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}, \mathbf{u})\mathbf{x} + \mathbf{B}(\mathbf{x}, \mathbf{u})\mathbf{u}, \tag{2.6}$$

with the state vector $\mathbf{x} \in \mathbb{R}^n$ and the input vector $\mathbf{u} \in \mathbb{R}^m$ by applying the sector nonlinearity approach [113]. The model (2.6) equals to (2.5) in a defined range where the blending functions fulfill the convex sum property

$$\sum_{i=1}^{r} h_i(\mathbf{z}_s) = 1, \quad h_i(\mathbf{z}_s) \geq 0 \ \forall\, h_i(\mathbf{z}_s). \tag{2.7}$$

More precisely, the elements of the premise vector $\mathbf{z}_s$ are the state and input variables that nonlinearly affect (2.6). The sets

$$\mathcal{M}_{s,x} = \left\{x_i, i \in \mathbb{N}_{1:n} : x_i \in \mathbf{z}_s,\ \forall\, i\right\}, \tag{2.8a}$$

$$\mathcal{M}_{s,u} = \left\{u_i, i \in \mathbb{N}_{1:m} : u_i \in \mathbf{z}_s,\ \forall\, i\right\}, \tag{2.8b}$$

summarize those variables henceforth. The sector nonlinearity approach is most often described based on examples [113]. Nevertheless, we will try to state a general procedure in the following: First, we define a compact set of the premise variables by

$$x_i^- < 0 < x_i^+, \quad \forall\, x_i \in \mathcal{M}_{s,x}, \tag{2.9a}$$

$$u_i^- < 0 < u_i^+, \quad \forall\, u_i \in \mathcal{M}_{s,u} \tag{2.9b}$$

where x_i^- and u_i^- represent the lower and x_i^+ as well as u_i^+ the upper bounds of the variables (2.8). The set (2.9) bounds the operating region, the *universe of discourse* (UoD), of the SE-NL T-S system. This is also the region, where (2.5) and (2.6) behave equivalent. To obtain that, we calculate the extreme values

$$\overline{\theta}_k(\mathbf{z}_s) = \max(\theta_k(\mathbf{z}_s)), \quad \underline{\theta}_k(\mathbf{z}_s) = \min(\theta_k(\mathbf{z}_s)) \tag{2.10}$$

within the set (2.9) for all $\hat{r}$ nonlinear functions $\theta_k(\mathbf{z}_s)$, $k \in \mathbb{N}_{1:\hat{r}}$, in $\mathbf{A}(\mathbf{x},\mathbf{u})$ and $\mathbf{B}(\mathbf{x},\mathbf{u})$. Then weighting functions

$$\mu_{k,1} = \frac{\theta_k(\mathbf{z}_s) - \underline{\theta}_k(\mathbf{z}_s)}{\overline{\theta}_k(\mathbf{z}_s) - \underline{\theta}_k(\mathbf{z}_s)}, \quad \mu_{k,2} = \frac{\overline{\theta}_k(\mathbf{z}_s) - \theta_k(\mathbf{z}_s)}{\overline{\theta}_k(\mathbf{z}_s) - \underline{\theta}_k(\mathbf{z}_s)} \tag{2.11}$$

with $\mu_{k,1} + \mu_{k,2} = 1$ and $\mu_{k,i} \geq 0$, $i \in \mathbb{N}_{1:2}$, are derived which denote the membership of each nonlinearity to its maximum $\overline{\theta}_k(\mathbf{z}_s)$ and minimum value $\underline{\theta}_k(\mathbf{z}_s)$. In other words, if $\mu_{k,1} = 1$ then $\theta_k(\mathbf{z}_s) = \overline{\theta}_k(\mathbf{z}_s)$ and if $\mu_{k,2} = 1$ the value of the nonlinearity is $\theta_k(\mathbf{z}_s) = \underline{\theta}_k(\mathbf{z}_s)$. The blending functions $h_i(\mathbf{z}_s)$ are given by aggregating one membership function (2.11) of each nonlinearity. Consequently, by permuting all possible combinations, we obtain $r = 2^{\hat{r}}$ different blending functions

$$h_1(\mathbf{z}_s) = \mu_{1,1} \cdot \mu_{2,1} \cdot \ldots \cdot \mu_{\hat{r},1}, \tag{2.12a}$$

$$h_2(\mathbf{z}_s) = \mu_{1,2} \cdot \mu_{2,1} \cdot \ldots \cdot \mu_{\hat{r},1}, \tag{2.12b}$$

$$\vdots$$

$$h_{2^r}(\mathbf{z}_s) = \mu_{1,2} \cdot \mu_{2,2} \cdot \ldots \cdot \mu_{\hat{r},2} \tag{2.12c}$$

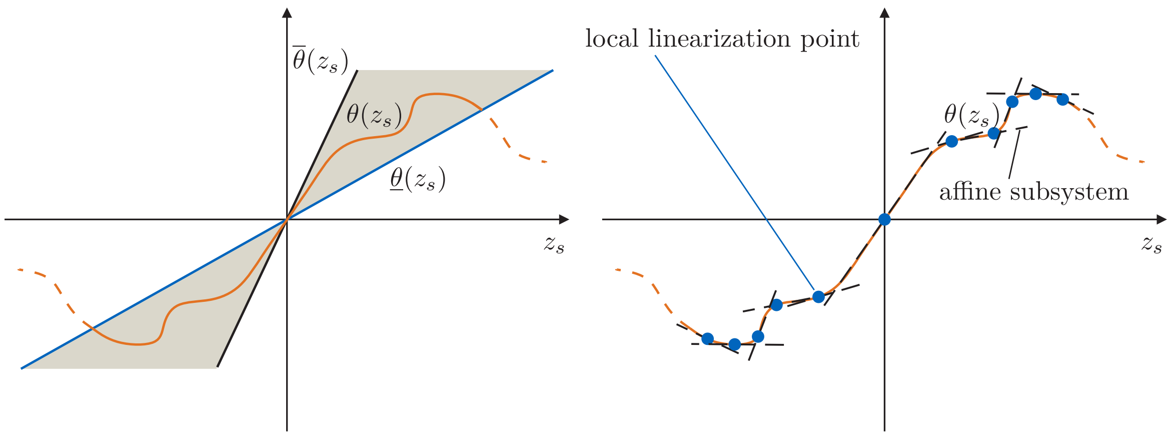
and thus subsystems within (2.5). For instance, $h_2(\mathbf{z}_s)$ is obtained from $h_1(\mathbf{z}_s)$ by changing $\mu_{1,1}$ to $\mu_{1,2}$. The subsystems $(\mathbf{A}_i, \mathbf{B}_i)$, $i \in \mathbb{N}_{1:\hat{r}}$, belonging to $h_i(\mathbf{z}_s)$ are given by inserting the extreme values (2.10) according to (2.12) into (2.6). Hence, as long as the state and input variables remain in the UoD (2.9) the convex sum property (2.7) is fulfilled and the SE-NL T-S system (2.5) is identical to (2.6).

The underlying idea of the SE-NL T-S formulation is summarized in Fig. 2.3(a) considering a single premise variable z_s and one nonlinearity $\theta(z_s)$. The nonlinearity is at least locally (solid orange line of $\theta(z_s)$) within the convex hull spanned by $\overline{\theta}(z_s)$ and $\underline{\theta}(z_s)$, meaning $\theta(z_s) \in \text{co}\left\{\overline{\theta}(z_s), \underline{\theta}(z_s)\right\}$. Consequently, $\theta(\mathbf{z})$ can be exactly represented by a convex combination of $\overline{\theta}(z_s)$ and $\underline{\theta}(z_s)$ which form the linear subsystems of the related T-S model. The size of the convex hull is defined by the compact set (2.9).

Example 2.1.2. Consider the system

$$\dot{\mathbf{x}} = \begin{bmatrix} \theta_1(\mathbf{z}_s) & 1 \\ 0 & 6 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \theta_2(\mathbf{z}_s) \end{bmatrix} u \tag{2.13}$$

of the form (2.6) with the nonlinearities $\theta_1(\mathbf{z}_s) = x_1^2 + x_2 + 1$ and $\theta_2(\mathbf{z}_s) = \exp(x_1)$. Consequently, the premise vector is $\mathbf{z}_s = [x_1, x_2]^T$. The maximum values for the UoD (2.9a) of the T-S model are set to $\left|x_1^-\right| = x_1^+ = 1$, $\left|x_2^-\right| = x_2^+ = 6$ which results according



(a) SE-NL T-S principle

(b) LO T-S principle

Figure 2.3: Handling of nonlinearities for a T-S model formulation.

to (2.10) in $\overline{\theta}_1(\mathbf{z}_s) = 8$, $\underline{\theta}_1(\mathbf{z}_s) = -4$ and $\overline{\theta}_2(\mathbf{z}_s) = 2.72$, $\underline{\theta}_2(\mathbf{z}_s) = 0.37$. Based on that the blending functions $h_i(\mathbf{z}_s)$, $i \in \mathbb{N}_{1:4}$, and the corresponding linear subsystems

$$\begin{aligned}\mathbf{A}_1 &= \begin{bmatrix}\overline{\theta}_1(\mathbf{z}_s) & 1\\ 0 & 6\end{bmatrix}, & \mathbf{A}_2 &= \begin{bmatrix}\underline{\theta}_1(\mathbf{z}_s) & 1\\ 0 & 6\end{bmatrix}, & \mathbf{A}_3 &= \begin{bmatrix}\overline{\theta}_1(\mathbf{z}_s) & 1\\ 0 & 6\end{bmatrix}, & \mathbf{A}_4 &= \begin{bmatrix}\underline{\theta}_1(\mathbf{z}_s) & 1\\ 0 & 6\end{bmatrix},\\ \mathbf{b}_1 &= \begin{bmatrix}0 & \overline{\theta}_2(\mathbf{z}_s)\end{bmatrix}^T, & \mathbf{b}_2 &= \begin{bmatrix}0 & \overline{\theta}_2(\mathbf{z}_s)\end{bmatrix}^T, & \mathbf{b}_3 &= \begin{bmatrix}0 & \underline{\theta}_2(\mathbf{z}_s)\end{bmatrix}^T, & \mathbf{b}_4 &= \begin{bmatrix}0 & \underline{\theta}_2(\mathbf{z}_s)\end{bmatrix}^T\end{aligned} \tag{2.14}$$

can be calculated.

Definition 2.1.2. A T-S model with local defined subsystems (LO T-S system)

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{z}_s)\left(\mathbf{A}_i\mathbf{x} + \mathbf{B}_i\mathbf{u} + \mathbf{a}_i\right) \tag{2.15}$$

consists of r affine systems $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{a}_i)$ that are blended together by scalar, nonlinear functions $h_i(\mathbf{z}_s)$. The subsystems are derived by locally linearizing a continuous-time nonlinear system at a defined number of r operating points [39]. The operating points define a range where a LO T-S system approximates (2.6) with a definable approximation error. Within this range the blending functions $h_i(\mathbf{z}_s)$ are given by fuzzy basis functions [70] and fulfill the convex sum property (2.7). The premise vector $\mathbf{z}_s$ summarizes the state, input and parameter variables (similar to (2.8)) that are varied for the local linearization at the operating points. Note that the affine term $\mathbf{a}_i$ vanishes if an operating point is an equilibrium of the system.

Hence, $h_i(\mathbf{z}_s)$ equals to one if $(\mathbf{A}_i\mathbf{x} + \mathbf{B}_i\mathbf{u} + \mathbf{a}_i)$ should be activated. For instance, trapezoid-shaped functions (like the membership function in Fig. 2.2(a)) can be considered. The calculation of the affine subsystems and the design of the blending functions $h_i(\mathbf{z}_s)$ is detailed in [39, 113] and thus omitted here. The idea of the LO T-S formulation is depicted in Fig. 2.3(b) analogous to the sketch of the SE-NL T-S formulation in Fig. 2.3(a). The nonlinearity $\theta(z_s)$ is approximated by affine subsystems. The linearization points and each subsystem are marked by dots and dashed lines, respectively. Hence, it becomes clear that a LO T-S model (2.15) consisting of $r = 11$ affine subsystems approximates the nonlinearity $\theta(z_s)$ well in the same range in which the SE-NL T-S exactly reflects it.

The question which T-S representation to chose is not easy to be answered in general. In our opinion, a SE-NL T-S model (2.5) is always preferable as it exactly represents a nonlinear system. However, the modeling process can become complicated for complex nonlinear systems. Compared to that, a LO T-S system (2.15) can be automatically generated. But, the number of required subsystems for approximating a nonlinear system sufficiently well becomes often unpractical for system analyzes, controller design and real time application. In this thesis, we will focus on the widely used T-S formulation without affine terms, meaning SE-NL T-S systems (2.6) or LO T-S models (2.15) where the linearization is only done at equilibria [78, 128]. However, most of our results can be directly applied to LO T-S systems with affine terms as well.

Analogous to a T-S system, a T-S controller can be designed. The mainly used scheme is

$$\mathbf{u} = \sum_{j=1}^{r} h_j(\mathbf{z}_c)\,(\mathbf{F}_j\mathbf{x}) \tag{2.16}$$

which schedules linear state feedback matrices $\mathbf{F}_j$ (one for each T-S subsystem (2.5) or (2.15)) based on the blending functions $h_j(\mathbf{z}_c)$. They can be freely chosen as long as they fulfill the convex sum property

$$\sum_{i=1}^{r} h_j(\mathbf{z}_c) = 1, \quad h_j(\mathbf{z}_c) \geq 0 \;\; \forall\, h_j(\mathbf{z}_c) \tag{2.17}$$

(analogous to the blending functions in (2.15)). In general, the blending is designed such that the premise vector $\mathbf{z}_c$ is independent of the input signal in order to avoid algebraic loops. Hence, a UoD is spanned in analogous manner to (2.9a) by

$$x_i^- < 0 < x_i^+, \quad \forall\, x_i \in \mathcal{M}_{c,x}. \tag{2.18}$$

An affine extension to the T-S controller

$$\mathbf{u} = \sum_{j=1}^{r} h_j(\mathbf{z}_c) \left(\mathbf{F}_j \mathbf{x}\right) + \mathbf{f}_j \tag{2.19}$$

is often considered for LO T-S systems (2.15). Merging the T-S systems (2.5) and (2.15) with the controllers (2.16) and (2.19), respectively, we obtain the closed-loop T-S formulations

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} \sum_{j=1}^{r} h_i(\mathbf{z}_s) h_j(\mathbf{z}_c) \left(\mathbf{A}_i + \mathbf{B}_i \mathbf{F}_j\right) \mathbf{x}, \tag{2.20a}$$

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} \sum_{j=1}^{r} h_i(\mathbf{z}_s) h_j(\mathbf{z}_c) \left(\left(\mathbf{A}_i + \mathbf{B}_i \mathbf{F}_j\right) \mathbf{x} + \mathbf{a}_i + \mathbf{B}_i \mathbf{f}_j\right). \tag{2.20b}$$

Remark 2.1.1. If $\mathbf{z}_c = \mathbf{z}_s$ and $h_j(\mathbf{z}_c) = h_i(\mathbf{z}_s)$ then the so-called *parallel distributed compensation* (PDC) control law is obtained.

Remark 2.1.2. A RFS of Section 2.1.1 can be seen as a special type of a Takagi-Sugeno system with scalar subsystems [43, 44].

2.2 LMI-based Estimation of the Domain of Attraction

The objective of this section is to review some fundamental results from the literature for effectively estimating a domain of attraction (DA) of a system set point and designing a controller based on *convex optimization* and *linear matrix inequalities* (LMIs). We briefly explain what a LMI is and how a DA as well as a controller can be designed based on them subject to constraints. A more in-depth explanation can be found in [18, 77, 114, 118]. In particular, we first consider the presence of input saturation, to which every physical system is subjected to. We show how the saturation nonlinearity can be described by a convex combination of linear feedback controllers (similar to a T-S formulation according to the Section 2.1.2). This formulation is useful for estimating a large DA of a system set point allowing the effective occurrence of input saturation. This will be focused for linear and T-S systems in Section 2.2.2.

2.2.1 Polytope Representation of Input Saturation

If all of the input capacity is exploited, which is called *effective input saturation* or *over-saturating*, the performance of the closed-loop system is improved. To this aim, the so-called *polytopic modeling* of the saturation nonlinearity

$$\boldsymbol{\sigma}(\mathbf{u}) = \begin{cases} -u_{i,max} & \text{if } u_i \leq -u_{i,max}, \\ u_{i,max} & \text{if } u_i \geq u_{i,max}, \\ u_i & \text{else,} \end{cases} \qquad \forall\, i \in \mathbb{N}_{1:m} \tag{2.21}$$

have been proposed for linear systems [57, 58] and extended to nonlinear systems in SE-NL T-S form in [23]. The underlying idea of the polytopic modeling approach is to compose the saturation nonlinearity as a convex combination of a desired state feedback law and an auxiliary one.

In order to catchy summarize the polytopic modeling approach, we will first consider a linear saturated system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\sigma}(\mathbf{u}) \tag{2.22}$$

with $\mathbf{x} \in \mathbb{R}^n$, a input $\mathbf{u} \in \mathbb{R}^m$ and a desired state feedback controller

$$\mathbf{u} = \mathbf{F}\mathbf{x}, \quad \mathbf{F} \in \mathbb{R}^{m \times n}. \tag{2.23}$$

Fig. 2.4 sketches the following explanation concerning the scalar case $u = fx$: Suppose that an auxiliary controller $\mathbf{u} = \mathbf{H}\mathbf{x}$, $\mathbf{H} \in \mathbb{R}^{m \times n}$, exists which is less aggressive than the desired one. In other words, $|\mathbf{H}\mathbf{x}| \leq |\mathbf{F}\mathbf{x}|$ as shown in Fig. 2.4. Hence, as long as the auxiliary controller does not saturate, the output of the saturation nonlinearity can be exactly computed by the convex combination of both control laws. In order to formulate that, we define a set $\mathcal{V}$ of all possible $m \times m$ boolean diagonal matrices

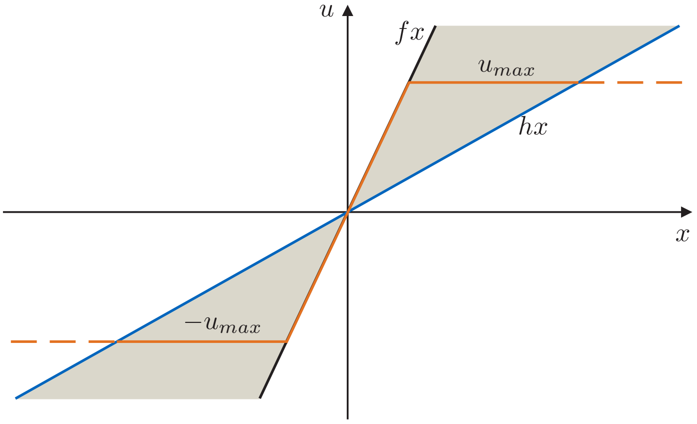


Figure 2.4: Polytopic model of the saturation nonlinearity $\boldsymbol{\sigma}(\mathbf{u})$.

$\mathbf{E}_s$, $s \in 2^m$ and $\mathbf{E}_s^- = \mathbf{I} - \mathbf{E}_s$ with $\mathbf{I}$ denoting the identity matrix. Then the convex combination is

$$\boldsymbol{\sigma}(\mathbf{F}\mathbf{x}) = \mathrm{co}\left\{ \left(\mathbf{E}_s\mathbf{F} + \mathbf{E}_s^-\mathbf{H} \right) \mathbf{x} : s \in \mathbb{N}_{1:2^m} \right\}. \tag{2.24}$$

The gray area in Fig. 2.4 depicted where the saturated control law can be modeled by (2.24). As a consequence, the closed-loop dynamics can be computed from the polytopic model

$$\dot{\mathbf{x}} = \sum_{s=1}^{2^m} \mu_s \left(\mathbf{A} + \mathbf{B} \left(\mathbf{E}_s\mathbf{F} + \mathbf{E}_s^-\mathbf{H} \right) \right), \quad \sum_{s=1}^{2^m} \mu_s = 1,\ 0 \geq \mu_s \geq 1. \tag{2.25}$$

That concept is very similar to a SE-NO T-S formulation (2.15): while the SE-NL T-S model represents a nonlinearity by a convex combination of linear systems, the saturation function is represented by a convex combination of two linear control laws. We state the following Lemma:

Lemma 2.2.1 (see [57, 114])**.** *Let a saturated linear controller* $\mathbf{u} = \boldsymbol{\sigma}(\mathbf{F}\mathbf{x})$*,* $\mathbf{F} \in \mathbb{R}^{m \times n}$ *for a system* (2.22)*, and an auxiliary controller with the feedback matrix* $\mathbf{H}$*,* $\mathbf{H} \in \mathbb{R}^{m\times n}$*, be given. Then* (2.24) *and* (2.25) *is fulfilled for all* $\mathbf{x}$ *inside the polytope*

$$\mathcal{L}(\mathbf{H}) = \left\{ \mathbf{x} \in \mathbb{R}^n : \left| \mathbf{h}_i^T \mathbf{x} \right| \leq u_{max,i},\ \forall\, i \in \mathbb{N}_{1:m} \right\}. \tag{2.26}$$

An extension of Lemma 2.2.1 for handling input signals with nested saturation functions (e.g. amplitude and rate limits) is given in [13]. Additionally, Lemma 2.2.1 has been naturally expanded to SE-NL T-S systems (2.5):

Lemma 2.2.2 (see [23])**.** *Let a T-S controller* (2.16) *for a SE-NL T-S system* (2.5) *with a saturating input and an auxiliary T-S controller with the feedback matrices* $\mathbf{H}_j$*,* $\mathbf{H}_j \in \mathbb{R}^{m\times n}$ *and* $j \in \mathbb{N}_{1:r}$*, be given. Then it follows that*

$$\boldsymbol{\sigma} \left(\sum_{j=1}^{r} h_j(\mathbf{z}_c) \left(\mathbf{F}_j \mathbf{x} \right) \right) = \mathrm{co} \left\{ \sum_{j=1}^{r} h_j(\mathbf{z}_c) \left(\mathbf{E}_s \mathbf{F}_j + \mathbf{E}_s^- \mathbf{H}_j \right) \mathbf{x} : s \in \mathbb{N}_{1:2^m} \right\} \tag{2.27}$$

and as a consequence, the closed-loop dynamics equals to the polytopic T-S model

$$\dot{\mathbf{x}} = \sum_{i=1}^{r} \sum_{j=1}^{r} \sum_{s=1}^{2^m} h_i(\mathbf{z}_s) h_j(\mathbf{z}_c) \mu_s \left(\mathbf{A}_i + \mathbf{B}_i \left(\mathbf{E}_s \mathbf{F}_j + \mathbf{E}_s^- \mathbf{H}_j \right) \right), \tag{2.28}$$

with $\sum_{s=1}^{2^m} \mu_s = 1$, $0 \geq \mu_s \geq 1$ *for all* $\mathbf{x}$ *inside the polytope*

$$\mathcal{L}(\tilde{\mathbf{H}}) = \bigcap_{j=1}^{r} \left(\mathcal{L}(\mathbf{H}_j) \right), \tag{2.29}$$

where each subpolytope $\mathcal{L}(\mathbf{H}_j)$ *is defined according to* (2.26)*.*

In other words, if $\mathbf{x} \in \mathcal{L}(\tilde{\mathbf{H}})$ then (2.27) is fulfilled. The corresponding convex hull is spanned by the linear controllers $\mathbf{F}_j$ and $\mathbf{H}_j$ which leads to the T-S system (2.28).

2.2.2 Quadratic Estimate and Controller Design

The controller design for a T-S system as well as a related estimation of a domain of attraction (DA) at the equilibrium $\mathbf{x}^*$ are often formulated as *convex optimization problems* based on a quadratic Lyapunov function

$$V_{\mathbf{x}^*} = (\mathbf{x} - \mathbf{x}^*)^T \mathbf{P} (\mathbf{x} - \mathbf{x}^*) \tag{2.30}$$

with the positive definite matrix $\mathbf{P} > \mathbf{0}$. Such optimization problems are tagged by the fact that both, the optimization objective function and all constraints are convex [118]. This attribute yields to an unique solution (global optimum) and allows an efficient solving of those problems. A constraint in form of a *linear matrix inequality* (LMI) is always convex. Roughly speaking, a LMI is an inequality where both sides consist of a sum of matrices. In each summand a maximum of one unknown variable is allowed (see Example 2.2.1).

Example 2.2.1 (see [18])**.** A controller $\mathbf{u} = \mathbf{F}\mathbf{x}$ globally asymptotically stabilizes the equilibrium $\mathbf{x}^* = \mathbf{0}$ of a linear system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2.31}$$

with $\mathbf{x} \in \mathbb{R}^n$, an input $\mathbf{u} \in \mathbb{R}^m$ if a function $V_{\mathbf{0}} = \mathbf{x}^T\mathbf{P}\mathbf{x}$ exists that fulfills:

$$\mathbf{P} > \mathbf{0} \tag{2.32a}$$

$$(\mathbf{A} + \mathbf{B}\mathbf{F})^T \mathbf{P} + \mathbf{P} (\mathbf{A} + \mathbf{B}\mathbf{F}) < \mathbf{0}. \tag{2.32b}$$

These two conditions are LMIs which prove the global asymptotic stability based on a quadratic Lyapunov function. It is commonly known that the function has to be positive definite (see (2.32a)) and its derivative (2.32b) to be negative definite for all $\mathbf{x} \neq \mathbf{0}$.

If in addition to the Lyapunov function the linear controller should be searched then (2.32b) is not a LMI anymore as in some terms of the sum both unknown variables $\mathbf{P}$ and $\mathbf{F}$ appear. However, the problem can be reformulated such that it becomes a LMI again:

Theorem 2.2.1 (see [18])**.** *If there exist matrices* $\mathbf{Q}$ *and* $\tilde{\mathbf{F}}$ *such that*

$$\mathbf{Q} > \mathbf{0} \tag{2.33a}$$

$$\mathbf{Q}\mathbf{A}^T + \tilde{\mathbf{F}}^T\mathbf{B}^T + \mathbf{A}\mathbf{Q} + \mathbf{B}\tilde{\mathbf{F}} < \mathbf{0}. \tag{2.33b}$$

is fulfilled then the control law $\mathbf{u} = \mathbf{F}\mathbf{x}$ *globally asymptotically stabilizes the equilibrium* $\mathbf{x}^* = \mathbf{0}$ *of a linear system* (2.31) *whereby* $\tilde{\mathbf{F}} = \mathbf{F}\mathbf{Q}$*.*

In other words, (2.32) is multiplied with $\mathbf{Q} = \mathbf{P}^{-1}$ from the left and the right-hand side. Substituting $\mathbf{F} = \tilde{\mathbf{F}}\mathbf{P}$ leads to the LMIs (2.33).

Due to the fact that a SE-NL T-S system (2.6) and a LO T-S model (2.15) without affine terms (operating points are set points) consists of a convexly weighted combination of several linear system, the LMIs (2.32) can be naturally extended for proving the asymptotic stability in the large:

Theorem 2.2.2 (see [113])**.** *The origin* $\mathbf{x}^* = \mathbf{0}$ *of the closed-loop SE-NL T-S system* (2.20a) *(or a LO T-S model without affine terms) is globally asymptotically stable if there exists a matrix* $\mathbf{P} > \mathbf{0}$ *such that*

$$\left(\mathbf{A}_i + \mathbf{B}_i\mathbf{F}_j\right)^T\mathbf{P} + \mathbf{P}\left(\mathbf{A}_i + \mathbf{B}_i\mathbf{F}_j\right) < \mathbf{0}, \quad \forall\, i, j \in \mathbb{N}_{1:r}. \tag{2.34}$$

The proof relies heavily on the fact that each convex combination of linear systems that share a common quadratic Lyapunov function results in a system for which the same Lyapunov function is valid [68, 80]. The result in Theorem (2.2.2) can be relaxed, either by reducing the number of LMI conditions or by diminishing its conservatism [66, 113]. For instance, one typical relaxation which mainly intents to reduce the number of LMIs is a common input matrix:

Theorem 2.2.3 (see [113])**.** *The origin* $\mathbf{x}^* = \mathbf{0}$ *of the closed-loop SE-NL T-S system* (2.20a) *(or a LO T-S model without affine terms) is globally asymptotically stable if the subsystems have a common input matrix* $\mathbf{B} = \mathbf{B}_i$*,* $i \in \mathbb{N}_{1:r}$ *and there exists a matrix* $\mathbf{P} > 0$ *such that*

$$\left(\mathbf{A}_i + \mathbf{B}\mathbf{F}_i\right)^T\mathbf{P} + \mathbf{P}\left(\mathbf{A}_i + \mathbf{B}\mathbf{F}_i\right) < \mathbf{0}, \quad \forall\, i \in \mathbb{N}_{1:r}. \tag{2.35}$$

Obviously, (2.34) and (2.35) can be also written according to Theorem 2.2.1 if a T-S controller is searched.

Example 2.2.2. Rewriting (2.34) according to Theorem 2.2.1 results in

$$\mathbf{Q}\mathbf{A}_i^T + \tilde{\mathbf{F}}_j^T\mathbf{B}_i^T + \mathbf{A}_i\mathbf{Q} + \mathbf{B}_i\tilde{\mathbf{F}} < \mathbf{0}, \quad \forall\, i, j \in \mathbb{N}_{1:r}. \tag{2.36}$$

A minimal decay rate

$$V_{\mathbf{0}} = -\alpha V_{\mathbf{0}}, \quad \alpha \geq 0 \tag{2.37}$$

of the Lyapunov function can be also postulated in all of the shown LMIs by replacing the right-hand side of the Lyapunov inequality (e.g. (2.32b) or (2.36) with $-\alpha\mathbf{P}$ and $-\alpha\mathbf{Q}$, respectively [18]. Beside a minimal decay rate, LMI constraints can be formulated for ensuring that the poles of the linear subsystem remain in a specific region of the complex left half-plane, e.g. within a circle or a cone [27, 78].

The asymptotic stability of an equilibrium cannot be globally proven anymore if system constraints become relevant. To this end, we will estimate ellipsoidal (DA)

$$\mathcal{X}_{\mathbf{x}^*}(\mathbf{P}, \eta_{\mathbf{u}^*}) = \{\mathbf{x} \in \mathbb{R}^n : V_{\mathbf{x}^*} \leq \eta_{\mathbf{u}^*}\}, \tag{2.38}$$

based on a quadratic Lyapunov function (2.30), with $\mathbf{x}^*$ denoting the considered equilibrium and $\mathbf{u}^*$ the corresponding steady-state input. The bounding level value of the DA is $\eta_{\mathbf{u}^*} > 0$ and the related bounding level set is denoted

$$\partial\mathcal{X}_{\mathbf{x}^*}(\mathbf{P}, \eta_{\mathbf{u}^*}) \tag{2.39}$$

henceforth. Based on that a LMI can be formulated which guarantees that the bounding level set does not violate a state depending constraint:

Lemma 2.2.3 (see [18, 57])**.** *Let a polytope*

$$\mathcal{L}(\mathbf{M}) = \left\{\mathbf{x} \in \mathbb{R}^n : \left|\mathbf{m}_i^T\mathbf{x}\right| \leq \zeta, \; \forall\, i\right\} \tag{2.40}$$

where $\mathbf{m}_i^T$ is the i–th row of a matrix $\mathbf{M}$ be given. Then $\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}) \subset \mathcal{L}(\mathbf{M})$ if and only if the LMI

$$\mathbf{m}_i^T\mathbf{Q}\mathbf{m}_i \leq \zeta^2 \Leftrightarrow \begin{bmatrix} \zeta^2 & \mathbf{m}_i^T\mathbf{Q} \\ \mathbf{Q}\mathbf{m}_i & \mathbf{Q} \end{bmatrix} \geq \mathbf{0}, \; \forall\, i \tag{2.41}$$

with $\mathbf{Q} = \left(\frac{\mathbf{P}}{\eta_{\mathbf{0}}}\right)^{-1}$ is fulfilled.

In other words, if a certain $\mathbf{Q}$ fulfills the LMI (2.41) then the corresponding $\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}})$ is obtained by choosing a value for $\eta_{\mathbf{0}}$ and calculating $\mathbf{P} = \left(\frac{\mathbf{Q}}{\eta_{\mathbf{0}}}\right)^{-1}$. For instance, effective input saturation by the Lemmas 2.2.1 (for linear systems) and 2.2.2 (for T-S systems) can be considered by (2.41) when setting $\zeta = u_{max,i}$ and $\mathbf{m}_i = \mathbf{h}_i$.

In order to finally obtain a convex optimization problem we still require an optimization objective which should be minimized subject to the relevant LMI constraints. In this thesis, we are mainly interested in finding the largest DA (2.38) which can be measured by its volume. The corresponding ellipsoid is obtained by the *determinant maximization* (MAXDET) problem [117]

$$\begin{aligned} &\min_{\mathbf{Q}>\mathbf{0}} \; -\log\,\det\left(\mathbf{Q}\right), \\ &\quad \text{s. t.} \quad \left.\begin{aligned} &(a) \quad \ldots, \\ &(b) \quad \ldots, \\ &\;\;\vdots \end{aligned}\right\} \text{LMI constraints} \end{aligned} \tag{2.42}$$

with $\mathbf{Q} = \left(\frac{\mathbf{P}}{\eta_{\mathbf{0}}}\right)^{-1}$. Alternatively, the objective function

$$\min_{\mathbf{P}>\mathbf{0}} \; \mathrm{trace}\left(\mathbf{P}\right) \tag{2.43}$$

can be considered if the LMI constraints are depending on $\mathbf{P}$. Note that the LMI has to be either written depending on $\mathbf{P}$ or $\mathbf{Q}$, e.g. by multiplying with $\mathbf{P}$ or $\mathbf{Q}$ as required for Example 2.2.2. Beside the size of an ellipsoid, *trace* can be also considered to synthesis T-S controllers with a *guaranteed-cost*, e.g. H_∞, H_2, LQR and dissipative performance measure [59, 78].

Different toolboxes and solvers exist for LMI-based convex optimization problems. We use *YALMIP* where LMIs can be implemented straightforward without any confusing special syntax [82]. An overview of available solvers can be found in [4]. We consider two solvers: first, the MATLAB standard solver form the *LMI-LAB* which is part of the *Robust Control Toolbox* and second, the *SDPT3* which is a commonly used solver with YALMIP.

Example 2.2.3. Consider the unstable nonlinear system

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & \theta_1(z_s) \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \sigma(u) \tag{2.44}$$

with $\theta_1(z_s) = x_1^2 + 1$. Consequently, a T-S model (2.5) of (2.44) is defined by two linear subsystems $(\mathbf{A}_i, \mathbf{B})$, $i \in \mathbb{N}_{1:2}$, with the system matrices

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ 0 & \underline{\theta}_1(z_s) \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 0 & \overline{\theta}_1(z_s) \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T. \tag{2.45}$$

We set the bound of the universe of discourse (UoD) (2.9a) of x_1 (x_1 nonlinearly affects (2.45)) to $\left|x_1^-\right| = x_1^+ = 1.6$ and the input amplitude is limited to $u_{max} = 7$. Two DAs (2.38) at $\mathbf{x}^* = \mathbf{0}$, $u^* = 0$ are shown in Fig. 2.5(a) by their bounding level sets avoiding input saturation to occur and setting the bounding level value to $\eta_0 = 1$. Concerning the first DA with the bounding level value $\partial\mathcal{X}_\mathbf{0}(\mathbf{P}_1, 1)$, we searched for a Lyapunov matrix $\mathbf{P} > \mathbf{0}$ and a PDC (2.16) such that a minimal decay rate of $V_\mathbf{0} = -\alpha V_\mathbf{0}$ with $\alpha = 1.5$ is ensured. Consequently, the LMIs required for guaranteeing asymptotic stability are $\mathbf{Q} < \mathbf{0}$ and (2.36) (subject to a decay rate). LMI (2.41), with $\mathbf{m} = \mathbf{f}$, has been considered for avoiding input saturation. The optimization problem is:

$$\min_{\mathbf{Q}>\mathbf{0},\tilde{\mathbf{K}}_i} \; -\log\det(\mathbf{Q}), \tag{2.46}$$

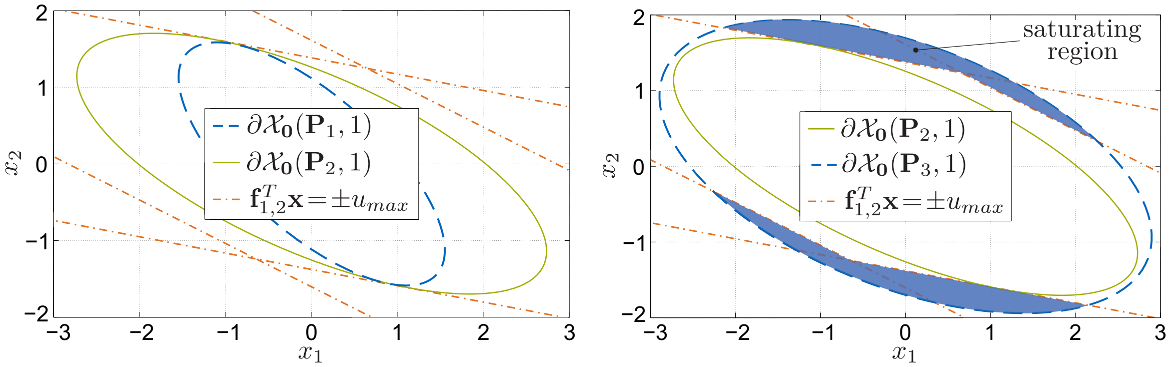
$$\text{s. t. } (a)\; \mathbf{Q}\mathbf{A}_i^T + \tilde{\mathbf{f}}_i\mathbf{b}^T + \mathbf{A}_i\mathbf{Q} + \mathbf{b}\tilde{\mathbf{f}}_i^T \leq -\alpha\mathbf{Q}, \quad i \in \mathbb{N}_{1:2},$$

$$(b)\; \begin{bmatrix} u_{max}^2 & \tilde{\mathbf{f}}_i^T \\ \tilde{\mathbf{f}}_i & \mathbf{Q} \end{bmatrix} \geq \mathbf{0}, \quad i \in \mathbb{N}_{1:2}.$$

The finally obtained PCD is

$$\mathbf{f}_1^T = \begin{bmatrix} -2.46 & -4.36 \end{bmatrix}, \quad \mathbf{f}_2^T = \begin{bmatrix} -1.07 & -5.06 \end{bmatrix}. \tag{2.47}$$

Now using this controller, we set the decay rate of the Lyapunov function $\alpha = 0$



(a) Avoiding saturation

(b) Effective saturation

Figure 2.5: T-S model based optimization of $\mathcal{X}_\mathbf{0}(\mathbf{P}_i, 1)$, $i \in \mathbb{N}_{1:3}$.

and solve (2.46) again which leads to the larger DA $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_2, 1)$. According to the LMI (2.46) (b), both ellipsoids are tangent to the input saturation lines $\mathbf{f}_{1,2}^T\mathbf{x} = \pm u_{max}$ of the T-S system. The larger ellipsoid $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_2, 1)$ is also depicted in Fig. 2.5(b) and compared to $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_3, 1)$ where effective saturation (over-saturation) is considered by the polytopic representation of input saturation (2.28). The optimization problem is:

$$\begin{aligned} &\min_{\mathbf{Q}>\mathbf{0},\tilde{\mathbf{H}}_i} \; -\log\det(\mathbf{Q}), \qquad (2.48)\\ &\text{s. t. } (a)\; \mathbf{Q}\mathbf{A}_i^T + \left(e_s\mathbf{f}_i^T\mathbf{Q} + e_s^-\tilde{\mathbf{h}}_i^T\right)^T\mathbf{B}^T + \mathbf{A}_i\mathbf{Q} + \mathbf{B}_i\left(e_s\mathbf{f}_j^T\mathbf{Q} + e_s^-\tilde{\mathbf{h}}_i^T\right) < \mathbf{0}\\ &\qquad\; (b)\; \begin{bmatrix} u_{max}^2 & \tilde{\mathbf{h}}_{i,l}^T \\ \tilde{\mathbf{h}}_{i,l} & \mathbf{Q} \end{bmatrix} \geq \mathbf{0}, \end{aligned}$$

with $i, s \in \mathbb{N}_{1:2}$. In order to obtain the largest possible DA, the auxiliary controller gains $\mathbf{H}_i$ are optimized in addition. Hence, a substitution $\tilde{\mathbf{h}}_i^T = \mathbf{h}_i^T\mathbf{Q}$ is required.

Putting the quintessence seen from this simple example in a nutshell: First, increasing the minimal decay rate of the Lyapunov function leads to a reduction of the DA. This is plausible as a decay rate restricts the possible solutions. Second, the largest DA can be found if effective input saturation is considered. Thereby, regions (not only single points) in the state space are included in the DA whereby the whole input amplitude is exploited. However, the drawback of this strategy is that the number of LMIs rises.

An equilibrium $\mathbf{x}^* \neq \mathbf{0}$ may require a steady-state input signal $\mathbf{u}^* \neq \mathbf{0}$ which in turn reduces its DA. In case that a linear state feedback law (2.23) is considered, the bounding level value of the considered closed-loop system is given by [114]

$$\eta_{\mathbf{u}^*} = \min_i \left(\eta_{\mathbf{0}} \cdot \frac{\left(u_{max,i} - |u_i^*|\right)^2}{u_{max,i}^2}\right). \qquad (2.49)$$

Consequently, the DA of an arbitrary equilibrium is directly determinable based on $\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}})$ for the origin. Unfortunately, such an analytical relation does not exist if a T-S controller (2.16) is considered. In that case, the DA has to be individually estimated for each equilibria $(\mathbf{x}^*, \mathbf{u}^*)$ we are interested in, e.g. by applying the shown estimation strategy for $\eta_{\mathbf{0}}$ subject to a coordinate shift.

2.3 Hybrid Automaton

A hybrid automaton is a general formal model of a switched or hybrid dynamical system [52]. It consists of the following components:

Control graph. A finite directed multigraph $(V; E)$. The vertices V are called *control modes*. The edges E which link the control modes together, are called *control switches*.

State variables. A finite set of state variables $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$. The number n defines the dimension of the hybrid automaton.

Switch conditions. Functions assigned to the edges. These functions define under which conditions a jump from one control mode to another one is allowed.

Flow, initial and invariant conditions. Functions assigned to the control modes. Differential equations are flow conditions that define the continuous change over time of the state variables in a certain control mode. Initial conditions show which control modes are allowed to be activated initially and the corresponding values for $\mathbf{x}_0$ for the initial time t_0. If no initial condition is assigned then initializing the hybrid automaton in an arbitrary control mode with arbitrary values $\mathbf{x}_0$ is allowed. Note, usually only one control mode is allowed to be active at a time. An invariant condition defines the region of the state vector $\mathbf{x}$ that is allowed in the corresponding control mode. Hence, a jump to a different control mode is forced, and must be allowed by a switching condition, if an invariant condition is violated.

Events. A finite set of events can (but do not have to) be assigned to an edge. An event leads to a jump (resetting) of the state variables when switching from one control mode to another one. Events are required for modeling so-called *impulsive systems* [50] which are not further considered in this thesis.

Example 2.3.1 (Temperature controller [99])**.** Fig. 2.6 shows a hybrid automaton model of a temperature controller with an internal heater engine. The temperature of the plant and the temperature of the heater engine x_p and x_e, respectively, are the state variables of the automaton. A thermostat continuously senses the temperatures and turns the heater on and off whereby the two control modes are defined. The fall and rise of the temperatures is governed by the differential equations located in each control mode. The acceptable region of the state space of the control mode "on" is given by the invariant condition $x_p \leq 25 \wedge x_e \leq 80$. If these limits are reached the heater has to be turned off. However, the controller can already switch the heater "off" as soon as

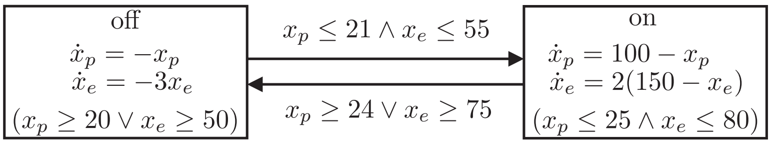

off
$\dot{x}_p = -x_p$
$\dot{x}_e = -3x_e$
$(x_p \geq 20 \vee x_e \geq 50)$
$x_p \leq 21 \wedge x_e \leq 55$
$x_p \geq 24 \vee x_e \geq 75$
on
$\dot{x}_p = 100 - x_p$
$\dot{x}_e = 2(150 - x_e)$
$(x_p \leq 25 \wedge x_e \leq 80)$

Figure 2.6: Temperature controller with an internal heater engine.

$x_p \geq 24 \vee x_e \geq 75$ (switch conditions). The heater can be turned on again as soon as the switch condition $x_p \leq 21 \wedge x_e \leq 55$ is fulfilled, but must be turned on before the temperature of the plant is below 20 or the temperature of the heater is below 50.

As the example shows, a hybrid automaton does not necessarily have an equilibrium and thus stability deviates from classical definitions in control theory.

Definition 2.3.1 (Region stability [99, 100])**.** A hybrid automaton is stable with respect to a region ϵ (called ϵ-region) if for every trajectory there exists a point of time t_p such that for $t > t_p$ the trajectory is always in the ϵ-region.

Region stability is essentially what is called practical stability in [124]. For instance, the ϵ-region of the temperature controller in Example 2.3.1 is $x_p \in [20, 25]$, $x_e \in [50, 80]$.

2.4 Two-Degree of Freedom Control Structure

An often considered control scheme for trajectory tracking problems is the *two-degree of freedom* (2-DOF) control structure [56]. It allows an independent design of the command and the disturbance response by combining a feedforward with a subsidiary feedback controller according to Fig. 2.7. The feedforward part is ideally a perfect inversion of the system and calculates a nominal input trajectory $\mathbf{u}_T$ as well as a corresponding trajectory of the state $\mathbf{x}_T$ such that the tracking output $\mathbf{y}_T$ exactly follows a desired *reference trajectory* $\mathbf{y}_{ref}$ in the absence of disturbances. All signals are defined over a time span $t \in \mathcal{T}$, $\mathcal{T} = \{t | t_0 \leq t \leq t_e\}$ where t_0 and t_e represent the starting time and the end time, respectively, of the reference trajectory. A tracking error $\mathbf{e} = \mathbf{x}_T - \mathbf{x}$ may occur for two reasons: First, the feedforward controller does not perfectly match the system's inverse, e.g. due to assumptions made while modeling the system. Second, an external disturbance or a measurement noise occurs. In both cases, the desired state trajectory $\mathbf{x}_T$ is asymptotically stabilized by superimposing the feedback signal $\mathbf{u}_e \neq \mathbf{0}$.

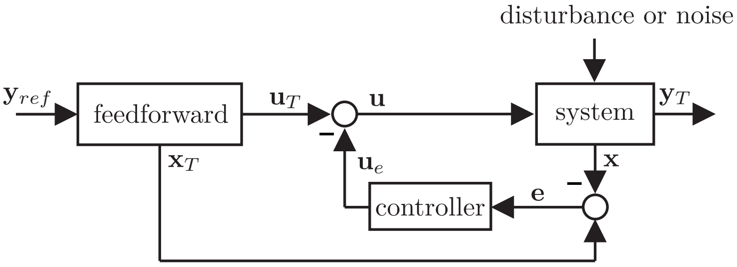


Figure 2.7: Two-Degree-off-Freedom (2-DOF) control structure.

Consequently, a desired tracking behavior can be a priori designed and thus the control performance is increased as only the tracking error has to be compensated.

If the system is subject to constraints, the trajectory tracking task becomes more complicated as both together, feedforward and feedback part, have to meet the system limits. Intuitively, a constraint can be subdivided into two parts: one for the feedforward part and the second for error compensation:

Assumption 2.4.1. Each system constraint can be formulated as a restriction ϖ_{max} of the absolute value of single state or input variables.

These absolute values can be simply subdivided into two parts

$$\varpi_{e,max} = \varpi_{max} - \varpi_{T,max} \tag{2.50}$$

with

$$\varpi_{T,max} = \rho_{\varpi}\varpi_{max}, \text{ with } 0 \leq \rho_{\varpi} < 1. \tag{2.51}$$

In other words, $\varpi_{T,max}$ and $\varpi_{e,max}$ subdivide a constraint ϖ_{max} into two: the first is valid for the feedforward and the second for error compensation. Equation (2.51) guarantees that $\varpi_{e,max} > 0$. This concept has been for instance considered in [62] for linear systems subject to input saturation.

2.5 Flatness-based Feedforward Design

The property of *differential flatness* (shortly *flatness*) was defined in [41]. Flatness enables, among others, a convenient way for calculating a feedforward controller, e.g. for the 2-DOF control structure according to Fig. 2.7. A mathematical definition can be stated as follows:

Definition 2.5.1. A general nonlinear system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{2.52}$$

with the state vector $\mathbf{x} \in \mathbb{R}^n$ and the input vector $\mathbf{u} \in \mathbb{R}^m$, rank$(\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}) = m$, is said to be *differential flat* (or *flat*) if: First, there exits a virtual output, the so-called *flat output*,

$$\mathbf{y}_f = \boldsymbol{\phi}(\mathbf{x}, u_1, \ldots, u_1^{(\zeta_1)}, \ldots, u_m, \ldots, u_m^{(\zeta_m)}), \quad \dim(\mathbf{y}_f) = \dim(\mathbf{u}) \tag{2.53}$$

that can be expressed in terms of the state vector, the input vector and a finite amount of time derivatives $u_i^{(k)}$, $k \in \mathbb{N}_{1:\zeta_i}$. Second, the complete state and input vector can be expressed in terms of $\mathbf{y}_f$ and finitely many of its time derivatives $y_{f,i}^{(k)}$, $k \in \mathbb{N}_{1:\xi_i+1}$, meaning functional relations

$$\begin{aligned} \mathbf{x} &= \boldsymbol{\psi}_x(y_{f,1}, \dots, y_{f,1}^{(\xi_1)}, \dots, y_{f,m}, \dots, y_{f,m}^{(\xi_m)}), \\ \mathbf{u} &= \boldsymbol{\psi}_u(y_{f,1}, \dots, y_{f,1}^{(\xi_1+1)}, \dots, y_{f,m}, \dots, y_{f,m}^{(\xi_m+1)}) \end{aligned} \tag{2.54}$$

exist.

In other words, if a flat output exists, then the whole dynamical system can be parametrized depending on this output. Concerning a feedforward design that can be advantageously exploited as the state and input trajectories

$$\mathbf{x}_T = \boldsymbol{\psi}_x(y_{f,1}^*, \dots, y_{f,1}^{*(\xi_1)}, \dots, y_{f,m}^*, \dots, y_{f,m}^{*(\xi_m)}), \tag{2.55a}$$

$$\mathbf{u}_T = \boldsymbol{\psi}_u(y_{f,1}^*, \dots, y_{f,1}^{*(\xi_1+1)}, \dots, y_{f,m}^*, \dots, y_{f,m}^{*(\xi_m+1)}) \tag{2.55b}$$

are analytically given depending on a desired trajectory of the flat output $\mathbf{y}_{ref} = \mathbf{y}_f^*$. Hence, the differential equations of the system (2.52) have not to be numerically solved while generating or optimizing a desired trajectory, whereby the required computational effort is reduced. The flat output is often physically or intuitively accessible and thus planning a flat output trajectory is convenient. An often used approach is to define a sufficiently smooth trajectory function for $\mathbf{y}_f^* \in \mathcal{C}^{\xi_m+1}$ for a time interval $t \in [t_0, t_e]$. For instance, based on splines or polynomials

$$y_{f,k}^*(p_{j,k}) = \sum_{j=1}^{q_k} p_{j,k} \left(\frac{t}{t_0 - t_e} \right)^{j-1}, \qquad k \in \mathbb{N}_{1:m} \tag{2.56}$$

for each component $y_{f,k}$ of the flat output. The order q_k of each polynomial depends on the boundary conditions that should be fulfilled, e.g. $q_k \geq 2n+1$ if the trajectory should start and end at a specific point in the state space $\mathbf{x}_T(t_0)$ and $\mathbf{x}_T(t_e)$, respectively. The order is thereby given by the required conditions that specify the location of the flat output and its derivatives

$$\begin{aligned} &y_{f,1}(t_0) = y_{f,1,0}^*, \dots, \quad y_{f,1}^{(\xi_1)}(t_0) = y_{f,1,0}^{*(\xi_1)}, \dots, \quad y_{f,m}(t_0) = y_{f,m,0}^*, \dots, \quad y_{f,m}^{(\xi_m)}(t_0) = y_{f,m,0}^{*(\xi_m)}, \\ &y_{f,1}(t_e) = y_{f,1,e}^*, \dots, \quad y_{f,1}^{(\xi_1)}(t_e) = y_{f,1,e}^{*(\xi_1)}, \dots, \quad y_{f,m}(t_e) = y_{f,m,e}^*, \dots, \quad y_{f,m}^{(\xi_m)}(t_e) = y_{f,m,e}^{*(\xi_m)}. \end{aligned} \tag{2.57}$$

The property of flatness is especially beneficial for linear systems as controllability means that they are flat as well. Additionally, the flat output can be constructively determined which is currently not the case for nonlinear systems in general. For a more detailed introduction of the flatness property, we refer the interested reader to [106, 127].

Example 2.5.1. According to [127],

$$y_f = \boldsymbol{\psi}^T \mathbf{x} \tag{2.58}$$

defines a flat output of a controllable linear system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u \tag{2.59}$$

with $\mathbf{x} \in \mathbb{R}^n$ and a single input u. The vector

$$\boldsymbol{\psi}^T = [0, \dots, 0, \kappa]\mathbf{Q}_s^{-1}. \tag{2.60}$$

denotes the last row of the inverse controllability matrix

$$\mathbf{Q}_s = [\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^{n-1}\mathbf{b}] \tag{2.61}$$

with $\kappa \neq 0$. The flat output and its time derivatives

$$\mathbf{z}_f = [y_f, \dot{y}_f, \dots, y_f^{(n-1)}] \tag{2.62}$$

form the flat coordinates of the system. The transformation into flat coordinates is

$$\mathbf{z}_f = \boldsymbol{\Psi}\mathbf{x} \tag{2.63}$$

whereby the transformation matrix is defined as

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}^T \\ \boldsymbol{\psi}^T \mathbf{A} \\ \vdots \\ \boldsymbol{\psi}^T \mathbf{A}^{n-1} \end{bmatrix}. \tag{2.64}$$

The input signal is computed as

$$u = y_f^{(n)} - \boldsymbol{\psi}^T \mathbf{A}^n \boldsymbol{\Psi}^{-1} \mathbf{z}_f. \tag{2.65}$$

Part I

Set Point Tracking Control

Chapter 3

Set Invariance Conditions

In this chapter, we are concerned with set invariance conditions for sector nonlinearity (SE-NL) T-S or local (LO) T-S models without affine terms for effectively estimating the domain of attraction (DA) based on quadratic Lyapunov functions. Recalling the scope of this thesis (Section 1.2) we contribute to the design of saturated controllers. A formal problem statement together with an intuitively accessible explanation how we will solve it within this chapter is given in Section 3.1. We establish LMI-based set invariance conditions to estimate a large DA, which is valid for the original nonlinear system, subject to state, input amplitude and rate constraints in Section 3.2. The conditions are embedded in a novel numerical procedure for a numerically efficient computation of the DA. We handle input rate constraints by augmenting the T-S representation with an actuator model and we formulate LMI conditions such that an over-saturating DA (*effective saturation*) can be estimated. A reduction of the required number of LMIs, which is beneficially from a numerical point of view, is thereby obtained for free. In addition, we show that the augmented T-S representation can be exploited for estimating a DA for LO T-S models with affine terms as well.

3.1 Problem Formulation

Generally, a T-S model is only able to represent the nonlinear system within a limited region of the state space including the origin. This *equivalence region* is bounded to the area where the convex sum property (2.7) is fulfilled. Hence, the state and input variables that are contained in one of the premise vectors $\mathbf{z}_s$, $\mathbf{z}_c$ and, more precisely, the universes of discourse (UoD) (2.18), (2.9), define the equivalence region. Recalling Example 2.2.3 back to mind, we see that all of the estimated DAs in Fig. 2.5 cross the UoD $\left|x_1^-\right| = x_1^+ = 1.6$ of the SE-NL T-S model. However, concluding that the DA is not valid for the original nonlinear system is not necessarily correct, but the optimality (maximized volume) of the estimated DA is questionable. So how to solve that problem, which is additionally tightened if further constraints, e.g. input rate limits, have to be taken into account. Based on that the first investigated problem is:

Problem 3.1.1. Estimate a large DA for a constrained system (state, input amplitude and rate constraints) by exploiting the T-S formulation such that the DA is valid for the original system.

In the following section, we derive a new numerical algorithm for solving Problem 3.1.1. Structurally, the algorithm is subdivided into two parts:

- First, we estimate an as large as possible domain of attraction (DA) based on LMIs subject to the T-S system's universe of discourse (UoD). We consider the UoD as optimization parameter and we formulate LMI conditions such that over-saturation is achieved for input amplitude and rate constraints.
- Second, we iteratively enlarge the critical level value obtained from the first part of the algorithm. As we may get beyond the UoD, the T-S formulation might become incorrect and thus we check the validity of the DA based on the original (nonlinear) system description.

3.2 Determining the Critical Level Value

In this section, we present a novel numerical algorithm that allows to effectively determine the critical level value of the DA (2.38). The algorithm is developed and improved step-by-step: In Section 3.2.1, we focus on estimating a DA subject to the UoD of the T-S model. To this end, we introduce LMI conditions which handle the universe of discourse (UoD) as state constraints. Thereafter, in Section 3.2.2, we incorporate the practical relevant problem if the actuator is not only subject to amplitude but also to rate saturation. We deliver LMI conditions for estimating an over-saturating DA (*effective saturation*). The obtained set invariance conditions are relaxed in Section 3.2.3, meaning we reduce the number of required LMIs and additionally diminish the conservatism of the solution. Without loss of generality, all estimations are done for the trivial equilibrium, meaning we estimate the critical level value $\eta_{\mathbf{0}}$.

3.2.1 Universe of Discourse (UoD): State Constraints

A symmetric polytope

$$\mathcal{S}_{\mathbf{x}} = \{\mathbf{x} \in \mathbb{R}^n : |x_i| \leq x_{i,max}, \ \forall \ x_i \in \mathcal{M}_{s,x}, \mathcal{M}_{c,x}\} \tag{3.1}$$

can be defined that is bounded by the smaller absolute value

$$x_{i,max} = \min(\left|x_i^-\right|, x_i^+) \tag{3.2}$$

of the UoD (2.9a), (2.18). Analogous to (3.1) a second set

$$\mathcal{S}_{\mathbf{u}} = \{\mathbf{u} \in \mathbb{R}^m : |u_i| \leq u_{i,max},\ \forall\ u_i \in \mathcal{M}_{s,u}\}\,, \tag{3.3}$$

with $u_{i,max} = \min(\left|u_i^-\right|, u_i^+)$, is spanned by the input signals which are relevant for defining the UoD (2.9b). Considering a feedback law (2.16), the set (3.3) can be written state-dependent as

$$\mathcal{S}_{\mathbf{u}} = \left\{\mathbf{x} \in \mathbb{R}^n : \left|\mathbf{f}_{i,l}^T\mathbf{x}\right| \leq u_{i,max},\ \forall\ l \in \mathbb{N}_{1:r}\ \forall\ u_i \in \mathcal{M}_{s,u}\right\} \tag{3.4}$$

with $\mathbf{f}_{i,l}^T \in \mathbb{R}^n$ denoting the i-th row vector of the l-th linear subcontroller. In order to estimate a DA, which it directly valid for the original nonlinear system, the related bounding level set (2.39) has to entirely lie within the state depending sets (3.1), (3.4):

Theorem 3.2.1. *A T-S controller* (2.16) *asymptotically stabilizes the origin* $\mathbf{x}^* = \mathbf{0}$, $\mathbf{u}^* = \mathbf{0}$ *of the original nonlinear system* (2.6) *for all* $\mathbf{x} \in \mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}})$ *if a positive definite matrix* $\mathbf{P} > \mathbf{0}$ *and a level value* $\eta_{\mathbf{0}} > 0$ *exist such that the corresponding closed-loop T-S system* (2.20a) *is globally asymptotically stable and*

$$\mathbf{g}_i^T\mathbf{Q}\mathbf{g}_i \leq x_{i,max}^2, \quad \forall\ x_i \in \mathcal{M}_{s,x}, \mathcal{M}_{c,x}, \tag{3.5a}$$

$$\mathbf{f}_{i,l}^T\mathbf{Q}\mathbf{f}_{i,l} \leq u_{i,max}^2, \quad \forall\ u_i \in \mathcal{M}_{s,u}\ \forall\ \in \mathbb{N}_{1:r}, \tag{3.5b}$$

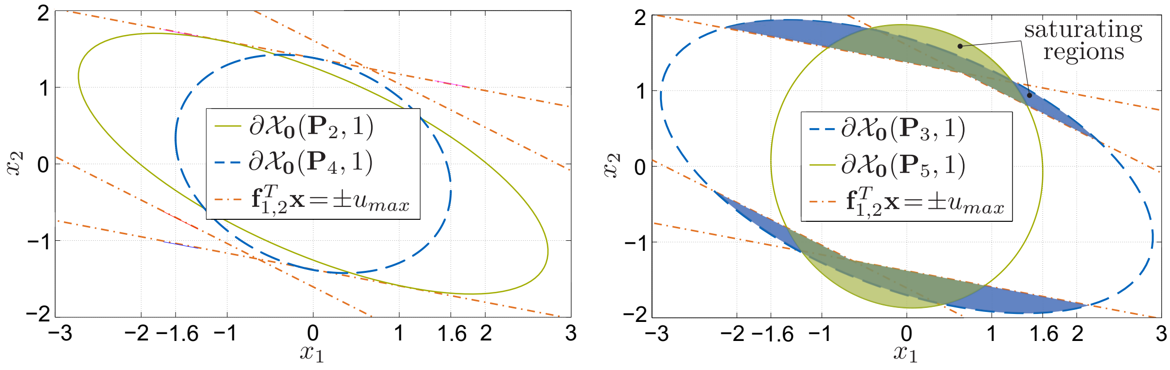
with $\mathbf{Q} = \left(\frac{\mathbf{P}}{\eta_{\mathbf{0}}}\right)^{-1}$ *is fulfilled. The vectors* $\mathbf{g}_i \in \mathbb{R}^n$ *are*

$$\mathbf{g}_i = [0,\ 0,\ \ldots,\ \underbrace{1}_{i-th\ element},\ 0,\ \ldots,\ 0]^T. \tag{3.6}$$

Proof: Replacing $\mathbf{m}_i$ by $\mathbf{g}_i$ and setting $\zeta = x_{i,max}$ in Lemma 2.2.3 results in (3.5a). Equation (3.5b) is obtained for $\mathbf{m}_i = \mathbf{f}_{i,l}$ and $\zeta = u_{i,max}$. According to Lemma 2.2.3

$$\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}) \subset \mathcal{S}_{\mathbf{x}} \cap \mathcal{S}_{\mathbf{u}} \tag{3.7}$$

is fulfilled. Consequently, if the global asymptotic stability of the origin is proven for the closed-loop T-S system, it follows directly that the closed-loop nonlinear system is asymptotically stable within (3.7) which concludes the proof. ■



(a) Avoiding saturation (b) Effective saturation

Figure 3.1: Comparison of T-S model based estimations of the DA: Without UoD restriction $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_i, 1)$, $i \in \mathbb{N}_{2:3}$, and with UoD restriction $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_i, 1)$, $i \in \mathbb{N}_{4:5}$.

Remark 3.2.1. Note that LMI (3.5a) can be also used to incorporate state constraints of the original nonlinear system while estimating a corresponding DA.

Example 3.2.1 (Example 2.2.3 cont'd)**.** As the state variable x_1 is the only premise variable, the UoD restriction is given by the LMI

$$[1,0]\mathbf{Q}[1,0]^T \leq (1.6)^2. \tag{3.8}$$

Equivalently to the optimized DAs $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_2, 1)$ and $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_3, 1)$ of Example 2.2.3 we solve the optimization problems (2.46) (with a decay rate $\alpha = 0$) and (2.48), respectively, using the PDC controller (2.47). However, thereby we add (3.8) as an additional constraint. Fig. 3.1(a) and Fig. 3.1(b) show the resulting estimates $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_4, 1)$ and $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_5, 1)$ in comparison to $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_2, 1)$ and $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_3, 1)$, respectively. As can be seen, the additional LMI constraint leads to solutions that fulfill (3.7) and are thus directly valid for the original nonlinear closed-loop system. Further examples as well as comparisons of further convex optimization objectives are detailed in [34].

3.2.2 Input Amplitude and Rate Constraints

Consider the practical relevant problem that the control input $\mathbf{u}$ is subject to amplitude saturation (2.21) and rate constraints

$$\sigma(\dot{u}_i) = \begin{cases} -\dot{u}_{i,max} & \text{if } \dot{u}_i \leq -\dot{u}_{i,max}, \\ \dot{u}_{i,max} & \text{if } \dot{u}_i \geq \dot{u}_{i,max}, \quad \forall\, i \in \mathbb{N}_{1:m} \\ \dot{u}_i & \text{else.} \end{cases} \tag{3.9}$$

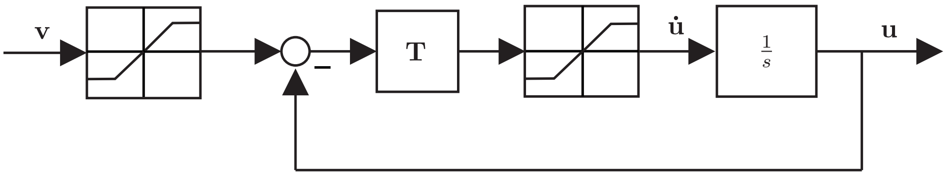


Figure 3.2: Actuator model subject to amplitude and rate saturation.

We consider these constraints by a first-order model (position-feedback-type model with speed limitation) [24, 89]

$$\dot{\mathbf{u}} = \boldsymbol{\sigma}\left(\mathbf{T}\left(\boldsymbol{\sigma}(\mathbf{v}) - \mathbf{u}\right)\right) \tag{3.10}$$

which is depicted in Fig. 3.2. The diagonal matrix $\mathbf{T} = \operatorname{diag}(\tau_1, \dots, \tau_m)$ summarizes the inverse actuator time constants τ_i and $\mathbf{v}$ denotes the input of the actuator with $u_{i,max} = v_{i,max}$. Obviously, small time constants lead to a slower time response and if $\tau_i \to \infty$ for all $i \in \mathbb{N}_{1:m}$ the actuator model (3.10) becomes an ideal rate limiter [109]. Compared to other existing modeling approaches, saturation is effectively considered in (3.10) and not avoided or approximated as for instance in [54, 114, 119].

We extend the T-S model (2.5) by (3.10) to

$$\dot{\hat{\mathbf{x}}} = \sum_{i=1}^{r} h_i(\mathbf{z}_s)\left(\hat{\mathbf{A}}_i\hat{\mathbf{x}} + \hat{\mathbf{B}}\,\boldsymbol{\sigma}\left(\mathbf{K}\hat{\mathbf{x}} + \mathbf{T}\boldsymbol{\sigma}(\mathbf{v})\right)\right) \tag{3.11}$$

with the augmented state vector $\hat{\mathbf{x}} = [\mathbf{x}, \mathbf{u}]^T \in \mathbb{R}^{\hat{n}}$, $\hat{n} = n + m$, and the matrices

$$\hat{\mathbf{A}}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{0} & -\mathbf{T} \end{bmatrix}, \tag{3.12}$$

where $\hat{\mathbf{A}}_i \in \mathbb{R}^{\hat{n}\times\hat{n}}$, $\hat{\mathbf{B}} \in \mathbb{R}^{\hat{n}\times m}$, $\mathbf{K} \in \mathbb{R}^{m\times n}$ and $\mathbf{I} \in \mathbb{R}^{m\times m}$ denotes the identity matrix. A T-S controller (similar to (2.16)) for the augmented T-S model (3.11) is given by

$$\mathbf{v} = \sum_{j=1}^{r} h_j(\mathbf{z}_c)\left(\mathbf{F}_j\hat{\mathbf{x}}\right) \tag{3.13}$$

with the feedback matrices $\mathbf{F}_j \in \mathbb{R}^{m\times\hat{n}}$.

In the following, we formulate a polytopic representation of the nested saturation in (3.10) (as shown in Section 2.2.1 for amplitude saturation) in order to represent it in form of LMI constraints. To this end, we introduce two identical sets $\mathcal{V}_1 = \mathcal{V}_2$. Their elements are the possible $m \times m$ boolean diagonal matrices, which lead to the cardinality of $|\mathcal{V}_1| = |\mathcal{V}_2| = 2^m$. From these sets we define the matrix tuples $\Phi_s = (\mathbf{E}_{s,1}, \mathbf{E}_{s,2}, \mathbf{E}_s^-)$ with

$$\mathbf{E}_{s,1} \in \mathcal{V}_1, \quad \mathbf{E}_{s,1} \neq \mathbf{E}_{s,2} \in \mathcal{V}_2, \quad \mathbf{E}_s^- = \mathbf{I} - [\mathbf{E}_{s,1} + \mathbf{E}_{s,2}] \,. \tag{3.14}$$

Hence, there exist $s = 3^m$ different matrix tuples Φ_s, $s \in \mathbb{N}_{1:3^m}$, which form the elements of a final set $\mathcal{V}$. Based on the elements in $\mathcal{V}$, we state the following Lemma for computing the closed-loop dynamics in from of a polytopic model:

Lemma 3.2.1. *Let a T-S controller* (3.13) *for a SE-NL T-S systems* (3.11) *(or a LO T-S model without affine terms) with an amplitude and rate saturating input* (3.11) *and two auxiliary T-S controllers with the feedback matrices* $\mathbf{H}_j$, $\mathbf{J}_j \in \mathbb{R}^{m \times \hat{n}}$, $j \in \mathbb{N}_{1:r}$, *be given. Then it follows that*

$$\boldsymbol{\sigma}\left(\mathbf{K}\hat{\mathbf{x}} + \mathbf{T}\boldsymbol{\sigma}\left(\sum_{j=1}^{r} h_j\left(\mathbf{F}_j\hat{\mathbf{x}}\right)\right)\right) \in \mathrm{co}\left\{\sum_{j=1}^{r} h_j\Big(\mathbf{E}_{s,1}\left(\mathbf{K}+\right.\right.$$
$$\left.\left. +\mathbf{T}\mathbf{F}_j\right) + \mathbf{E}_{s,2}\left(\mathbf{K} + \mathbf{T}\mathbf{H}_j\right) + \mathbf{E}_s^-\mathbf{J}_j\Big)\hat{\mathbf{x}} : \; s \in \mathbb{N}_{1:3^m}\right\} \tag{3.15}$$

and as a consequence, the closed-loop dynamics can be computed from the polytopic model

$$\dot{\hat{\mathbf{x}}} = \sum_{i=1}^{r}\sum_{j=1}^{r}\sum_{s=1}^{3^m} h_i(\mathbf{z}_s)h_j(\mathbf{z}_c)\mu_s\left(\hat{\mathbf{A}}_i + \hat{\mathbf{B}}\boldsymbol{\Delta}\right)\hat{\mathbf{x}}, \tag{3.16}$$

with

$$\boldsymbol{\Delta} = \mathbf{E}_{s,1}\left(\mathbf{K} + \mathbf{T}\mathbf{F}_j\right) + \mathbf{E}_{s,2}\left(\mathbf{K} + \mathbf{T}\mathbf{H}_j\right) + \mathbf{E}_s^-\mathbf{J}_j \tag{3.17}$$

and $\sum_{s=1}^{3^m} \mu_s = 1$, $0 \geq \mu_s \geq 1$ *for all* $\hat{\mathbf{x}}$ *inside the polytope*

$$\mathcal{L}(\tilde{\mathbf{H}}) = \bigcap_{j=1}^{r}\left(\mathcal{L}(\mathbf{H}_j)\right) \cap \bigcap_{j=1}^{r}\left(\mathcal{L}(\mathbf{J}_j)\right), \tag{3.18}$$

where each subpolytope $\mathcal{L}(\mathbf{H}_j)$, $\mathcal{L}(\mathbf{J}_j)$ *are defined according to* (2.40).

Proof: Applying Lemma 2.2.2 two times subsequently (once for each saturation function on the left-hand side of (3.15)) leads to the convex hull according to the right-hand side of the equation. For each application of Lemma 2.2.2 auxiliary feedback matrices are required which are $\mathbf{J}_j$ and $\mathbf{H}_j$. In other words, if $\mathbf{x} \in \mathcal{L}(\tilde{\mathbf{H}})$ then (3.15) is fulfilled and (3.16) follows which concludes the proof. ■

Based on (3.16) the following novel set invariance condition can be stated:

Theorem 3.2.2. *The ellipsoid $\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}})$ around the origin is a contractively invariant set of a closed-loop T-S system* (3.11), (3.13) *if there exist matrices $\mathbf{H}_j$, $\mathbf{J}_j$ such that $\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}) \subset \mathcal{L}(\tilde{\mathbf{H}})$, and*

$$\left(\hat{\mathbf{A}}_i + \hat{\mathbf{B}}\boldsymbol{\Delta}\right)^T \mathbf{P} + \mathbf{P}\left(\hat{\mathbf{A}}_i + \hat{\mathbf{B}}\boldsymbol{\Delta}\right) < \mathbf{0} \tag{3.19}$$

holds for all $i, j \in \mathbb{N}_{1:r}$ and $s \in \mathbb{N}_{1:3^m}$.

Proof: Consider the Lyapunov function $V_{\mathbf{0}} = \hat{\mathbf{x}}^T \mathbf{P} \hat{\mathbf{x}}$ at $\hat{\mathbf{x}}^* = \mathbf{0}$. Then

$$\dot{V}_{\mathbf{0}} = \hat{\mathbf{x}}^T \left[\sum_{s=1}^{3^m} \sum_{i=1}^{r} \sum_{j=1}^{r} \eta_s h_i(\mathbf{z}_s) h_j(\mathbf{z}_c) \left(\left(\hat{\mathbf{A}}_i + \hat{\mathbf{B}}\boldsymbol{\Delta}\right)^T \mathbf{P} + \mathbf{P}\left(\hat{\mathbf{A}}_i + \hat{\mathbf{B}}\boldsymbol{\Delta}\right) \right) \right] \hat{\mathbf{x}} \tag{3.20}$$

for all $\hat{\mathbf{x}} \in \mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}) \subset \mathcal{L}(\tilde{\mathbf{H}})$ according to Lemma 3.2.1. As η_s, h_i and h_j fulfill the convex sum property, $\dot{V}_{\mathbf{0}} < 0$ if (3.19) is satisfied which concludes the proof. ■

Based on that, we are able to state the following LMI conditions for estimating an over-saturating DA:

$$\mathbf{Q}\left(\hat{\mathbf{A}}_i + \hat{\mathbf{B}}\boldsymbol{\Delta}\right)^T + \left(\hat{\mathbf{A}}_i + \hat{\mathbf{B}}\boldsymbol{\Delta}\right)\mathbf{Q} < \mathbf{0}, \quad \forall\, i, j \in \mathbb{N}_{1:r}, s \in \mathbb{N}_{1:3^m}, \tag{3.21a}$$

$$\begin{bmatrix} u_{max,l}^2 & \tilde{\mathbf{h}}_{j,l}^T \\ \tilde{\mathbf{h}}_{j,l} & \mathbf{Q} \end{bmatrix} \geq \mathbf{0}, \quad \forall\, l \in \mathbb{N}_{1:m}, j \in \mathbb{N}_{1:r}, \tag{3.21b}$$

$$\begin{bmatrix} \dot{u}_{max,l}^2 & \tilde{\mathbf{j}}_{j,l}^T \\ \tilde{\mathbf{j}}_{j,l} & \mathbf{Q} \end{bmatrix} \geq \mathbf{0}, \quad \forall\, l \in \mathbb{N}_{1:m}, j \in \mathbb{N}_{1:r}, \tag{3.21c}$$

with $\mathbf{Q} = \left(\frac{\mathbf{P}}{\eta_{\mathbf{0}}}\right)^{-1}$, $\tilde{\mathbf{H}}_j = \mathbf{H}_j \mathbf{Q}$ ($\tilde{\mathbf{h}}_{l,j}^T$ denote the l-th row of the matrix $\tilde{\mathbf{H}}_j$) and $\tilde{\mathbf{J}}_j = \mathbf{J}_j \mathbf{Q}$ ($\tilde{\mathbf{j}}_{l,j}^T$ denote the l-th row of the matrix $\tilde{\mathbf{J}}_j$). From Proposition 2.2.3 follows that the constraints (3.21b) and (3.21c) ensure $\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}) \subset \mathcal{L}(\tilde{\mathbf{H}})$ which is required in Theorem 3.2.2. Multiplying the Lyapunov inequality (3.19) with $\mathbf{Q}$ from the left and the right-hand side results in condition (3.21a). The LMI conditions can be simply added to a convex optimization according to Section 2.2.2. Thereby, the matrices $\tilde{\mathbf{H}}_j$ and $\tilde{\mathbf{J}}_j$ should be considered as optimization variables, e.g. for finding the largest invariant set similar to (2.48).

Remark 3.2.2. Analogous to (2.36), the control gains $\mathbf{F}_j$ can be additionally considered as optimization parameters by substituting $\tilde{\mathbf{F}}_j = \mathbf{F}_j \mathbf{Q}$. Also a certain closed-loop performance can be simultaneously ensured, e.g. by claiming a required exponential decay rate $\alpha \geq 0$ for the Lyapunov function. Other possibilities are detailed in Section 2.2.2.

Remark 3.2.3. The input matrix $\hat{\mathbf{B}}$ of the extended T-S formulation (3.11) is in a form that is often required for the estimation of a DA for T-S systems with affine terms [67]. The corresponding LMI conditions are based on the fact that the input is constant. However, often the T-S formulation is already assumed to be in the right from, but how to achieve that form is not answered. Hence, the extension with an actuator model is a noticeable way to transform each affine T-S models in their needed form.

Example 3.2.2 (Example 3.2.1 cont'd)**.** System (2.44) is extended by the saturating actuator dynamics (3.10) which leads to an augmented T-S model (3.11) that consists of the matrices

$$\hat{\mathbf{A}}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \underline{\theta}_1(z_s) & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{\mathbf{A}}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \overline{\theta}_1(z_s) & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{\mathbf{b}} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T, \quad \mathbf{K} = \begin{bmatrix} 0 & -\tau \end{bmatrix}^T. \tag{3.22}$$

The augmented state vector is $\hat{\mathbf{x}} = [\mathbf{x}, u]^T$ and $(\hat{\mathbf{A}}_i, \hat{\mathbf{B}})$, $i \in \mathbb{N}_{1:2}$. The UoD ($x_{1,max} = 1.6$) and the input saturation ($u_{max} = v_{max} = 7$) is set equal to Example 3.2.1. Additionally, the input rate is limited with $\dot{u}_{max} = 50$, $\tau = 50$ and the PDC (2.47) is extended by a zero element

$$\mathbf{f}_1^T = \begin{bmatrix} -2.46 & -4.36 & 0 \end{bmatrix}, \quad \mathbf{f}_2^T = \begin{bmatrix} -1.07 & -5.06 & 0 \end{bmatrix}. \tag{3.23}$$

The extended optimization problem is given by:

$$\begin{aligned} &\min_{\mathbf{Q}>\mathbf{0}, \tilde{\mathbf{h}}_j^T, \tilde{\mathbf{j}}_j^T} -\log\det(\mathbf{Q}) \qquad (3.24) \\ &(a)\ \mathbf{Q}\left(\hat{\mathbf{A}}_i + \hat{\mathbf{b}}\boldsymbol{\Delta}\right)^T + \left(\hat{\mathbf{A}}_i + \hat{\mathbf{b}}\boldsymbol{\Delta}\right)\mathbf{Q} < \mathbf{0}, \quad \forall\, i, j \in \mathbb{N}_{1:r}, s \in \mathbb{N}_{1:3}, \\ &(b)\ \begin{bmatrix} u_{max}^2 & \tilde{\mathbf{h}}_i^T \\ \tilde{\mathbf{h}}_i & \mathbf{Q} \end{bmatrix} \geq \mathbf{0}, \quad i \in \mathbb{N}_{1:2} \\ &(c)\ \begin{bmatrix} \dot{u}_{max}^2 & \tilde{\mathbf{j}}_l^T \\ \tilde{\mathbf{j}}_l & \mathbf{Q} \end{bmatrix} \geq \mathbf{0},\ l \in \mathbb{N}_{1:m}, j \in \mathbb{N}_{1:r}, \\ &(d)\ [1,0,0]\mathbf{Q}[1,0,0]^T \leq (1.6)^2\,. \end{aligned}$$

Fig. 3.3 depicts the finally estimated three-dimensional DA. It can be seen that asymptotically stable regions with effective saturation $|\dot{u}| > \dot{u}_{max}$ can be realized. This is also

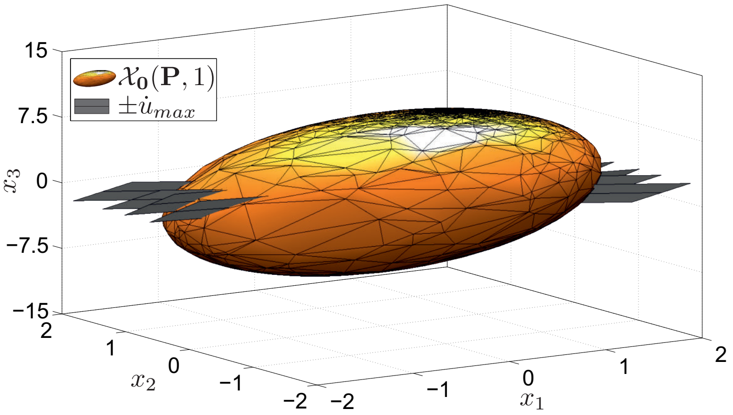


Figure 3.3: Estimation $\mathcal{X}_0(\mathbf{P},1)$ subject to the UoD, input amplitude and rate limits.

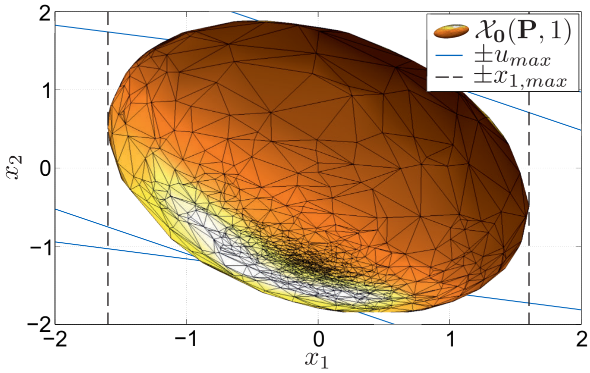


Figure 3.4: Topview of Fig. 3.3.

the case concerning input amplitude saturation $|u| > u_{max}$ which is shown in Fig. 3.4. Here, the UoD is marked by $\pm x_{1,max}$ in addition. Consequently, the over-saturating DA lies within the UoD such that it is directly valid for the original nonlinear system.

3.2.3 Relaxation of the Set Invariance Condition

Based on the LMI conditions (3.21) a DA can be estimated that avoids saturation to occur, meaning that the linear region of the actuator dynamics (3.10) is not left. We formulate the following invariance condition which relaxes the required number of LMI conditions but at the cost of the size of the estimated DA:

Proposition 3.2.1. *The ellipsoid $\mathcal{X}_\mathbf{0}(\mathbf{P},\eta_\mathbf{0})$ around the origin is a contractively invariant set of a closed-loop T-S system* (3.11), (3.13) *if $\mathcal{X}_\mathbf{0}(\mathbf{P},\eta_\mathbf{0}) \subset \bigcap_{j=1}^{r} \mathcal{L}(\mathbf{F}_j)$, where each subpolytope $\mathcal{L}(\mathbf{F}_j)$ is defined according to* (2.40)*, and* (3.19) *holds for all $i, j \in \mathbb{N}_{1:r}$ setting*

$$\mathbf{\Delta} = \mathbf{K} + \mathbf{T}\mathbf{F}_j. \tag{3.25}$$

Proof: Due to $\mathcal{X}_{\mathbf{0}}(\mathbf{P},\eta_{\mathbf{0}}) \subset \bigcap_{j=1}^{r} \mathcal{L}(\mathbf{F}_j)$ the linear region is not left and thus the polytopic modeling (3.15) of the nested saturation function (3.10) can be omitted. Following the proof of Theorem 3.2.2 based on that leads to the LMI (3.21) with (3.25). The conditions (3.21b) and (3.21c) are thereby depending on $\mathbf{F}_j$ (instead of the axillary matrices $\mathbf{H}_j$ and $\mathbf{J}_j$). In other words, $\tilde{\mathbf{H}}_j$ has to be replaced by $\mathbf{F}_j$ and $\tilde{\mathbf{J}}_j$ by $\mathbf{K}+\mathbf{TF}_j$. Thereby, the proof is concluded. ■

Proposition 3.2.1 reduces the number of LMIs hugely (3^m less LMI conditions) whereby the computational solvability increases. However, this to the disadvantage of the DA size. Motivated by that, we bypass this disadvantage in the following. We exploit the structure of the augmented T-S formulation (3.11) in order to reduce the number of LMIs without simultaneously increasing the conservatism of the solution. More precisely, we eliminate all LMIs in Theorem 2.2.3 with $j \in \mathbb{N}_{1:r}$:

Theorem 3.2.3. *The ellipsoid $\mathcal{X}_{\mathbf{0}}(\mathbf{P},\eta_{\mathbf{0}})$ around the origin is a contractively invariant set of a closed-loop T-S system* (3.11), (3.13) *if a PDC controller (see Remark 2.1.1) is considered and if there exist matrices $\mathbf{H}_j$, $\mathbf{J}_j$ such that $\mathcal{X}_{\mathbf{0}}(\mathbf{P},\eta_{\mathbf{0}}) \subset \mathcal{L}(\tilde{\mathbf{H}})$ as well as*

$$(\mathbf{A}_i+\mathbf{B\Delta})^T\mathbf{P}+\mathbf{P}\left(\mathbf{A}_i+\mathbf{B\Delta}\right)<\mathbf{0}, \tag{3.26}$$

with

$$\mathbf{\Delta}=\mathbf{E}_{s,1}\left(\mathbf{K}+\mathbf{TF}_i\right)+\mathbf{E}_{s,2}\left(\mathbf{K}+\mathbf{TH}_i\right)+\mathbf{E}_s^{-}\mathbf{J}_i \tag{3.27}$$

holds for all $i \in \mathbb{N}_{1:r}$ and $s \in \mathbb{N}_{1:3^m}$.

Proof: All linear subsystems $(\hat{\mathbf{A}}_i,\hat{\mathbf{B}})$, $i \in \mathbb{N}_{1:r}$ of the augmented T-S formulation (3.11) have a common input matrix $\hat{\mathbf{B}}_i = \hat{\mathbf{B}}$. Following Theorem 2.2.3, equation (3.16) can be formulated as

$$\dot{\hat{\mathbf{x}}}=\sum_{i=1}^{r}\sum_{s=1}^{3^m}h_i(\mathbf{z}_s)\mu_s\left(\hat{\mathbf{A}}_i+\hat{\mathbf{B}}\mathbf{\Delta}\right)\hat{\mathbf{x}}, \tag{3.28}$$

with $\mathbf{\Delta}$ according to (3.27). Based on that, we follow the proof of Theorem (3.2.2) which results in the time derivative of the Lyapunov function

$$\dot{V}_{\mathbf{0}}=\hat{\mathbf{x}}^T\left[\sum_{s=1}^{3^m}\sum_{i=1}^{r}\eta_s h_i(\mathbf{z}_s)\left((\mathbf{A}_i+\mathbf{B\Delta})^T\mathbf{P}+\mathbf{P}\left(\mathbf{A}_i+\mathbf{B\Delta}\right)\right)\right]\hat{\mathbf{x}}. \tag{3.29}$$

As η_s and $h_i(\mathbf{z}_s)$ fulfill the convex sum property, $\dot{V}_{\mathbf{0}} < 0$ if (3.26) is satisfied which concludes the proof. ■

Beside the required number of LMIs, the polytopic nature of the T-S formulation of a nonlinear system itself causes that the estimated DA is rather a conservative approximation of the original region of attraction. From that point of view it is clearly evident that the shape of the DA, meaning its form and size, essentially depends on the specified universe of discourse (UoD) of the T-S model. Moreover, it is probable that a level value $\tilde{\eta}_{\mathbf{0}} > \eta_{\mathbf{0}}$ still bounds a contractively invariant set of the nonlinear system with

$$\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}) \subset \mathcal{X}_{\mathbf{0}}(\mathbf{P}, \tilde{\eta}_{\mathbf{0}}), \tag{3.30}$$

even if the corresponding bounding level set $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \tilde{\eta}_{\mathbf{0}})$ crosses the UoD of the T-S model. Motivated by that, we propose a novel procedure for tackling both of the mentioned drawbacks which is summarized in Algorithm 3.2.1. The algorithm is subdivided in two parts: In the first part (line 1 and 6), the DA is estimated whereby the UoD of the T-S model can be considered as optimization variable. More precisely, the level value $\eta_{\mathbf{0}} = c$ and allowed upper values of the UoD (3.1), (3.3)

$$\begin{aligned} &0 < x_{i,max} \leq \tilde{x}_i, \quad x_i^- = -x_{i,max},\ x_i^+ = x_{i,max}, \\ &0 < u_{i,max} \leq \tilde{u}_i, \quad u_i^- = -u_{i,max},\ u_i^+ = u_{i,max} \end{aligned} \tag{3.31}$$

are first defined (line 2 and 3). These bounds might be given by physical state limitations of the system at hand, e.g. elevator and aileron deflection in aircraft systems or the maximum thermal energy in a chemical process. If no such limitations exit they can be meaningfully chosen. In the subsequent optimization (line 4 to 6) the objective is in maximizing the volume of $\mathcal{X}_{\mathbf{0}}(\mathbf{P}, c)$ based on convex optimization, e.g. based on (3.24),

Algorithm 3.2.1 Reducing the conservatism of the estimated domain of attraction

```
 1: UoD optimization:
 2: define $\eta_{\mathbf{0}} = c > 0$
 3: define $\tilde{x}_i > 0,\ \tilde{u}_i > 0,\ \forall\ x_i \in \mathcal{M}_{s,x}, \mathcal{M}_{c,x},\ \forall\ u_i \in \mathcal{M}_{s,u}$
 4: $\min_{(x_{i,max}, u_{i,max}) \in \mathcal{M}_{s,x}, \mathcal{M}_{c,x}, \mathcal{M}_{s,u}} -\det(\mathbf{Q})$
 5:    s.t. $0 < x_{i,max} \leq \tilde{x}_i,\ 0 < u_{i,max} \leq \tilde{u}_i$
 6:          DA optimization, e.g. by (3.24) and Remark 3.2.1, 3.2.2,

 7: Enlarging beyond UoD:
 8: define step size $\delta\eta << 1,\ \tilde{\eta}_{\mathbf{0}} = \eta_{\mathbf{0}}$
 9: while ($2\mathbf{x}_j^T \mathbf{P} \dot{\mathbf{x}}_j < 0$ for (2.6) with (3.10) and (3.13) $\forall\ \mathbf{x}_j \in \partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \tilde{\eta}_{\mathbf{0}}),\ j >> 1$)
    and (LMI (3.5a) is satisfied for all state constraints with $\mathbf{Q} = (\mathbf{P}/\tilde{\eta}_{\mathbf{0}})^{-1}$) do
10:    $\tilde{\eta}_{\mathbf{0}} = \tilde{\eta}_{\mathbf{0}} + \delta\eta$
11: $\tilde{\eta}_{\mathbf{0}} = \tilde{\eta}_{\mathbf{0}} - \delta\eta$
```

Table 3.1: Domain of attraction by applying Algorithm 3.2.1

description	UoD $x_{1,max}$	level value	det $(\mathbf{Q})$
example 3.2.2	1.6	1	580.33
line 1 till 6 of Algorithm 3.2.1	1.27	1	836.50
after Algorithm 3.2.1	1.27	1.28	1754.30

considering the UoD of the T-S formulation as additional optimization parameter. For instance, a genetic algorithm (for the UoD) with an embedded convex optimization (for the size of the DA) can be considered for implementing that optimization problem.

Starting point of the second part of the algorithm (lines 7 till 11) is the optimized DA $\mathcal{X}_{\mathbf{0}}(\mathbf{P}, c)$ which is the final outcome of the first part. Now, we attempt to find a larger DA $\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \tilde{\eta}_{\mathbf{0}})$ according to (3.30) by iteratively enlarging the critical level value as long as the time derivative of the Lyapunov function is negative for the original closed-loop system and LMI (3.5a) for the physical (not the UoD) state constraints. Therefore, starting from $\tilde{\eta}_{\mathbf{0}} = \eta_{\mathbf{0}} = c$ the value of $\tilde{\eta}_{\mathbf{0}}$ is increased by the step size $\delta\eta << 1$ until $\dot{V}_{\mathbf{0}} \geq 0$ for at least one $\mathbf{x} \in \mathcal{X}_{\mathbf{0}}(\mathbf{P}, \tilde{\eta}_{\mathbf{0}})$ or a violation of LMI (3.5a) has been detected. For a numerical realization, the actual bounding level set $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \tilde{\eta}_{\mathbf{0}})$ is discretized with $\mathbf{x}_j$, $j >> 1$. As soon as $\dot{V} \geq 0$ has been detected for a $\mathbf{x}_j \in \partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \tilde{\eta}_{\mathbf{0}})$, the actual level set does not bound a DA for the nonlinear system. Hence, the level value $\tilde{\eta}_{\mathbf{0}}$ is finally set to its value from the prior iteration step (line 11) such that a valid DA is obtained.

Example 3.2.3 (Example 3.2.2 cont'd)**.** The UoD of $x_{1,max} = 1.6$ results in a DA with the volume measure of $\det(\mathbf{Q}) = 580.33$. Table 3.1 compares this with the optimization according to Algorithm 3.2.1. After the UoD optimization (line 1 till 6) the volume measure has increased by 44 %. The subsequent expansion of the level value (Algorithm 3.2.1 (lines 7 till 11)) results in $\det(\mathbf{Q}) = 1754.30$ which corespondents to an additional enlargement of the volume measure by 110 %. Consequently, the reduction of the conservatism of the estimated DA becomes obvious.

3.3 Summary

In this chapter, we were concerned with the question how a closed-loop T-S notation can be exploited to estimate an as large as possible ellipsoidal domain of attraction (DA) of a constrained nonlinear system. We have developed set invariance conditions for determining the critical level value of the DA subject to state, input amplitude and rate constraints. The main contribution of this chapter has been a numerical procedure to effectively compute a preferably large estimation of the DA of a system's equilibrium. Thereby, the DA has been first estimated based on convex optimization and thereafter the bounding level value has been iteratively enlarged.

Chapter 4

Switched Controller Design

Generally, the size of an estimated domain of attraction (DA) of a closed-loop system is at the expense of the achievable control performance. For instance, an aggressive control law reaches input limitations in regions of the state space where a low performing control law is still far from saturation. That trade-off is relaxed in this chapter by designing switched controllers. The considered problems are stated in Section 4.1. In the Sections 4.2 and 4.3 we derive switching control laws based on nested and non-nested DAs. Thereby, we represent a dynamical system by multiple T-S models, each having a individual universe of discourse (UoD), controller and estimated DA. We develop set invariance conditions that allow an estimation of nested DAs based on convex optimization.

4.1 Problem Formulation

A desired control performance can be claimed while estimating a DA according to the previous chapter (see Remark 3.2.2). However, this obviously influences the size of the estimated DA whereby a trade-off occurs that needs to be balanced. Hence, the question to be answered is whether a highly performant controller or a large invariant set is more important. However, an alternative question might be: Is there an opportunity to gather both advantages at the same time? Exactly this is the problem that we address in this chapter:

Problem 4.1.1. Estimating a large DA for a constrained system (state, input amplitude and rate constraints) while simultaneously guaranteeing a performant behavior of the closed-loop system by exploiting the T-S framework.

We tackle Problem 4.1.1 by a novel LMI condition for estimating nested domains of attraction (DAs) for T-S systems. The condition is integrated in Algorithm 3.2.1 such that a defined amount of nested invariant sets can be estimated in a numerical efficient way. Each set is based on an individual universe of discourse (UoD). As the sets become smaller, the performance of the corresponding closed-loop system can be increased. In order to benefit from the variety of the estimates (large DA and a fast tracking) we introduce asymptotically stable switching conditions. We generalize the switching condition such that asymptotically stable tracking of a set point is guaranteed even if the DAs are not nested.

4.2 Nested Control Architecture

The objective of this section is to extend Algorithm 3.2.1 in order to effectively estimate nested invariant sets. The extension of the algorithm is introduced in Section 4.2.1. Based on that, we derive a switched T-S controller based on multiple T-S models in Section 4.2.2.

4.2.1 Nested Invariant Sets

We extend our Algorithm 3.2.1 such that $\zeta \in \mathbb{N}$, $\zeta > 1$, nested invariant sets

$$\mathcal{X}_{\mathbf{0}}(\mathbf{P}_1, \eta_{\mathbf{0},1}) \subset \mathcal{X}_{\mathbf{0}}(\mathbf{P}_2, \eta_{\mathbf{0},2}) \ldots \subset \mathcal{X}_{\mathbf{0}}(\mathbf{P}_\zeta, \eta_{\mathbf{0},\zeta}) \tag{4.1}$$

with optimized volumes can be estimated. Each set has an individual bounding level value $\eta_{\mathbf{0},q}$ and Lyapunov matrix $\mathbf{P}_q$, $q \in \mathbb{N}_{1:\zeta}$. To this end, we derive the following LMI condition:

Lemma 4.2.1. *The DAs $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_1, \eta_{\mathbf{0},1})$ and $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_2, \eta_{\mathbf{0},2})$ are nested with $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_1, \eta_{\mathbf{0},1}) \subset \mathcal{X}_{\mathbf{0}}(\mathbf{P}_2, \eta_{\mathbf{0},2})$ if*

$$\mathbf{Q}_1 < \mathbf{Q}_2 \tag{4.2}$$

with $\mathbf{Q}_i = \left(\frac{\mathbf{P}_i}{\eta_{\mathbf{0},i}}\right)^{-1}$, $i \in \mathbb{N}_{1:2}$.

Proof: Two ellipsoids are nested $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_1, \eta_{\mathbf{0},1}) \subset \mathcal{X}_{\mathbf{0}}(\mathbf{P}_2, \eta_{\mathbf{0},2})$ if $\frac{\mathbf{P}_1}{\eta_{\mathbf{0},1}} > \frac{\mathbf{P}_2}{\eta_{\mathbf{0},2}}$ holds true. Hence,

$$\mathbf{Q}_1 = \left(\frac{\mathbf{P}_2}{\eta_{\mathbf{0},2}} + \mathbf{M}\right)^{-1} \tag{4.3}$$

where $\mathbf{M} > \mathbf{0}$. According to the Woodbury identity [94], (4.3) is written as

$$\mathbf{Q}_1 = \mathbf{Q}_2 - \underbrace{\mathbf{Q}_2 \left(\mathbf{M}^{-1} + \mathbf{Q}_2\right)^{-1} \mathbf{Q}_2}_{\mathbf{L} > \mathbf{0}} . \tag{4.4}$$

Thus, $\mathbf{Q}_1 = \mathbf{Q}_2 - \mathbf{L}$ leads to $\mathbf{Q}_2 > \mathbf{Q}_1$ which concludes the proof. ■

For estimating nested DAs, we introduce ζ SE-NL T-S models (or LO T-S models without affine terms) of the original nonlinear system with nested universes of discourse

$$\mathcal{S}_{\mathbf{x},1} \subset \mathcal{S}_{\mathbf{x},2} \subset \ldots \subset \mathcal{S}_{\mathbf{x},\zeta}, \tag{4.5a}$$

$$\mathcal{S}_{\mathbf{u},1} \subset \mathcal{S}_{\mathbf{u},2} \subset \ldots \subset \mathcal{S}_{\mathbf{u},\zeta}. \tag{4.5b}$$

Fig. 4.1 exemplarily sketches that for $\zeta = 3$ concerning a system $\mathbf{x} \in \mathbb{R}^2$ with one nonlinearity in the first state variable x_1 (like in Example 2.2.3). Depicted are the bounding level sets $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_q, \eta_{\mathbf{0},q})$, $q \in \mathbb{N}_{1:3}$, and the nested universes of discourse. The simple step of introducing nested T-S models delivers two advantages:

1. The state and input variables which are part of the premise variables are nested by definition. Hence, the numerical optimization of nested DAs is eased.

2. Due to the nesting of the DAs, they become smaller from $\mathcal{S}_\zeta$ towards $\mathcal{S}_1$ and thus the control performance can be increased successively. Consequently, Problem 4.1.1 is solved.

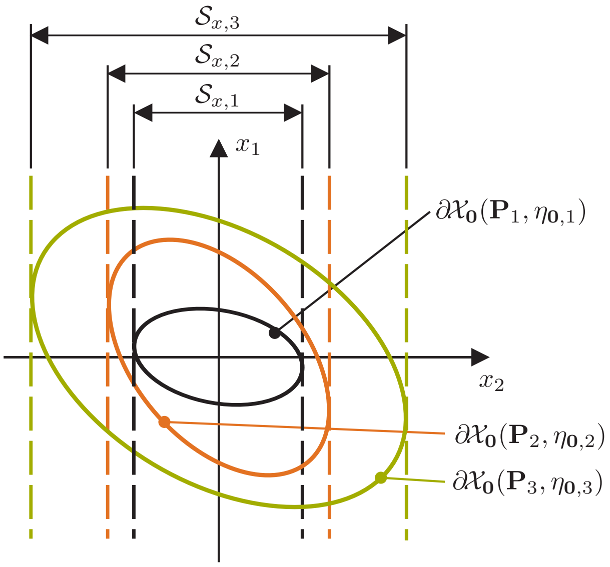

$\mathcal{S}_{x,3}$
$\mathcal{S}_{x,2}$
$\mathcal{S}_{x,1}$
x_1
$\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_1, \eta_{\mathbf{0},1})$
x_2
$\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_2, \eta_{\mathbf{0},2})$
$\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_3, \eta_{\mathbf{0},3})$

Figure 4.1: Nested Lyapunov functions for T-S systems.

Algorithm 4.2.1 Estimating optimized nested DAs

```
1:  define number of desired nested sets $\zeta > 1$, $\zeta \in \mathbb{N}$
2:  define $\tilde{x}_i > 0$, $\tilde{u}_i > 0$, $\forall\, x_i \in \mathcal{M}_{s,x}, \mathcal{M}_{c,x}$, $\forall\, u_i \in \mathcal{M}_{s,u}$
3:  for q = $\zeta$ do
4:    apply Algorithm 3.2.1
5:  UoD optimization:
6:  for q = $\zeta$-1:-1:1 do
7:    define $\eta_{\mathbf{0},q} = c > 0$
8:    $\min_{(x_{i,max,q}, u_{i,max,q}) \in \mathcal{M}_{s,x}, \mathcal{M}_{c,x}, \mathcal{M}_{s,u}}$ $-\det\left(\mathbf{Q}_q\right)$
9:      s.t. $0 < x_{i,max,q} \leq \tilde{x}_i$, $0 < u_{i,max,q} \leq \tilde{u}_i$, $x_{i,max,q} \leq x_{i,max,q+1}$, $u_{i,max,q} \leq u_{i,max,q+1}$,
10:          $\mathbf{Q}_{q+1} < \mathbf{Q}_q$ (Lemma 4.2.1)
11:          DA optimization, e.g. by (3.24) and Remark 3.2.1, 3.2.2,

12:   Enlarging beyond UoD:
13:   define step size $\delta\eta << 1$, $\tilde{\eta}_{\mathbf{0},q} = \eta_{\mathbf{0}}$
14:   while ($2\mathbf{x}_j^T \mathbf{P}_q \dot{\mathbf{x}}_j < 0$ for (2.6) with (3.10) and (3.13) $\forall\, \mathbf{x}_j \in \partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_q, \tilde{\eta}_{\mathbf{0}})$, $j >> 1$)
      and (LMI (3.5a) is satisfied for all state constraints with $\mathbf{Q}_q = (\mathbf{P}_q/\tilde{\eta}_{\mathbf{0}})^{-1}$) and
      $\frac{P_{q+1}}{\tilde{\eta}_{\mathbf{0},q+1}} < \frac{P_q}{\eta_{\mathbf{0},q}}$ do
15:     $\tilde{\eta} = \tilde{\eta}_q + \delta\eta$
16:   $\tilde{\eta}_q = \tilde{\eta}_q - \delta\eta$
```

A constructive numerical procedure for estimating nested DAs is summarized in Algorithm 4.2.1: The procedure is based on Algorithm 3.2.1. For the largest of the nested sets $q = \zeta$ Algorithm 3.2.1 is directly considered. For the remaining set (starting on line 6) the optimization is sequentially applied from $q = \zeta - 1$ towards $q = 1$. Compared to Algorithm 3.2.1 the constraints $x_{i,max,q} \leq x_{i,max,q+1}$, $u_{i,max,q} \leq u_{i,max,q+1}$ and Lemma 4.2.1 are added which ensure that the UoD and the invariant sets are nested according to (4.5) and (4.1), respectively. In the second part of the algorithm (line 12) new constraints $\frac{P_q}{\tilde{\eta}_{\mathbf{0},q}} > \frac{P_{q+1}}{\tilde{\eta}_{\mathbf{0},q+1}}$ are required in line 14 to ensure that the DAs remain nested while iteratively increasing their bounding level values.

Example 4.2.1 (Example 3.2.3 cont'd)**.** We are interested in designing $\zeta = 3$ nested invariant sets for the system (3.22). The largest set $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_3, \tilde{\eta}_{\mathbf{0},3})$ is equivalent to the one optimized in Example 3.2.3, i.e. the PDC equals to (3.23), a UoD of $x_{1,max} = 1.27$, a zero decay rate $\alpha_3 = 0$ and a critical level value of $\tilde{\eta}_{\mathbf{0},3} = 1.28$. For the remaining two sets $q \in \mathbb{N}_{1:2}$, we set the desired decay rates of the Lyapunov function (performance indicator of the closed-loop systems) to $\alpha_2 = 1.5$ and $\alpha_1 = 3.0$. The PDC controller gains and the UoDs are considered as optimization variables. Table 4.1 summarizes the obtained results according to Algorithm 4.2.1. The optimized DAs are nested as $\min\left(\text{eig}\left(\frac{P_q}{\tilde{\eta}_{\mathbf{0},q}} > \frac{P_{q+1}}{\tilde{\eta}_{\mathbf{0},q+1}}\right)\right) > 0$ for $q \in \mathbb{N}_{1:2}$ (see also Lemma 4.2.1).

Table 4.1: Domain of attraction by applying Algorithm 4.2.1

q	decay rate	UoD $x_{1,max}$	level value	det $(\mathbf{Q})$	$\min\left(\mathrm{eig}\left(\frac{P_q}{\tilde{\eta}_{\mathbf{0},q}} > \frac{P_{q+1}}{\tilde{\eta}_{\mathbf{0},q+1}}\right)\right)$
3	0	1.27	1.28	1754.30	–
2	1.5	1.09	1.23	543.55	$9.93 \cdot 10^{-6}$
1	3	0.81	1.38	181.26	$8.84 \cdot 10^{-6}$

4.2.2 Controller Design

In the prior section, we have introduced a method for constructively designing nested domains of attraction (DAs) for a nonlinear system by optimizing nested closed-loop T-S models. In this section, we focus on designing an asymptotically stable condition for switching between the obtained T-S controllers

$$\mathbf{u} = \mathbf{u}_l = \sum_{i=1}^{r} h_j(\mathbf{z}_c)\left(\mathbf{F}_{j,l}\mathbf{x}\right),\ l \in \mathbb{N}_{1:y}, \tag{4.6}$$

where l denotes the active DA (4.1). The switching strategy is a natural extension of the nested controller design for linear systems [5]:

Theorem 4.2.1. *Let $\zeta > 1$, $\zeta \in \mathbb{N}$, closed-loop SE-NL T-S models (or LO T-S models without affine terms) for a dynamical system be given with nested DAs* (4.1) *for $\mathbf{x}^* = \mathbf{0}$. Then switching between the corresponding T-S controllers* (4.6) *asymptotically stabilizes the origin for all $\mathbf{x} \in \mathcal{X}_{\mathbf{0}}(\mathbf{P}_\zeta, \eta_{\mathbf{0},\zeta})$ if the switching signal is*

$$l = \arg\max_{q}\left(\mathbf{x}^T\mathbf{P}_q\mathbf{x}\right)\ s.t.\ \mathbf{x}^T\mathbf{P}_q\mathbf{x} \leq \eta_{\mathbf{0},q}, \quad q \in \mathbb{N}_{1:\zeta}. \tag{4.7}$$

Proof: The switching condition (4.7) activates a T-S controller l as soon as the trajectory runs into the corresponding DA $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_l, \eta_{\mathbf{0},l})$, $l \in \mathbb{N}_{1:\zeta}$. The maximum level value of an active DA is $V_{\mathbf{0},l,max} = \eta_{\mathbf{0},l}$ and its derivative is negative $\dot{V}_{\mathbf{0},l} < 0$. Hence, the switching is only scheduled in one direction $l : \zeta \to 1$ and the maximum possible level value of each previously activated Lyapunov function decreases after each switching, e.g. if switching from $l+1 \to l \to l-1$ means that

$$V_{\mathbf{0},l+1,max} = \eta_{\mathbf{0},l+1} \to V_{\mathbf{0},l+1,max} = \hat{\eta}_{\mathbf{0},l+1} < \eta_{\mathbf{0},l+1} \to V_{\mathbf{0},l+1,max} < \hat{\eta}_{\mathbf{0},l+1}, \tag{4.8a}$$

$$V_{\mathbf{0},l,max} = \eta_{\mathbf{0},l} \to V_{\mathbf{0},l,max} < \eta_{\mathbf{0},l}. \tag{4.8b}$$

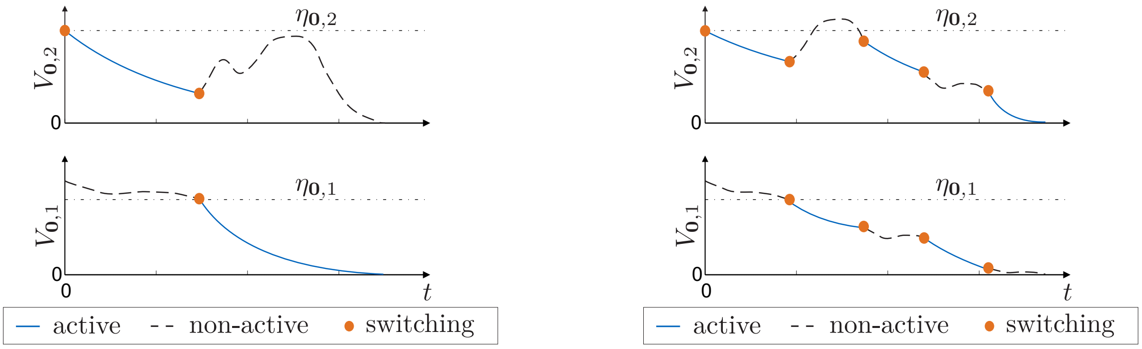
After the final switching $l = 1$, the active Lyapunov function $V_{\mathbf{0},1}$ will decrease to zero whereby all previously activated Lyapunov functions will become zero, too (due to their quadratic nature). Consequently, $\mathbf{x}^* = \mathbf{0}$ is asymptotically stabilized and the proof concluded. ■

The quintessence of the proof is sketched for clarification in Fig. 4.2(a) concerning two nested DAs $\mathcal{X}_\mathbf{0}(\mathbf{P}_1, \eta_{\mathbf{0},1}) \subset \mathcal{X}_\mathbf{0}(\mathbf{P}_2, \eta_{\mathbf{0},2})$. The time intervals within which a certain Lyapunov function is active are drawn by solid lines and deactivated time spans are depicted by dashed lines. The outer DA and the corresponding T-S controller are active first assuming an initial state vector at the bounding level set, meaning $V_{\mathbf{0},2} = \eta_{\mathbf{0},2}$. The value of the corresponding Lyapunov function decreases continuously while the inactive Lyapunov function $V_{\mathbf{0},1}$ might increase. As soon as the trajectory runs into the region $\mathcal{X}_\mathbf{0}(\mathbf{P}_1, \eta_{\mathbf{0},1})$ the Lyapunov function $V_{\mathbf{0},1}$ becomes active and decreases for sure. Due to the nesting of the two DAs the value of the inactive Lyapunov function is according to (4.8) bounded by $V_{\mathbf{0},l,max} \leq \eta_{\mathbf{0},l}$. When $V_{\mathbf{0},1}$ reaches zero, $V_{\mathbf{0},2}$ becomes zero too.

Obviously, if the Lyapunov functions are equivalent with $V_{\mathbf{0},1} = V_{\mathbf{0},2} = \ldots = V_{\mathbf{0},\zeta}$, they will continuously decrease independent of the switching. Hence, we formulate the following proposition for the sake of completeness:

Proposition 4.2.1. *Let $\zeta > 1$, $\zeta \in \mathbb{N}$, closed-loop SE-NL T-S models (or LO T-S models without affine terms) for a dynamical system be given with nested DAs* (4.1) *for $\mathbf{x}^* = \mathbf{0}$. Then switching between the corresponding T-S controllers* (4.6) *asymptotically stabilizes the origin for all $\mathbf{x} \in \mathcal{X}_\mathbf{0}(\mathbf{P}_\zeta, \eta_{\mathbf{0},\zeta})$ for arbitrary switching (e.g. according to* (4.7)*) if $\mathbf{P} = \mathbf{P}_q$ for all $q \in \mathbb{N}_{1:\zeta}$.*

Proof: The asymptotic stability follows directly from the fact that switching between systems that have a common (quadratic) Lyapunov function behave asymptotically stable independent of the switching signal [65, 80]. ■



(a) Nested

(b) Non-nested

Figure 4.2: Stability condition for multiple Lyapunov functions.

4.3 Non-Nested Control Architecture

In this section, we generalize the nested T-S control architecture of the previous section to the non-nested case. More precisely, the universes of discourse (UoDs) of the $\zeta > 1$, $\zeta \in \mathbb{N}$, T-S models as well as the estimated domains of attraction (DAs) do not have to be nested. To this aim, the switching condition (4.7) is generalized:

Theorem 4.3.1. *Let $\zeta > 1$, $\zeta \in \mathbb{N}$, closed-loop SE-NL T-S models (or LO T-S models without affine terms) for a dynamical system, each with an estimated DAs for $\mathbf{x}^* = \mathbf{0}$, be given. Then switching between the corresponding T-S controllers* (4.6) *asymptotically stabilizes the origin for all $\mathbf{x}$ that fulfill $\mathbf{x} \in \mathcal{X}_{\mathbf{0}}(\mathbf{P}_l, \eta_{\mathbf{0}})$ for at least one $l \in \mathbb{N}_{1:\zeta}$ if the switching between the T-S controllers fulfills the following conditions:*

i) if a T-S controller l is active then $\mathbf{x} \in \mathcal{X}_{\mathbf{0}}(\mathbf{P}_l, \eta_{\mathbf{0},l})$ (e.g. switching law (4.7)*)*

ii) if a T-S controller l has been activated for the i-th time within the time interval $t \in [t_{a,i}, t_{e,i}]$ then a subsequent reactivation at a time $t = t_{a,i+1} > t_{e,i}$ is only allowed iff

$$V_{\mathbf{0},l}(t_{a,i+1}) < V_{\mathbf{0},l}(t_{a,i}). \tag{4.9}$$

Proof: Condition i) ensures that the state vector $\mathbf{x}$ is within the DA of the active controller. Hence, only controllers which ensure asymptotic stability are allowed to become active. The asymptotic stability is maintained for arbitrary switching by condition ii) in the sense of multi Lyapunov [19]: A deactivated Lyapunov function might increase but its reactivation is only allowed if its level value has decreased since its last activation. Consequently, the level value for activating a certain DA decreases continuously which concludes the proof. ■

Fig. 4.2(b) clarifies the conditions i) and ii) considering two Lyapunov functions $V_{\mathbf{0},1}$ and $V_{\mathbf{0},2}$ (analogous to Fig. 4.2(a)). Whenever a Lyapunov function is active its level value surely decreases (condition i)). Although the value of a deactivated Lyapunov function may increase, its reactivation value decreases continuously (condition ii)). Finally, if the active Lyapunov function reaches zero, all deactivated functions are zero as well.

4.4 Summary

Within this chapter, we have addressed the problem of estimating a large domain of attraction (DA) for a set point of a constrained system while simultaneously guaranteeing a performant behavior of the closed-loop. These two at first glance contrary goals have been both successfully solved by exploiting the T-S notation. We have derived a LMI condition that allows to estimate nested DAs based on convex optimization. This LMI has been added to our Algorithm 3.2.1. Each of the invariant set is calculated based on a separate T-S model. The parameters of each model, e.g. the UoD, have been considered as degrees of freedom for optimally estimating the nested DAs. By successively increasing the control performance from the outer towards the inner DA, both a large DA and a performant control behavior have been achieved. In order to ensure the asymptotic stability of a desired set point, we have established a proper switching condition. This condition has been finally generalized to non-nested invariant sets.

Chapter 5

Smooth Switching

As a matter of fact in switching-based control, a non-smooth changing of the control input occurs while switching. Although a switched controller relaxes the trade-off between different control objectives, e.g. size of the estimated domain of attraction (DA) and performance as shown in the previous chapter, a discontinuous control input might not always be desired or allowed. Due to that, smoothly switching controller algorithms are highly recommended. For instance, a switched controller for an active suspension system reduces the ride comfort for the carrying persons which is bypassed by smoothly blending between the regulators [68].

Motivated by that we extend our results for switched controller design of the previous chapter to smooth switching. After a formal problem statement in Section 5.1 we derive smoothly switched control laws in Section 5.2. Finally, we introduce an unified approach for reformulating hard switched systems in T-S notation into smoothly switched ones. The approach is not restricted to systems that have an equilibrium, hybrid automata (see Section 2.3) can be handled as well. In addition, an easy linguistical interpretation and understanding of the switching conditions is achieved.

5.1 Problem Formulation

Due to the fact that a hard switched control law might not always be allowed due to safety, comfort or material stress, we address the following problem:

Problem 5.1.1. Modify the switched control schemes introduced in Chapter 4 such that smooth switching is achieved.

We solve the above problem in Section 5.2 by a novel convex interpolation between switched controllers. The basic idea is to find an auxiliary T-S controller for each switching condition. This controller has to ensure the asymptotic stability of the system for both DAs, the currently active one and the one that should be activated according to the switching law. Hence, we are able to interpolate from the actual controller to the desired one by blending to the auxiliary controller first. Thereby, the active Lyapunov function only changes once whereby asymptotic stability is ensured. We formulate a LMI condition for computing the required auxiliary controllers with the desired performance.

An additional problem is tha the interpretability of mathematical switching conditions becomes worse with an increasing number of switching conditions:

Problem 5.1.2. Reformulate a hard switched T-S system into a smoothly switched one such that an easy linguistical interpretation and understanding of the smooth switching conditions is achieved.

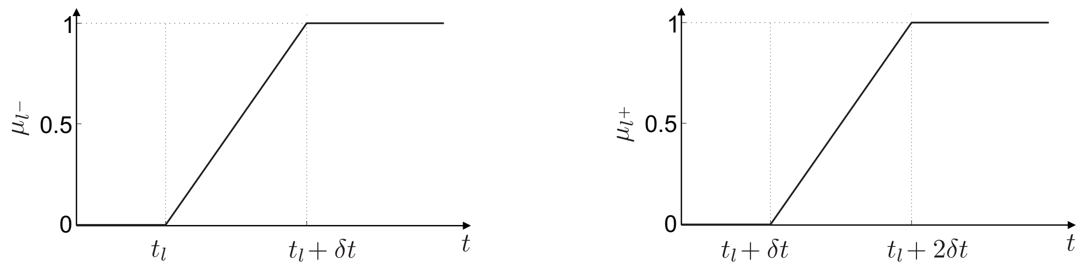
We contribute in solving this problem in Section 5.3 by introducing a recurrent fuzzy system (RFS) for switched T-S formulations. Contrary to a conventional RFS (see 2.1.1) the linguistical state variables are mapped to the switched closed-loop T-S models (instead to the system state variables). Thereby both, the interpretability and the smoothness of RFS can be directly exploited for switched system design.

5.2 Nested and Non-nested Smoothing

Consider that an asymptotically stable switching signal $l \in \mathbb{N}_{1:\zeta}$ based on Theorem 4.2.1 or Theorem 4.3.1 is given. Then we smooth the switched input signal to

$$\mathbf{u} = \begin{cases} \mathbf{u}_{l^-} & = \sum_{j=1}^{r} h_j(\mathbf{z}_c) \left(\mu_{l^-} \mathbf{F}_j^{l\pm} \mathbf{x} + (1 - \mu_{l^-}) \, \mathbf{F}_{j,l^-} \mathbf{x} \right) \text{ if } l : l^- \to l^+, \\ \mathbf{u}_{l^+} & = \sum_{j=1}^{r} h_j(\mathbf{z}_c) \left(\mu_{l^+} \mathbf{F}_{j,l^+} \mathbf{x} + (1 - \mu_{l^+}) \, \mathbf{F}_j^{l\pm} \mathbf{x} \right) \text{ if } l = l^+ \text{ and } \mu_{l^-} = 1, \end{cases} \tag{5.1}$$

where l^- and l^+ denote the values of the actual switching signal right before and right after switching, respectively. The fuzzy interpolation variables μ_{l^-} and μ_{l^+} fulfill the property $0 \leq \mu_{l^-} \leq 1$, $0 \leq \mu_{l^+} \leq 1$ and Fig. 5.1 depicts the corresponding membership functions. In other words, before and after the interpolation the T-S controller according to the pure switching condition l (Theorem 4.2.1 or 4.3.1) is active. If a switching occurs $l : l^- \to l^+$ at a time $t = t_l$ then μ_{l^-} increases until it becomes one at a time $t = t_l + \delta t$, $\delta t > 0$ (see Fig. 5.1(a)). During that time-span, the controller is interpolated according



(a) Membership function μ_{l^-}

(b) Membership function μ_{l^+}

Figure 5.1: Convex interpolation variables for smooth switching.

to the upper case in (5.1) from the active one towards an auxiliary controller

$$\mathbf{u} = \sum_{j=1}^{r} h_j(\mathbf{z}_c) \left(\mathbf{F}_j^{l\pm} \mathbf{x} \right) \tag{5.2}$$

which is individually defined for a particular switching $l : l^- \to l^+$. Consequently, a T-S controller (5.2) exists for any switching that can occur. As soon as this controller is fully active, the interpolation according to the lower case in (5.1) smoothly activates the desired T-S controller $\sum_{j=1}^{r} h_j(\mathbf{z}_c) \left(\mathbf{F}_{j,l^+} \mathbf{x} \right)$ until $t = t_l + 2\delta t$ (see Fig. 5.1(b)). Concerning the smoothly switched control law (5.1), asymptotic stability of the closed-loop system is guaranteed by the following Theorem:

Theorem 5.2.1. *Let* $\zeta > 1$, $\zeta \in \mathbb{N}$, *closed-loop SE-NL T-S models (or LO T-S models without affine terms) for a dynamical system and an asymptotically stable switching signal* $l \in \mathbb{N}_{1:\zeta}$ *based on Theorem 4.2.1 or 4.3.1 for the corresponding controllers* (4.6) *be given. Then the smooth switching signal* (5.1) *asymptotically stabilizes the origin if the feedback matrices* $\mathbf{F}_j^{l\pm}$ *for a switching* $l : l^- \to l^+$ *fulfill*

$$\mathbf{Q}_w \left(\mathbf{A}_{iw} + \mathbf{B}_{iw} \mathbf{F}_j^{l\pm} \right)^T + \left(\mathbf{A}_{iw} + \mathbf{B}_{iw} \mathbf{F}_j^{l\pm} \right) \mathbf{Q}_w < \mathbf{0}, \quad \forall\, i, j \in \mathbb{N}_{1:r}, w \in \left\{ l^-, l^+ \right\} \tag{5.3}$$

with $\mathbf{Q}_w = \left(\frac{\mathbf{P}_w}{\eta_{\mathbf{0},w}} \right)^{-1}$ *and* $w \in \{l^-, l^+\}$.

Proof: Equation (5.3) is obtained from (2.36) replacing $\mathbf{Q}$ with $\mathbf{Q}_w$ and $\tilde{\mathbf{F}}_j$ with $\mathbf{F}_j^{l\pm} \mathbf{Q}_w$, $w \in \{l^-, l^+\}$. Hence, the asymptotic stability of the T-S controllers (5.2). In addition, $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_{l^-}, \eta_{\mathbf{0},l^-})$ and $\mathcal{X}_{\mathbf{0}}(\mathbf{P}_{l^+}, \eta_{\mathbf{0},l^+})$ are valid DAs for the closed-loop T-S models with $w = l^-$ and $w = \to l^+$, respectively. Thereby, the interpolation (5.1) results in a single change of the active Lyapunov function, equivalent to the original switching signal $l : l^- \to l^+$. Consequently, the asymptotic stability is still guaranteed based on Theorem 4.2.1 or Theorem 4.3.1 which concludes the proof. ■

Indeed, (5.3) can be used to check whether a desired smoothing controller (5.2) is valid or not. But more important, they can be also used for constructively designing and optimizing a valid controller by considering the feedback matrices $\mathbf{F}_j^{l\pm}$ as free LMI variable (e.g. see Example 2.2.3).

Remark 5.2.1. The smoothness of the input signal is proportional to δt. However, a large δt increases the required time span of the interpolation whereby a further interpolation (switching) might have to be delayed until $\mu_{l^+} = 1$ is fulfilled.

If some of the $\zeta > 1$ closed-loop SE-NL T-S models (or LO T-S models without affine terms) have a common quadratic Lyapunov function and hence their DAs only differ in the bounding level value (see Proposition 4.2.1), a controller (5.2) is not required for a smooth switching:

Corollary 5.2.1. *Let $\zeta > 1$, $\zeta \in \mathbb{N}$, closed-loop SE-NL T-S models (or LO T-S models without affine terms) for a dynamical system and an asymptotically stable switching signal $l \in \mathbb{N}_{1:\zeta}$ based on Theorem 4.2.1 or 4.3.1 for the corresponding controllers* (4.6) *be given. Then the smooth switching signal*

$$\mathbf{u} = \sum_{j=1}^{r} h_j(\mathbf{z}_c)\left(\mu_{l^-}\mathbf{F}_{j,l^+}\mathbf{x} + (1-\mu_{l^-})\,\mathbf{F}_{j,l^-}\mathbf{x}\right) \quad \textit{if } l : l^- \to l^+, \tag{5.4}$$

for a switching $l : l^- \to l^+$ asymptotically stabilizes the origin if the DAs $\mathcal{X}_\mathbf{0}(\mathbf{P}_{l^-}, \eta_{\mathbf{0},l^-})$ and $\mathcal{X}_\mathbf{0}(\mathbf{P}_{l^+}, \eta_{\mathbf{0},l})$ share a common $\mathbf{P} = \mathbf{P}_{l^-} = \mathbf{P}_{l^+}$.

Proof: The asymptotic stability follows directly from Theorem 5.2.1 when replacing $\mathbf{F}_j^{l\pm}$ with $\mathbf{F}_{j,l^+}$ and $\mathbf{F}_{j,l^-}$. In other words, an arbitrary convex interpolation between linear systems that have a common (quadratic) Lyapunov function asymptotically stabilizes the origin whereby the proof is concluded. ■

5.3 Recurrent Fuzzy Switching: A Unified Approach

Generally, the understanding of switching laws becomes harder with a rising amount of switching rules (conditions). In addition, there exist, up to the author's knowledge, no framework for transforming an asymptotically stable switched system into an asymptotically stable and smoothly interpolated one. Both disadvantages, and thus Problem 5.1.2, are tackled in this section.

Consider a hard switched closed-loop system which is written in T-S notation

$$\dot{\mathbf{x}} = \sum_{p=1}^{\zeta} h^p(\mathbf{z}_w) \underbrace{\sum_{i=1}^{r}\sum_{j=1}^{r} h_i^p(\mathbf{z}_s) h_j^p(\mathbf{z}_c) \left(\left(\mathbf{A}_i^p + \mathbf{B}_i^p \mathbf{F}_j \right) \mathbf{x} + \mathbf{a}_i^p \right)}_{\Sigma^p}, \tag{5.5}$$

where $p \in \mathbb{N}_{1:\zeta}$ denotes the currently active T-S system Σ^p. The hard switching signal

$$\sum_{p=1}^{\zeta} h^p(\mathbf{z}_w) = 1, \quad h^p(\mathbf{z}_w) \in \mathbb{N}_{0:1} \;\; \forall\, h^p(\mathbf{z}_w) \tag{5.6}$$

depends on premise variables $\mathbf{z}_w$ which consist of input and state parameters as well as external signals, such as a clock signal, which might the system cause to switch. Note that (5.5) represents both, a closed-loop SE-NL T-S and a closed-loop LO T-S formulation according to the Definitions 2.1.1 and 2.1.2, respectively.

We interpret each closed-loop T-S system Σ^p as a linguistic partition of a state variable $\bar{x}$ of a discrete-time recurrent fuzzy system (RFS), which has been introduced in Section 2.1.1). Therefore, we introduce the fuzzy set $L_j^{\bar{x}} \in \left\{ L_1^{\bar{x}}, \dots, L_\zeta^{\bar{x}} \right\}$. The RFS describes the switching rule base as

$$\text{If } \bar{x}(k) \text{ is } L_j^{\bar{x}} \text{ and } \bar{\mathbf{u}}(k) \text{ is } \mathbf{L}_{\mathbf{q}}^{\bar{\mathbf{u}}} \text{ then } \bar{x}(k+1) \text{ is } L_{w(j,\mathbf{q})}^{\bar{x}} \tag{5.7}$$

analogous to the rule (2.1). Hence, we will call it *switched RFS* henceforth. The vector $\mathbf{L}_{\mathbf{q}}^{\bar{\mathbf{u}}} = \left[L_q^{\bar{u}_1}, \dots, L_q^{\bar{u}_m} \right]^T$ summarizes linguistic values for each element $\bar{u}_p$, $p \in \mathbb{N}_{1:m}$. These linguistic values are $L_q^{\bar{u}_p}$ $(q \in \mathbb{N}_{1:z})$. Note, the switching rule base (5.7) has only one state variable, which denotes the active subsystem Σ^p, but m input signals that can cause a switching to occur. Analogous to the mathematical expression (2.4) of a conventional RFS rule basis

$$\sum_{p=1}^{\zeta} h^p(\mathbf{z}_w) = \bar{x}(k+1) = \frac{\sum_{j,\mathbf{q}} \mathbf{s}_{\mathbf{L}_{w(j,\mathbf{q})}}^{\bar{x}} \prod_p \mu_{L_j}^{\bar{x}}(\bar{x}) \mu_{L_q}^{\bar{u}_p}(\bar{u}_p)}{\sum_{j,\mathbf{q}} \prod_p \mu_{L_j}^{\bar{x}}(\bar{x}) \mu_{L_q}^{\bar{u}_p}(\bar{u}_p)}, \tag{5.8a}$$

$$\sum_j \mu_{L_j}^{\bar{x}}(\bar{x}) = \sum_q \mu_{L_q}^{\bar{u}_p}(\bar{u}_p) = 1, \qquad \mu_{L_j}^{\bar{x}}(\bar{x}) \geq 0,\ \mu_{L_q}^{\bar{u}_p}(\bar{u}_p) \geq 0 \tag{5.8b}$$

represents the smooth mathematical form of (5.7). We postulate the following three definitions for deriving a switched RFS. Thereby, ξ is considered as a general switching parameter:

Definition 5.3.1 (Fuzzification of non-strict switching conditions)**.** A non-strict switching rule defines a region of ξ within which switching is allowed. For instance, switch to system Σ^p with $p = 1$ if $\xi \leq 10$. A non-strict rule can thus be seen as a switching condition of a hybrid automaton (see Section 2.3). Such a switching condition can be fuzzyfied as input signal of the switched RSF by each commonly used membership function except singletons (see Fig. 2.2). Singletons are not allowed as they prevent a smooth interpolation. The core positions of the membership functions can be freely chosen as long as it is ensured that: First, the interpolation starts not before the non-strict switching condition is fulfilled. Second, the interpolation is done before a strict switching condition (see the following definition) is violated. Third, an interpolation towards a subsystem has to be done before interpolating from that subsystem towards another one is allowed.

Definition 5.3.2 (Fuzzification of strict switching conditions)**.** A strict switching rule contains a strict equal sign, e.g. switch to system Σ^p with $p = 1$ if $\xi = 10$. Thus, it can be seen as invariant condition of a hybrid automaton (see Section 2.3). Such a condition has to be fuzzyfied by using a rectangular membership function which is either 0 or 1 [64]. A rectangular membership function can be easily obtained from a trapezoidal one. For instance, by merging $s_1(.)$ and $s_2(.)$ as well as $s_3(.)$ and $s_4(.)$ in Fig. 2.2(a). Consequently, a strict switching condition forces switching to occur.

Definition 5.3.3 (Fuzzification/Defuzzification of subsystems)**.** The subsystems in (5.5) represent the linguistic set of the RFS state variable $\bar{x}$. Here, singletons according to Fig. 2.2(c) are considered for defuzzification. The fuzzification is bypassed and the actual values of the singletons are directly used instead.

Fig. 5.2 sketches the final structure of the novel switched RFS: Shown are the T-S subsystems Σ^p, $p \in \mathbb{N}_{1:\zeta}$. The switching rule base (5.7) is depicted by a deterministic automation (as introduced in Section 2.1.1). The rule base interpolates between the T-S subsystems based on (5.8) by triggering the blending functions h^p, $p \in \mathbb{N}_{1:\zeta}$. For instance, if the automaton state is $L_2^{\bar{x}}$ then $h^2 = 1$ and the T-S system Σ^2 is active. A change of the automation state and thus a smooth switching between the T-S systems is here caused by $\mathbf{x}$, $\dot{\mathbf{x}}$ or an external trigger τ.

The switched RFS is an unified smoothing approach for systems in form of (5.5) allowing a high interpretability of the switching conditions due to the linguistical rule base (5.7). For instance, all smooth switching approaches of Section 5.2 can be directly transferred to a switched RFS which is summarized in the following Corollary:

Corollary 5.3.1. *The smooth switching approaches according to Theorem 5.2.1 and Corollary 5.2.1 can be rewritten as switched RFS.*

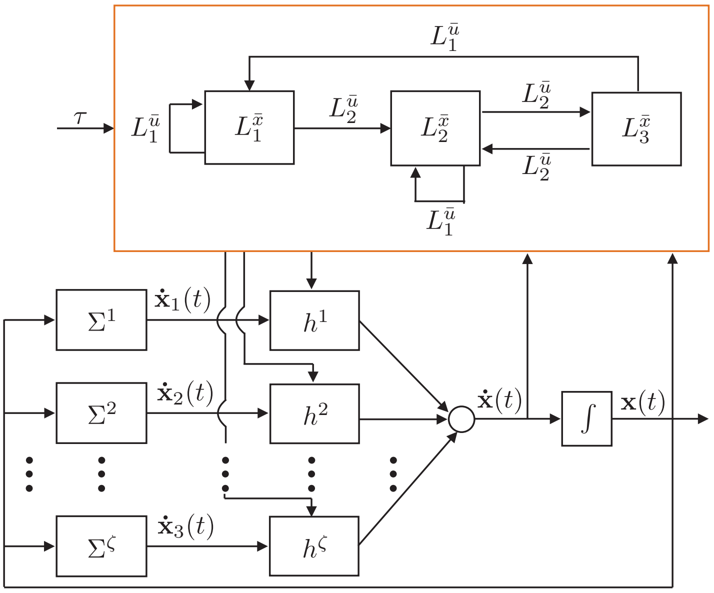


Figure 5.2: Block schematic of a swiched recurrent fuzzy system.

Proof: A transfer of the results is possible because of the following three facts: First, in Theorem 5.2.1 and Corollary 5.2.1 switched closed-loop system in T-S notation are considered which can equivalently be written according to (5.5). Second, the switched RFS ensures a convex combination (5.8b) of the subsystems which is equivalent to the smooth switching according to (5.1) and (5.4). That is required for the proofs of Theorem 5.2.1 and Corollary 5.2.1. Third, the Definitions 5.3.1 till 5.3.3 guarantee that the interpolation of the switched RFS is in accordance with (5.1) and (5.4). ■

In addition, a hybrid automaton that does not have a conventional equilibrium but a stable ϵ-region (region stability according to Definition 2.3.1) can be reformulated and smoothed by a switched RFS:

Theorem 5.3.1. *Let a hybrid automaton that is regionally stable in a known ϵ-region be given. Then the system is regionally stable within a region $\tilde{\epsilon} \leq \epsilon$ if the hybrid automaton is transformed into a switched RFS and the switching and invariant conditions are reformulated based on the Definitions 5.3.1 to 5.3.3.*

Proof: The stable ϵ-region of a regionally stable hybrid automaton is defined by certain combinations of subsystems of the overall switched system and switching conditions, i.e. worst case scenarios or switching sequences. Its reformulation to a switched RFS does interpolate but not change the original switching laws. In other words, due to the Definitions 5.3.1 to 5.3.3, the extreme values of the switched RFS are equivalent to the original switching conditions. Hence, the stable region $\tilde{\epsilon}$ of the switched RFS cannot become larger than the ϵ-region of the original hybrid automaton. However, the $\tilde{\epsilon}$-region can become smaller than the ϵ-region because the interpolation balances contrary switching rules. Consequently, the ϵ-region is an upper bound of the $\tilde{\epsilon}$-region which concludes the proof. ■

Remark 5.3.1. An adjustment of the switched RFS might be of interest. For example, for minimizing the $\tilde{\epsilon}$-region (see Theorem 5.3.1). Due to its formulation as fuzzy system, several fuzzy or neuro-fuzzy based optimization algorithms are directly applicable [46, 111]. In [30] for instance, weighting coefficients

$$\bar{\mathbf{p}}^{x}_{\mathbf{C}_w} = \begin{bmatrix} \bar{p}^{\bar{x}}_{C_1} & \bar{p}^{\bar{x}}_{C_2} & \cdots & \bar{p}^{\bar{x}}_{C_z} \end{bmatrix}, \quad \bar{p}^{\bar{x}}_{C_i} > 0, \ \forall \ i \in \mathbb{N}_{1:z} \tag{5.9}$$

have been considered to extend (5.8a) to

$$\sum_{p=1}^{\zeta} h^p(\mathbf{z}_w) = \bar{x}(k+1) = \frac{\sum_{j,\mathbf{q}} \mathbf{s}^{\bar{x}}_{\mathbf{L}_{w(j,\mathbf{q})}} \bar{\mathbf{p}}^{\bar{x}}_{\mathbf{C}_{w(j,\mathbf{q})}} \prod_p \mu^{\bar{x}}_{L_j}(\bar{x}) \mu^{\bar{u}_p}_{L_q}(\bar{u}_p)}{\sum_{j,\mathbf{q}} \bar{\mathbf{p}}^{\bar{x}}_{\mathbf{C}_{w(j,\mathbf{q})}} \prod_p \mu^{\bar{x}}_{L_j}(\bar{x}) \mu^{\bar{u}_p}_{L_q}(\bar{u}_p)}. \tag{5.10}$$

Thereby, a modification of the interpolation law is achieved by strengthening or weakening single switched RFS rules. As the rule base itself is not changed, the stability results according to Corollary 5.3.1 and Theorem 5.3.1 stay valid. This approach has been successfully applied for modeling and optimizing a production environment and for adapting an active cruise control system [32, 107].

5.4 Summary

In this chapter, we have addressed the problems that arise from switching-based control: First, non-smooth changes of the control input might cause performance and comfort losses or safety risks. Second, the loss of interpretability and adaptability of switching conditions. We solve that problems by extending our switched controller strategies (Chapter 4) to enable an asymptotically stable and smooth switching. Therefore, we have invoked arguments based on common and multiple Lyapunov functions. The required stability conditions have been formulated in terms of linear matrix inequalities (LMIs) such that a numerically efficient convex optimization procedure can be applied.

Concerning the second problem, we have derived a general representation of a switched T-S model by a switched recurrent fuzzy system (RFS). The switched RFS allows a linguistic interpretation of the switching rules while simultaneously smoothing them. Also modifying the switching parameter has thereby been enhanced.

Chapter 6

Governor Integrated Nominal-Value Adaptation: GINA Controller

In the Chapters 3 and 4 we have derived several numerical methods for estimating a large ellipsoidal domain of attraction (DA) of a desired equilibrium point while ensuring a requested control performance. As a matter of fact, an estimated DA of a constrained dynamical system (linear and nonlinear) is always bounded by a critical level value. Hence, the asymptotic stabilization of a desired set point cannot be guaranteed for every initial state that is outside of the bounding level set. In this chapter, we investigate how a set point change can be realized, even if the current state vector is not within the estimated DA of the desired equilibrium. To this end, we develop the *GINA controller* (Governor Integrated Nominal-Value Adaptation) which belongs to the class of reference governors. We revive the Lyapunov function-based set point governor [21] for linear systems with a single input. We generalize its operation principle to systems in T-S form with multiple inputs. We propose several strategies to balance the required computational costs with the optimality of the solution. Consequently, recalling the scope of this thesis (Section 1.2) we contribute to the design of reference governors.

After a formal problem statement in Section 6.1, we explain the desired operation principle of the GINA controller in general. Section 6.3 is devoted to the optimal real-time calculation of the governor's solution. We derive an algorithm for realizing an optimal real-time calculation. Strategies for relaxing the required computational effort are developed in Section 6.4. We also generalize the governor algorithm such that its applicability is only restricted by a single condition. In Section 6.5 we even relax that condition by developing an unified iterative implementation of the governor principle. Thereby, most of the required calculations are done before the real-time execution such that the real-time computation is reduced to a selection of a valid solution among the pre-calculated ones.

6.1 Problem Formulation

Each physical system is subject to constraints, e.g. every actuator has a limited operating range. Hence, the domain of attraction (DA) of a real world system is most often restricted. An estimation of such a *local* DA can for instance be done based on the Chapters 3 and 4. Consequently, if the state vector is outside of the estimated DA we can not guarantee the asymptotic stability of the system anymore. Now, the question arises how this drawback can be solved. We will face the following problem:

Problem 6.1.1. Let a saturated input-affine nonlinear system in the form

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\boldsymbol{\sigma}\left(\mathbf{x}\right) + \mathbf{B}(\mathbf{x})\boldsymbol{\sigma}\left(\mathbf{u}\right), \quad (\text{or } \boldsymbol{\sigma}\left(\dot{\mathbf{u}}\right)) \tag{6.1}$$

with the state vector $\mathbf{x} \in \mathbb{R}^n$ and the input vector $\mathbf{u} \in \mathbb{R}^m$. The system might have state constraints. The input might be either restricted in its amplitude or rate or even both and a related T-S formulation (2.5) or (3.11) is given. Suppose that a corresponding T-S controller and an estimated DA $\mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$ of the closed-loop system of the desired equilibrium $(\mathbf{x}_d^*, \mathbf{u}_d^*)$ subject to all relevant system constraints is computed. Then the problem is to stabilize the desired set point if the actual state vector is not within the estimated DA, i.e. $\mathbf{x} \notin \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$.

The following assumption is required in order to tackle this problem:

Assumption 6.1.1. A functional relation of allowed system's equilibria is known. Either as smooth piecewise analytical function (at least C^0-continuous)

$$\mathbf{x}^* = \hat{\mathbf{f}}(\mathbf{x}), \quad \text{s. t.}\, |\mathbf{u}^*| < \mathbf{u}_{max}\; \forall\, \mathbf{x}^* \in \hat{\mathbf{f}}(\mathbf{x}) \tag{6.2}$$

with $\mathbf{x}^* < \boldsymbol{\sigma}\left(\mathbf{x}\right)$ or in form of a numerically calculated steady-state curve

$$\mathbf{x}^*(k) = \hat{\mathbf{f}}(\mathbf{x}(k)), \quad \text{s. t.}\, |\mathbf{u}^*(k)| < \mathbf{u}_{max}\; \forall\, \mathbf{x}^*(k) \in \hat{\mathbf{f}}(\mathbf{x}(k)) \tag{6.3}$$

consisting of discrete equilibria points $\mathbf{x}^*(k) < \boldsymbol{\sigma}\left(\mathbf{x}(k)\right)$, where k is the discretization index (of discrete equilibria).

As a matter of fact, this will be inherently satisfied for many physical real-wold systems. For instance, stirred tank reactor, wind turbine systems, mobile robots, manipulators or quadrotors as considered in [92, 105, 128].

Based on that preparation, we solve Problem 6.1.1 by establishing a reference governor, called *GINA controller* (Governor Integrated Nominal-Value Adaptation), for T-S systems, up to the author's knowledge, for the first time. If the actual state vector of the closed-loop system is outside of the estimated DA then the GINA controller computes an auxiliary set point from the allowed system's equilibria (see Assumption 6.1.1) that can be stabilized subject to system constraints. While the system is approaching to this set point, it is shifted as fast as possible towards the actual desired equilibrium. Thereby, robustness against external disturbances or measurement noise is achieved simultaneously. Different algorithms for realizing this working principle are introduced for guaranteeing the applicability of the GINA controller to a wide range of systems and for handling different levels of available computational power. For instance, we derive conditions such that the computation becomes independent of the number of system inputs.

6.2 General Operation Principle

Assume the situation according to Problem 6.1.1 where the current state vector is not within the DA of the desired equilibrium, meaning $V_{\mathbf{x}_d^*} > \eta_{\mathbf{u}_d^*}$. Then the general idea of the GINA controller is to determine an alternative equilibrium (a temporary set point $(\mathbf{x}_t^*, \mathbf{u}_t^*)$) which is as close as possible towards the desired one, but can be guaranteed to be stabilized. To this end, the closed-loop system is expanded by the GINA controller as depicted in Fig. 6.1 whereby the saturation function denotes input limitations in general (amplitude and rate).

In order to ensure a stabilization, the temporary equilibrium $\mathbf{x}_t^*$ and its corresponding steady-state input $\mathbf{u}_t^*$ should be calculated such that

$$V_{\mathbf{x}_t^*} = (\mathbf{x} - \mathbf{x}_t^*)^T \mathbf{P} (\mathbf{x} - \mathbf{x}_t^*) \leq \eta_{\mathbf{u}_t^*}. \tag{6.4}$$

According to Assumption 6.1.1, all possible temporary equilibria are either part of a

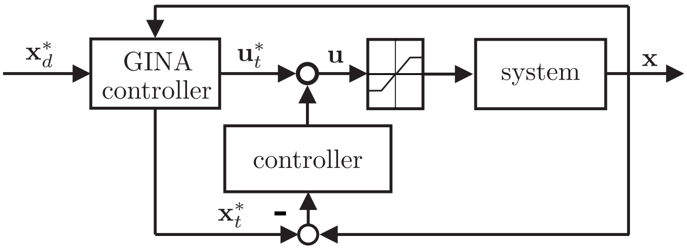


Figure 6.1: GINA controller extended closed-loop system.

piecewise analytical function (6.2) or a numerically calculated steady-state curve (6.3). The temporary equilibrium $\mathbf{x}_t^*$ is derived by interpolating between a reference equilibrium $(\mathbf{x}_r^*, \mathbf{u}_r^*)$ and the desired one, i.e.

$$\mathbf{x}_t^* = \hat{\mathbf{f}}\left(\mathbf{x}_r^* + c\left(\mathbf{x}_d^* - \mathbf{x}_r^*\right)\right), \tag{6.5a}$$

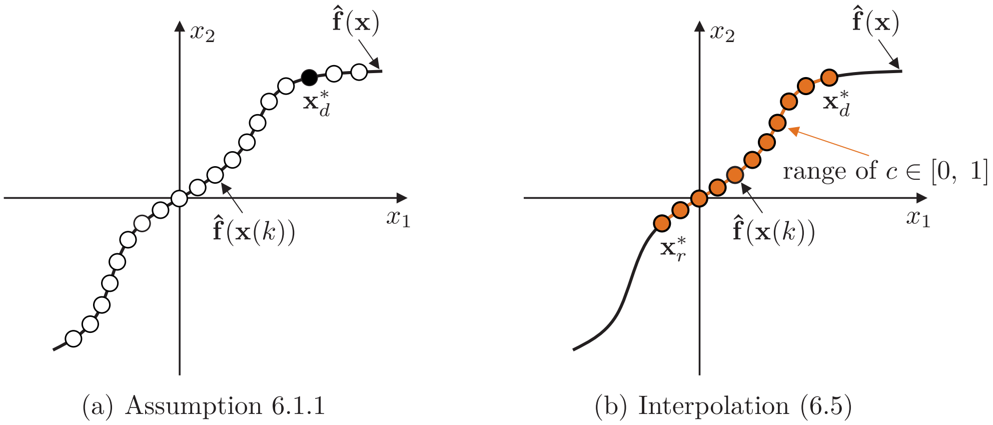
$$\mathbf{x}_t^*(k) = \hat{\mathbf{f}}\left(\mathbf{x}_r^* + c(k)\left(\mathbf{x}_d^* - \mathbf{x}_r^*\right)\right) \tag{6.5b}$$

with $c \in [0, 1]$ being the scaling (interpolation) factor. In other words, if $c = 0$ then $\mathbf{x}_t^* = \mathbf{x}_r^*$ and $c = 1$ results in $\mathbf{x}_t^* = \mathbf{x}_d^*$. The interpolation principle is clarified in Fig. 6.2 for a two-dimensional state space. Fig. 6.2(a) shows a possible analytical function of allowed system equilibria $\hat{\mathbf{f}}(\mathbf{x})$ as well as possible discrete equilibria $\hat{\mathbf{f}}(\mathbf{x}(k))$ according to Assumption 6.1.1. The desired set point $\mathbf{x}_d^*$ is marked. Assuming that a stabilizable reference equilibrium $\mathbf{x}_r^*$ is known the resulting range for c is depicted in Fig. 6.2(b). While a certain temporary equilibrium is stabilized, the scaling factor c should be increased as soon as possible, meaning that $\mathbf{x}_t^*$ is shifted towards $\mathbf{x}_d^*$ whenever stability can be guaranteed for the updated temporary equilibrium. Repeating will finally lead to $c = 1$ whereby $\mathbf{x}_t^* = \mathbf{x}_d^*$ and thus $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$.

However, after this more or less conceptional explanation of the GINA controller two fundamental questions remain open:

1. How to calculate the reference equilibrium $(\mathbf{x}_r^*, \mathbf{u}_r^*)$?
2. How to update c?

This questions will by answered in the following sections.


x_2
$\hat{\mathbf{f}}(\mathbf{x})$
$\mathbf{x}_d^*$
$\hat{\mathbf{f}}(\mathbf{x}(k))$
x_1
x_2
$\hat{\mathbf{f}}(\mathbf{x})$
$\mathbf{x}_d^*$
range of $c \in [0,\ 1]$
$\mathbf{x}_r^*$
$\hat{\mathbf{f}}(\mathbf{x}(k))$
x_1

(a) Assumption 6.1.1

(b) Interpolation (6.5)

Figure 6.2: Sketch of the calculation principle of $\mathbf{x}_t^*$.

6.3 Optimal Real-Time Computation

In this section we will develop an algorithm for analytically calculating the reference equilibrium and optimally updating the scaling factor c based on a function (6.5a). Optimal means that $\mathbf{x}_t^*$ is shifted as close as possible towards $\mathbf{x}_d^*$ while ensuring stability. This is fulfilled if (6.4) is reformulated to

$$V_{\mathbf{x}_t^*} = (\mathbf{x} - \mathbf{x}_t^*)^T \mathbf{P} (\mathbf{x} - \mathbf{x}_t^*) \overset{!}{=} \eta_{\mathbf{u}_t^*}. \tag{6.6}$$

To this end, we consider the class of systems (6.1) which have an equilibrium function (6.2) and fulfill the following additional assumption:

Assumption 6.3.1. The estimated DA $\mathcal{X}_{\mathbf{x}^*}(\mathbf{P}, \eta_{\mathbf{u}^*})$ for each equilibrium of (6.2) share a common matrix $\mathbf{P}$.

Although this might sound restrictive, indeed it is not. Lots of technical systems fulfill this assumption inherently, e.g. mobile robots, manipulators or aircraft systems and quadrotors [92, 103]. Mathematically spoken, if the values of the system's nonlinearities are constant in each allowed equilibrium then Assumption (6.3.1) is fulfilled anyway. For instance, the tilt angles of wheeled inverted pendulums (like the commercial Segway product [3]) are zero in each equilibrium.

The assigned objective is to continuously update c such that (6.6) holds true until $\mathbf{x}_t^* = \mathbf{x}_d^*$ and thus $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$. This is sketched in Fig. 6.3 concerning the function $\hat{\mathbf{f}}(\mathbf{x})$ of Fig. 6.2. The figure illustrates the continuous shifting of the temporary equilibrium along the allowed equilibrium function. The initial state vector $\mathbf{x}_0$ is not within the DA of the desired set point, but $\mathbf{x}_0 \in \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$. A temporary equilibrium is set as close as possible towards $\mathbf{x}_d^*$ such that $\mathbf{x}_0 \in \mathcal{X}_{\mathbf{x}_t^*(t_1)}(\mathbf{P}, \eta_{\mathbf{u}_t^*(t_1)})$. In order to keep the state vector at the bounding level set of the temporary equilibrium, is has to be continuously moved until $\mathbf{x}_t^* = \mathbf{x}_d^*$ and thus $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$. Three snap shots at a time t_1, t_2 and t_3 visualize the moving of the temporary equilibrium.

Replacing $\mathbf{x}_t^*$ in (6.6) with (6.5a) results in

$$V_{\mathbf{x}_t^*} = (\mathbf{x} - \hat{\mathbf{f}}(\mathbf{x}_r^*, \mathbf{x}_d^*, c))^T \mathbf{P} (\mathbf{x} - \hat{\mathbf{f}}(\mathbf{x}_r^*, \mathbf{x}_d^*, c)) \overset{!}{=} \eta_{\mathbf{u}_t^*} \tag{6.7}$$

which has to be solved in c.

Remark 6.3.1. Equation (6.7) only depends on the continuous state vector $\mathbf{x}$ and the smooth function (6.5a). In view of that, the solution for c is a smooth function as well (no jumps occur).

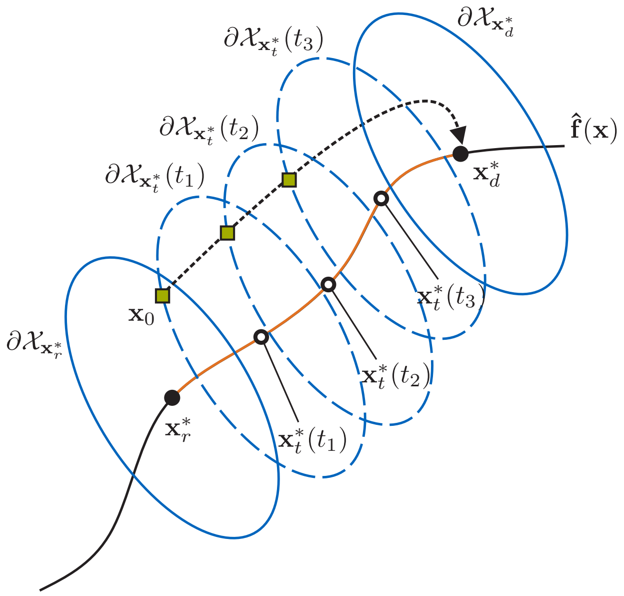

$\partial\mathcal{X}_{\mathbf{x}_t^*}(t_3)$
$\partial\mathcal{X}_{\mathbf{x}_d^*}$
$\hat{\mathbf{f}}(\mathbf{x})$
$\partial\mathcal{X}_{\mathbf{x}_t^*}(t_2)$
$\partial\mathcal{X}_{\mathbf{x}_t^*}(t_1)$
$\mathbf{x}_d^*$
$\mathbf{x}_t^*(t_3)$
$\mathbf{x}_0$
$\partial\mathcal{X}_{\mathbf{x}_r^*}$
$\mathbf{x}_t^*(t_2)$
$\mathbf{x}_r^*$
$\mathbf{x}_t^*(t_1)$

Figure 6.3: Operation principle of the optimal real-time computation.

Algorithm 6.3.1 summarizes the strategy for updating c: We initially calculate $\mathbf{x}_r^*$ while the rest of the algorithm is supposed to be executed in real-time, meaning in every simulation or real-time execution step. The reference equilibrium is derived based on the initial state vector $\mathbf{x}_0$ by solving the optimization problem

$$\mathbf{x}_r^* = \underset{\mathbf{x}^* \in \hat{\mathbf{f}}(\mathbf{x})}{\operatorname{argmin}} \; (\mathbf{x}_0 - \mathbf{x}^*)^T \mathbf{P} (\mathbf{x}_0 - \mathbf{x}^*) \geq 0. \tag{6.8}$$

In other words, $\mathbf{x}_r^*$ equals to the energy-optimal equilibrium concerning the ellipsoidal DA (2.38). The advantage of calculating $\mathbf{x}_r^*$ in such a way is that stability is enured as long as a stabilizable equilibrium which ensures (6.4) exist. The condition in line 4 is

Algorithm 6.3.1 Optimally updating the scaling factor c

```
 1: Initialization:
 2: calculate $\mathbf{x}_r^*$ by (6.8)
 3: Real-Time:
 4: if $V_{\mathbf{x}_d^*} \leq \eta_{\mathbf{u}_d^*}$ = true then
 5:     $c = 1$
 6: else
 7:     solve (6.7), (6.9) for every $i$ (ignoring the min-operator) and select $\mathcal{C}_{\eta_{\mathbf{u}_t^*}}$
 8:     $c = \max_q (c_{q,i})$, $q \in \mathbb{N}_{1:|\mathcal{C}_i|}$, calculate $\mathbf{u}_t^*$ by (6.10)
 9:     if $c \notin [0,\ 1]$ then
10:         recalculate $\mathbf{x}_r^*$ by (6.8) with the current $\mathbf{x}$
11: return to line 4
```

continuously checked. If it is fulfilled then the actual state vector is within the DA of the desired equilibrium $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$, whereby with $c = 1$ the desired equilibrium is activated. If line 4 is not fulfilled, the GINA controller becomes active.

Let us first consider a linear feedback law (2.23) (or (2.16) with $\mathbf{F}_j = \mathbf{F}$ for all j). Then the right-hand side of (6.7) is given by

$$\eta_{\mathbf{u}_t^*} = \min_i \left(\underbrace{\eta_{\mathbf{0}} \cdot \frac{\left(u_{max,i} - \left|f_{c,i}^T(\mathbf{x}_r^*, \mathbf{x}_d^*, c_i)\right|\right)^2}{u_{max,i}^2}}_{\eta_{u_{t,i}^*}} \right) \tag{6.9}$$

based on (2.49). The index i denotes the i-th input. Consequently, equation (6.7), (6.9) has to be solved and from the obtained values for c the one that shifts $\mathbf{x}_t^*$ as close as possible towards $\mathbf{x}_d^*$ has to be selected. This is executed in line 7 of the algorithm, where equation (6.7), (6.9) is solved for every input i while ignoring the min-operator. Thereby, sets $\mathcal{C}_i = [c_{1,i}, \dots, c_{j,i}]$ each containing $j \in \mathbb{N}$ valid solutions for the scaling parameter c are obtained. From these sets $\mathcal{C}_{\eta_{\mathbf{u}_t^*}}$ belonging to the smallest level value $\eta_{\mathbf{u}_t^*}$ (6.9) is selected (line 8) whereby the ignored min-operator is finally considered. The final value of the scaling parameter equals to the largest element in $\mathcal{C}_{\eta_{\mathbf{u}_t^*}}$ as that shifts $\mathbf{x}_t^*$ as far as possible towards $\mathbf{x}_d^*$. The corresponding steady-state input signal $\mathbf{u}_t^*$ which corresponds to a certain $\mathbf{x}_t^*$ is determinable by reformulating the equilibrium condition for a system (6.1) to

$$\mathbf{u}_t^* = -\left(\mathbf{B}^T(\mathbf{x}_t^*)\mathbf{B}(\mathbf{x}_t^*)\right)^{-1} \mathbf{B}^T(\mathbf{x}_t^*)\mathbf{A}(\mathbf{x}_t^*), \tag{6.10a}$$

$$\mathbf{u}_t^* = \mathbf{f}_c^T(\mathbf{x}_r^*, \mathbf{x}_d^*, c). \tag{6.10b}$$

The function $\mathbf{f}_c^T(\mathbf{x}_r^*, \mathbf{x}_d^*, c)$ is obtained by inserting (6.5a) in (6.10a). Note that due to assumption 6.1.1 and (6.5) no input saturation can occur in (6.10). The scaling factor c and thus the temporary equilibrium is continuously recalculated (updated) by repeating the real-time execution part of the algorithm (line 3). Line 10 of the algorithm is only executed in the case that $c \notin [0, 1]$ (out of its working range), e.g. due to external disturbances or measurement noise. A new valid reference equilibrium has to be calculated by solving (6.8) with $\mathbf{x}_0 = \mathbf{x}$ whereby robustness against disturbances is simultaneously achieved.

Analogous to a conventional control law the GINA controller requires the current state vector and is thus part of the feedback loop (see Fig. 6.1). The asymptotic stability of the GINA controller extended closed-loop system is given by the following Theorem:

Theorem 6.3.1. *Let a system* (6.1) *and an affine controller* $\mathbf{u} = \mathbf{u}_d^* + \mathbf{F}(\mathbf{x} - \mathbf{x}_d^*)$ *that asymptotically stabilizes* $\mathbf{x}_d^*$ $\forall$ $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$ *be given. Then the GINA controller extended closed-loop system asymptotically stabilizes* $\mathbf{x}_d^*$ *for all* $\mathbf{x} \in \bigcup_{(\forall\, \mathbf{x}_r^*)} \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$ *if a function* (6.5a) *exists, which fulfills*

$$(\mathbf{x}_t^* - \mathbf{x}_d^*)^T \frac{\partial \left(\hat{\mathbf{f}} \left(\mathbf{x}_r^* + c \left(\mathbf{x}_d^* - \mathbf{x}_r^* \right) \right) \right)}{\partial c} > 0 \tag{6.11}$$

for all $c \in [0, 1[$, *and* c *is updated by Algorithm 6.3.1.*

Proof: See the Appendix A.1 ■

Even in the case that (6.5a) does not fulfill the condition (6.11), the attraction of $\mathbf{x}_d^*$ is ensured by Theorem 6.3.1 which is proven by the following Lemma:

Lemma 6.3.1. *If a smooth piecewise equilibrium function* (6.5a) *exists which does not fulfill condition* (6.11) *then* $\mathbf{x}_d^*$ *is attractive concerning the GINA controller extended closed-loop system for all* $\mathbf{x} \in \bigcup_{(\forall\, \mathbf{x}_r^*)} \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$ *when updating* c *by Algorithm 6.3.1.*

Proof: The scaling factor $c \in [0, 1[$ will continuously increase due to $\dot{c} > 0$ according to the proof (Case 2) of Theorem 6.3.1. However, the value of the Lyapunov-like function (A.1) does not necessarily decrease as condition (6.11) is not guaranteed for all $c \in [0, 1[$. This becomes clear when looking at (A.5). However, $c = 1$ is guaranteed for $t \to \infty$ at least. As soon as $c = 1$ the GINA controller extended closed-loop system asymptotically stabilizes $\mathbf{x}_\mathbf{d}^*$ by Case 1 of the proof of Theorem 6.3.1. In other words, the temporary set point is shifted along (6.5a) until $\mathbf{x}_\mathbf{d}^*$ can be asymptotically stabilized. Hence, $\mathbf{x}_d^*$ is attractive which completes the proof. ■

Concerning a T-S controller (2.16) or (3.13), the equation (6.9) is not valid anymore. However, the asymptotic stability of the GINA controller extended closed-loop system can be still guaranteed in the following way:

Lemma 6.3.2. *Let a system* (6.1) *be given in T-S form. Let further an affine T-S controller* $\mathbf{u} = \mathbf{u}_d^* + \sum_{j=1}^{r} h_j(\mathbf{z}_c) \left(\mathbf{F}_j(\mathbf{x} - \mathbf{x}_d^*) \right)$ *that asymptotically stabilizes* $\mathbf{x}_d^*$, $\forall$ $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$ *be given. Then the GINA controller extended closed-loop system asymptotically stabilizes* $\mathbf{x}_d^*$ *for all* $\mathbf{x} \in \bigcup_{(\forall\, \mathbf{x}_r^*)} \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$ *when updating* c *by Algorithm 6.3.1 if equation* (6.11) *is valid and all linear subsystems* $(\mathbf{A}_i, \mathbf{B}_i)$ *of the T-S model of* (6.1) *fulfill the rank conditions:*

$$\operatorname{rank}(\mathbf{A}_i) < n, \quad \operatorname{rank}(\mathbf{A}_i, \mathbf{B}_i) = n \quad \forall\, i \in \mathbb{N}_{1:r}. \tag{6.12}$$

Proof: The steady state input signal $\mathbf{u}^*$ is zero at each possible equilibrium $\mathbf{x}^*$ of the system, due to the rank conditions (6.12). Thus, the bounding level value at each equilibrium is $\eta_{\mathbf{u}^*} = \eta_{\mathbf{0}}$, whereby the right-hand side of (6.7) is fixed at $\eta_{\mathbf{u}_t^*} = \eta_{\mathbf{0}}$. Considering that instead of (6.9), Algorithm 6.3.1 is applicable for a T-S controller and the proof of Theorem 6.3.1 can be applied to ensure the asymptotic stability which concludes the proof. ■

6.4 Relaxing the Real-Time Computation

Although Algorithm 6.3.1 provides an optimal updating of the temporary equilibrium $\mathbf{x}_t^*$ and its steady-state input signal $\mathbf{u}_t^*$, its practical applicability cannot be guaranteed in general. The reason for that derives its origin from the difficulty of solving optimization problems in real-time subject to an usually limited available computational power. For instance, microcontrollers offer a budget-friendly possibility for real-time tasks but at the price of a rather low complexity of executable real-time operations. Such problems are often not mentioned in the literature when optimal or nonlinear control laws are developed. In this section we develop strategies for relaxing the real-time applicability of the GINA controller. The Section 6.4.1 examines the recalculation of the reference equilibrium (line 10 of Algorithm 6.3.1) which might be required in case of an unexpected disturbance. The need of solving equation (6.7), (6.9) (line 7) for every input is relaxed in Section 6.4.2 such that only a single computation is required. In Section 6.4.3, we bypass the need of solving (6.7), (6.9) by introducing a fuzzy-based interpolation for iteratively improving the location of the temporary equilibrium.

6.4.1 Calculating the Reference Equilibrium

The need of solving the optimization problem (6.8) in real-time if a recalculation of the reference equilibrium is needed can be bypassed by the following Theorem:

Theorem 6.4.1. *The reference equilibrium $\mathbf{x}_r^*$ is given by*

$$\mathbf{x}_r^* = \mathbf{N}\left(\mathbf{N}^T\mathbf{P}\mathbf{N}\right)^{-1}\left(\mathbf{N}^T\mathbf{P}\mathbf{x}_0\right), \tag{6.13}$$

if all subsystems $(\mathbf{A}_i, \mathbf{B}_i)$, $i \in \mathbb{N}_{1:r}$, of the T-S notation of the system (6.1) share the same subspace of allowed equilibria, which is spanned by a constant matrix $\mathbf{N} \in \mathbb{R}^{n\times l}$.

Proof: If a matrix $\mathbf{N} \in \mathbb{R}^{n\times l}$ of system's equilibria exist then this subspace can be considered instead of the functional relation $\mathbf{x}^* = \hat{\mathbf{f}}(\mathbf{x})$ (see (6.2)). In other words, each allowed equilibrium is parametrized by $\mathbf{x}^* = \mathbf{N}\mathbf{z}$ where $\mathbf{z} = [z_1,\ z_2,\ ...,z_l]^T$ is the vector of independent directions within $\mathbf{N}$. Hence, the optimization problem (6.8) can be equivalently written to

$$\begin{aligned}\mathbf{x}_r^* &= \arg\min_{\mathbf{x}^*} \left((\mathbf{x}_0 - \mathbf{x}^*)^T \mathbf{P} \left(\mathbf{x}_0 - \mathbf{x}^*\right) \right), \quad \mathbf{x}^* = \mathbf{N}\mathbf{z}, \\ &= \mathbf{N}\left[\arg\min_{z} \left(\mathbf{x}_0^T \mathbf{P}\mathbf{x}_0 - 2\mathbf{z}^T\mathbf{N}^T\mathbf{P}\mathbf{x}_0 + \mathbf{z}^T\mathbf{N}^T\mathbf{P}\mathbf{N}\mathbf{z} \right)\right], \\ &= \mathbf{N}\left(\mathbf{N}^T\mathbf{P}\mathbf{N}\right)^{-1}\left(\mathbf{N}^T\mathbf{P}\mathbf{x}_0\right)\end{aligned}$$

where $\arg\min_z (...)$ is computed by solving

$$\frac{\delta}{\delta \mathbf{z}} = -2\mathbf{N}^T\mathbf{P}\mathbf{x}_0 + 2\mathbf{N}^T\mathbf{P}\mathbf{N}\mathbf{z} \overset{!}{=} \mathbf{0},$$

for $\mathbf{z}$ which concludes the proof. ■

Example 6.4.1. A new variant of mobile inverted pendulum robots with a single contact to the ground are the recently introduced *Ballbots* [55]. These omnidirectionally movable unstable robots balance upright on a sphere. The Ballbot system at our institute is shown in Fig. 6.4. The robot mainly consists of an aluminum cuboid frame F that carries batteries, electronics and possibly loads. The driving mechanism is based on three equidistantly positioned (120°) omniwheels W_i with $i \in \{1, 2, 3\}$ and a rigid ball B. Each of the omniwheels is driven by a DC-motor M_i. Ball clips C_i prevent slip between the omniwheels and the ball by three small roles at the end of the clips. Unstable (or dynamically stable) robots like a Ballbot achieve a higher agility, can be designed taller and carry more loads compared to statically stable robots having a similar footprint [79, 112]. That gives Ballbots a great potential for supporting humans

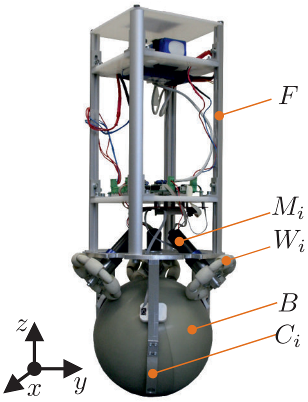


Figure 6.4: Ballbot test rig.

life which often takes place in a narrowed, cluttered or crowed environment – whether used as transportation or service robot or as a mobile information guide. A first existing commercial version of a Ballbot platform is *mObi* [1]. Some additional future visions for Ballbot applications are shown in [2]. Even without analyzing the equations of motion of the Ballbot, it is quiet obvious that the robot can be stabilized at each location (position in the (x, y)-plane) which span an allowed subspace of set points for analytically deriving the reference equilibrium $\mathbf{x}_r^*$.

Further possibilities of calculating and properly selecting the reference equilibrium are summarized in Appendix B.

6.4.2 Single Level Value

Generally, the equation (6.7), (6.9) must be solved for each input i in every execution step of Algorithm 6.3.1. Hence, it is quite obvious that the required computational power grows with the number of system inputs. In the following, we state some possibilities such that only a single input has to be considered:

Corollary 6.4.1. *Let a system* (6.1) *with multiple inputs in T-S form and a reference equilibrium* $\mathbf{x}_r^*$ *be given such that* $\mathbf{x}_0 \in \mathcal{X}(\mathbf{x}_r^*)$. *Then equation* (6.7) *and* (6.9) *have to be solved only once if all linear subsystems* $(\mathbf{A}_i, \mathbf{B}_i)$, $i \in \mathbb{N}_{1:r}$, *fulfill the rank conditions in* (6.12).

Proof: In Lemma 6.3.2 it is shown that a system which fulfills the rank conditions in (6.12) has a constant level value $\eta_\mathbf{0}$ at each equilibrium point. Consequently, (6.9) can be replaced wit $\eta_{\mathbf{u}_t^*} = \eta_\mathbf{0}$ whereby equation (6.7) and (6.9) have only to be solved once. ■

Even for a system which violates at least one of the rank conditions the evaluation of (6.7) and (6.9) can be reduced to a single input case. To this end, we state the following theorem and the subsequent corollary:

Theorem 6.4.2. *Let a system* (6.1) *with multiple inputs in T-S notation be given that violate at least on of the rank conditions* (6.12). *Consider that* $\mathbf{x}_r^* = \mathbf{0}$ *is a valid reference equilibrium. Then the minimal level value* $\eta_{\mathbf{u}_t^*}$ *equals always to the level value* $\eta_{u_{t,i}^*}$ *that belongs to a fixed (non-changing) input i concerning a defined* $\mathbf{x}_d^*$ *if linear equilibrium functions* (6.5a) *and* (6.10b) *exist, meaning*

$$\mathbf{x}_t^* = \mathbf{x}_r^* + c\left(\mathbf{x}_d^* - \mathbf{x}_r^*\right), \tag{6.14a}$$

$$\mathbf{u}_t^* = \mathbf{u}_r^* + c\left(\mathbf{u}_d^* - \mathbf{u}_r^*\right), \quad s.\ t.\, |\mathbf{u}_t^*| < \mathbf{u}_{max}\ \forall\, \mathbf{x}_t^*. \tag{6.14b}$$

Proof: See Appendix A.2. ■

In Theorem 6.4.2, $\mathbf{x}_d^* = \mathbf{0}$ is not permissible as this results in $\mathbf{x}_d^* = \mathbf{x}_r^*$ whereby the updating of c becomes meaningless ($c = 0$ is the only valid solution). This drawback is solved by the following corollary:

Corollary 6.4.2. *Let a system* (6.1) *with multiple inputs in T-S notation be given that violate at least on of the rank conditions* (6.12)*. Consider* $\mathbf{x}_d^* = \mathbf{0}$*,* $\mathbf{u}_d^* = \mathbf{0}$ *as desired equilibrium. Then the minimal level value* $\eta_{\mathbf{u}_t^*}$ *equals always to the level value* $\eta_{u_{t,i}^*}$ *that belongs to a fixed (non-changing) input i if a stabilizable reference equilibrium exist and the linear set point functions* (6.14) *are valid.*

Proof: Permuting the reference and the desired equilibria in (6.14) leads to

$$\mathbf{x}_t^* = \mathbf{x}_d^* + c\left(\mathbf{x}_r^* - \mathbf{x}_d^*\right), \mathbf{u}_t^* = \mathbf{u}_d^* + c\left(\mathbf{u}_r^* - \mathbf{u}_d^*\right), \quad \text{s. t.} \left|\mathbf{u}_t^*\right| < \mathbf{u}_{max} \; \forall \, \mathbf{x}_t^*. \tag{6.15}$$

Following Theorem 6.4.2 with the new equilibrium function (6.15) concludes the proof. ■

If Theorem 6.4.2 is not applicable due to the fact that no temporary equilibrium with $c \notin [0,1]$ for $\mathbf{x}_r^* = \mathbf{0}$ can be found, then Corollary 6.4.2 can be considered until Theorem 6.4.2 is applicable again. Consequently, Theorem 6.4.2 and Corollary 6.4.2 are always applicable if (6.14) is an equilibrium function.

Example 6.4.2 (Example 6.4.1 cont'd)**.** Let us recall the Ballbot system of Fig. 6.4 with the purpose of controlling a defined position in the (x, y)-plane. Due to the fact that each positon is a possible set point, the linear equilibrium functions (6.14) are valid and thus Theorem 6.4.2 as well as Corollary 6.4.2 can be applied.

6.4.3 Generalization: Implicit Recurrent Fuzzy Interpolation

According to Lemma 6.3.2 and Corollary 6.4.1 a T-S controller can only be considered when the linear subsystems of the T-S model fulfill certain conditions. This might be the case for lots of interesting physical systems, e.g. see Example 6.4.2, but not for all. In this section, we will tackle and relax this problem while simultaneously reducing the required computational power for updating the scaling variable c (see Algorithm 6.3.1). More precisely, solving equation (6.7), (6.9) is not required any more. The updating strategy is based on a recurrent fuzzy interpolation with the objective of iteratively improving the location of the temporary equilibrium while ensuring its asymptotic stabilization at any time. The method is summarized in Algorithm 6.4.1 and will be detailed in the following:

Algorithm 6.4.1 Implicit Recurrent Fuzzy Interpolation of $\mathbf{x}_t^*$

```
 1: Initialization:
 2: calculate $\mathbf{x}_r^*$ by (6.8) or Section 6.4.1 for $\mathbf{x}_0$
 3: set $\mathbf{x}_t^* = \mathbf{x}_r^*$, $\mathbf{u}_t^* = \mathbf{u}_r^*$ by (6.10)
 4: set k = 0
 5: set $\mathbf{x}_t^*(k) = \mathbf{x}_t^*$, $\mathbf{u}_t^*(k) = \mathbf{u}_t^*$
 6: Real-Time:
 7: while termination = false do
 8:   if $V_{\mathbf{x}_d^*} \leq \eta_{\mathbf{u}_d^*}$ = true then
 9:     $\mathbf{x}_t^* = \mathbf{x}_d^*$, $\mathbf{u}_t^* = \mathbf{u}_d^*$
10:     exit: set termination = true
11:   else
12:     calculate a candidate $\mathbf{x}_t^*(k+1)$ based on (6.16) and $\mathbf{u}_t^*(k+1)$ by (6.10)
13:     if $V_{\mathbf{x}_t^*}(k+1) \leq \eta_{\mathbf{u}_t^*}(k+1)$ and $\mathbf{x}_t^*(k+1)$ closer at $\mathbf{x}_d^*$ than $\mathbf{x}_t^*$ = true then
14:       set $\mathbf{x}_t^* = \mathbf{x}_t^*(k+1)$
15:     else
16:       reduce $\eta_{\mathbf{u}_t^*}(k+1) = \iota\eta_{\mathbf{u}_t^*}(k+1)$
17:       if $V_{\mathbf{x}_t^*} > \eta_{\mathbf{u}_t^*}$ = true then
18:         recalculate $\mathbf{x}_r^*$ by (6.8) or Section 6.4.1 for $\mathbf{x}$
19:         set $\mathbf{x}_t^* = \mathbf{x}_r^*$, $\mathbf{u}_t^* = \mathbf{u}_r^*$ by (6.10)
20:         set $\mathbf{x}_t^*(k+1) = \mathbf{x}_t^*$, $\mathbf{u}_t^*(k+1) = \mathbf{u}_t^*$
21:         set $\eta_{\mathbf{u}_t^*}(k+1) = \eta_{\mathbf{u}_t^*}(k+1)$
```

Initially (line 1 until 5), we calculate a reference equilibrium that can be stabilized $\mathbf{x}_0 \in \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$ and consider it as temporary equilibrium. We introduce an iteration (discretization) index k and the variable $\mathbf{x}_t^*(k)$ (and the corresponding $\mathbf{u}_t^*(k)$) which denotes a candidate for the temporary equilibrium, meaning an equilibrium that might act as updated $\mathbf{x}_t^*$ if stability is ensured. Both are initially set equal to the values of the temporary equilibrium (line 5). The remainder of the algorithm (line 6 until 21) is executed in real-time, whereby the objective is to derive a $\mathbf{x}_t^*(k)$, to check its stability, to update $\mathbf{x}_t^*$ and to iteratively optimize the candidate $\mathbf{x}_t^*(k) \rightarrow \mathbf{x}_t^*(k+1)$: Due to line 7 the iteration of $\mathbf{x}_t^*(k)$ is repeated multiple times consecutively in a single time step until termination is forced, e.g. due to real-time execution constraints. At the beginning of each iteration it is checked whether the desired equilibrium can already be activated (analogues to Algorithm 6.3.1). If not, we calculate a new candidate for the temporary equilibrium $\mathbf{x}_t^*(k+1)$ (line 12 of Algorithm 6.4.1) by

$$c(k+1) = c(k) + \Delta(k), \tag{6.16a}$$

$$\mathbf{x}_t^*(k+1) = \hat{\mathbf{f}}\left(\mathbf{x}_r^* + c(k+1)\left(\mathbf{x}_d^* - \mathbf{x}_r^*\right)\right) \tag{6.16b}$$

whereby the parameter $\Delta(k)$ is used to adapt the scaling variable $c(k)$ of $\mathbf{x}_t^*(k)$ from one iteration step to the next along the function of allowed set points (6.5a). In other words, $c(k)$ can be seen as the scaling variable c in (6.7), (6.9). For imitating the optimal operation principle of Algorithm 6.3.1, we would like to set $c(k)$ such that the current state variable is at the bounding level set, meaning

$$\eta_{\mathbf{u}_t^*}(k) - V_{\mathbf{x}_{t,c}^*}(k) = 0. \tag{6.17}$$

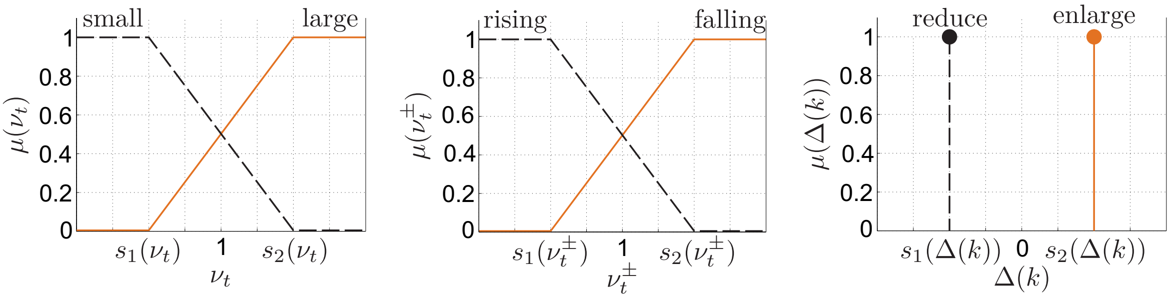
To that aim, we define the ratios

$$\nu_t^{\pm}(k) = \frac{\nu_t^+(k)}{\nu_t^-(k)}, \quad \nu_t(k) = \frac{V_{\mathbf{x}_t^*}(k)}{\eta_{\mathbf{u}_t^*}(k)}, \tag{6.18}$$

with $\nu_t^-(k) = \frac{V_{\mathbf{x}_t^{*-}(k)}}{\eta_{\mathbf{u}_t^{*-}(k)}}$ and $\nu_t^+(k) = \frac{V_{\mathbf{x}_t^{*+}(k)}}{\eta_{\mathbf{u}_t^{*+}(k)}}$. Thereby, $\mathbf{x}_t^{*-}(k)$ and $\mathbf{x}_t^{*+}(k)$ denote temporary equilibria that are an infinitesimal step ($\delta << 1$) closer at $\mathbf{x}_r^*$ and $\mathbf{x}_d^*$, respectively, than the current candidate $\mathbf{x}_t^*(k)$. In other words, these two set points are reached when moving from $\mathbf{x}_t^*(k)$ a little backward and forwards along (6.5a). Hence, the ratio $\nu_t^{\pm}(k)$ states if the Lyapunov function increases or decrease in the direction of $\mathbf{x}_d^*$, i.e. the gradient of the Lyapunov function at $\mathbf{x}_t^*(k)$ (increasing if $\nu_t^{\pm}(k) > 1$ and decreasing if $\nu_t^{\pm}(k) < 1$). The second ratio $\nu_t(k)$ states if the current candidate for the temporary equilibrium can be stabilized ($\nu_t(k) \leq 1$) or not. Based on these two ratios, two important information can be gained: First, is the current candidate $\mathbf{x}_t^*(k)$ a valid temporary equilibrium and second, in what direction along (6.5a) do we have to shift $\mathbf{x}_t^*(k)$ in order to fulfill (6.17). As we will shift $\mathbf{x}_t^*(k)$ by the parameter $c(k)$, which is adapted by $\Delta(k)$ in (6.16a), we exploit the gathered information for deriving the fuzzy rule base in Table 6.1. For instance, the first rule is: "If $\nu_t(k)$ is small and $\nu_t^{\pm}(k)$ is rising then enlarge $\Delta(k)$". Note, although the rule base is not recurrent according to Section 2.1.1, the updating law (6.16) is. This is due to the fact that the rule base in Table 6.1 implicitly depends on the prior candidate $\mathbf{x}_t^*(k)$. For implementing the fuzzy rule base, we consider trapezoidal membership functions for fuzzification and singletons for defuzzification as shown in Fig. 6.5 and explained in Section 2.1.1.

Table 6.1: Basic fuzzy rule base for updating $c(k)$

$[\nu_t(k), \nu_t^{\pm}(k)]$	[small, rising]	[large, rising]	[small, falling]	[large, falling]
$\Delta(k)$	enlarge	reduce	reduce	enlarge



(a) Fuzzification $\nu_{t,c}(k)$ (b) Fuzzification $\nu_{t,c}^\pm(k)$ (c) Defuzzification $\Delta(k)$

Figure 6.5: Membership functions of the rule base in Table 6.1.

The new candidate is activated as temporary equilibrium $\mathbf{x}_t^* = \mathbf{x}_t^*(k)$ if it can be stabilized $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_t^*(k)}(\mathbf{P}, \eta_{\mathbf{u}_t^*}(k))$ and it is closer at $\mathbf{x}_d^*$ than the currently active temporary equilibrium (line 13). Otherwise the temporary equilibrium is kept constant at its actual value and the bounding level value is reduced by the scalar $\iota < 1$, $\iota \approx 1$, in order to reduce the iteration step for the subsequent calculation of a new candidate $\mathbf{x}_t^*(k+1)$. In other words, when moving from $\mathbf{x}_t^*$ towards $\mathbf{x}_d^*$ along (6.5a) than $\mathbf{x}_{t,c}^*(k)$ should be past. If the temporary equilibrium is not updated we check if it still can be stabilized. In that case the iteration is aborted. Otherwise stabilization is not ensured, e.g. due to external disturbances or measurement noise. For robustness reasons (analogous to Algorithm 6.3.1), a new valid reference equilibrium has to be calculated (line 18) subject to the current state vector $\mathbf{x}$, which analogous to the initialization part is considered as new temporary equilibrium. Note, the lines 17 and 21 are only executed in the unlike event that no valid candidate $\mathbf{x}_t^*(k)$ can be found.

Concerning Algorithm 6.4.1 for a recurrent updating of the temporary equilibrium we state the following stability results:

Theorem 6.4.3. *Let a system* (6.1) *in T-S form and a corresponding T-S controller* $\mathbf{u} = \mathbf{u}_d^* + \sum_{j=1}^r h_j(\mathbf{z}_c)\left(\mathbf{F}_j(\mathbf{x} - \mathbf{x}_d^*)\right)$ *that asymptotically stabilizes* $\mathbf{x}_d^*$ $\forall$ $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$ *be given. Then the GINA controller extended closed-loop system asymptotically stabilizes* $\mathbf{x}_d^*$ *for all* $\mathbf{x} \in \bigcup_{(\forall\, \mathbf{x}_r^*)} \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$ *when updating* $\mathbf{x}_t^*$ *by Algorithm 6.4.1 if an equilibrium function* (6.5a) *exists which fulfills* (6.11)*.*

Proof: See Appendix A.3. ■

In case that (6.5a) does not fulfill the condition (6.11), the desired equilibrium $\mathbf{x}_d^*$ is attractive which is proven by the following Corollary:

Corollary 6.4.3. *Let a system* (6.1) *in T-S form and a corresponding T-S controller* $\mathbf{u} = \mathbf{u}_d^* + \sum_{j=1}^r h_j(\mathbf{z}_c)\left(\mathbf{F}_j(\mathbf{x} - \mathbf{x}_d^*)\right)$ *that asymptotically stabilizes* $\mathbf{x}_d^*$ $\forall$ $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$

be given. Then $\mathbf{x}_d^*$ is attractive with the GINA controller extended closed-loop system for all $\mathbf{x} \in \bigcup_{(\forall\, \mathbf{x}_r^*)} \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$ when updating $\mathbf{x}_t^*$ by Algorithm 6.4.1 if a function (6.5a) *exists which does not fulfill* (6.11).

Proof: Theorem 6.4.3 shows that the value of the Lyapunov-like function (A.1) does not necessarily decrease and might even increase if condition in (6.11) is not guaranteed. However, Algorithm 6.4.1 ensues that $\mathbf{x}_t^* = \mathbf{x}_d^*$ (for $t \to \infty$ at least) whereby the asymptotic stability of $\mathbf{x}_d^*$ follows from Case 1 of the proof of Theorem 6.3.1. Consequently, $\mathbf{x}_d^*$ is attractive which concludes the proof. ■

6.5 An Unified Iteration Algorithm

In this section, we develop a variation of the GINA controller that is applicable to both functional relations mentioned in Assumption 6.1.1. Hence, in comparison to the prior sections of this chapter numerically calculated steady-state curves can be handled in addition to smooth piecewise analytical functions. In addition, we show how most of the required calculations can be done prior to the real-time execution such that the required real-time computation is reduced to a selection of a valid solution.

The key point is a curve of discrete allowed equilibria (and thus a function (6.5b)) and corresponding input signals. Either by discretizing a known analytical function or having a numerically calculated steady-state curve available (see Assumption 6.1.1). For each discrete equilibrium point $\mathbf{x}_l^*$, with $l \in \mathbb{N}_{1:o}$ the steady-state input $\mathbf{u}_l^*$ and the domain of attraction (DA) is calculated, e.g. based on the methods of Section 3. Each $\mathbf{x}_l^*$ can either act as reference, temporary or desired set point. The objective is to shift the temporary equilibrium in discrete steps along $\mathbf{x}_l^*$ towards $\mathbf{x}_d^*$ such that stability is maintained. For the latter, we state the following assumption:

Assumption 6.5.1. An estimated DA $\mathcal{X}_{\mathbf{x}_l^*}(\mathbf{P}, \eta_{\mathbf{u}_l^*})$ of a discrete set point $\mathbf{x}_l^*$ includes the neighboring ones along the discrete equilibria curve. These set points are merged in a set $\mathcal{S}_{\mathbf{x}_l^*}$.

This assumption is not a restriction as it can always be fulfilled in an iterative two-step procedure: First, we try to estimate a DA of a set point $\mathbf{x}_l^*$ such that the surrounding ones are included. This can be done in a constructively manner by the following proposition:

Proposition 6.5.1. *A domain of attraction $\mathcal{X}_{\mathbf{x}_l^*}(\mathbf{P}, \eta_{\mathbf{u}_l^*})$ of a discrete set point $\mathbf{x}_l^*$ includes the neighboring ones along the discrete equilibria curve, meaning $\mathcal{S}_{\mathbf{x}_l^*} \subset \mathcal{X}_{\mathbf{x}_l^*}(\mathbf{P}, \eta_{\mathbf{u}_l^*})$*

if the LMI condition

$$\begin{bmatrix} 1 & \left(\mathbf{x}_{l,i}^{*}\right)^{T} \\ \mathbf{x}_{l,i}^{*} & \mathbf{Q} \end{bmatrix} \geq \mathbf{0}, \tag{6.19}$$

with $\mathbf{Q} = \left(\frac{\mathbf{P}}{\eta_{\mathbf{u}_l^*}}\right)^{-1}$*, is fulfilled for all* $\mathbf{x}_{l,i}^{*} \in \mathcal{S}_{\mathbf{x}_l^*}$*.*

Proof: The equilibria $\mathbf{x}_{l,i}^{*} \in \mathcal{S}_{\mathbf{x}_l^*}$ are contained in $\mathcal{X}_{\mathbf{x}_l^*}(\mathbf{P}, \eta_{\mathbf{u}_l^*})$ if

$$\left(\mathbf{x}_{l,i}^{*}\right)^{T} \mathbf{P} \left(\mathbf{x}_{l,i}^{*}\right) \leq \eta_{\mathbf{u}_l^*}, \quad \forall\, \mathbf{x}_{l,i}^{*} \in \mathcal{S}_{\mathbf{x}_l^*}. \tag{6.20}$$

That can be rewritten based on (2.41) to

$$\begin{bmatrix} \eta_{\mathbf{u}_l^*} & (\mathbf{x}_{l,i}^{*})^{T}\mathbf{P} \\ \mathbf{P}\mathbf{x}_{l,i}^{*} & \mathbf{P} \end{bmatrix} \geq \mathbf{0}. \tag{6.21}$$

Multiplying it form both sides with the regular, symmetric matrix

$$\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix}, \tag{6.22}$$

where $\mathbf{Q} = \left(\frac{\mathbf{P}}{\eta_{\mathbf{u}_l^*}}\right)^{-1}$ results in

$$\eta_{\mathbf{u}_l^*} \begin{bmatrix} 1 & \left(\mathbf{x}_{l,i}^{*}\right)^{T} \\ \mathbf{x}_{l,i}^{*} & \mathbf{Q} \end{bmatrix} \geq \mathbf{0}, \tag{6.23}$$

which concludes the proof as $\eta_{\mathbf{u}_l^*} > 0$. ■

The LMIs (6.19) (or their equivalent formulation (6.21)) can be easily added as constraints when estimating $\mathcal{X}_{\mathbf{x}_l^*}(\mathbf{P}, \eta_{\mathbf{u}_l^*})$ based on convex optimization. In the second step, the amount of discrete set points has to be enlarged if Proposition 6.5.1 can not be fulfilled for every equilibria $\mathbf{x}_l^*$. This means if $\mathcal{S}_{\mathbf{x}_l^*} \not\subset \mathcal{X}_{\mathbf{x}_l^*}(\mathbf{P}, \eta_{\mathbf{u}_l^*})$ then a set point needs to be added on each side of $\mathbf{x}_l^*$ along the equilibria curve. Repeating these two steps (estimating the DA and increasing the number of set points) iteratively leads to a number of equilibria that finally fulfills Assumption 6.5.1.

A unified procedure for iteratively updating the temporary equilibrium is summarized in Algorithm 6.5.1. From the conceptual point of view it is strongly linked to Algorithm 6.4.1 and will be explained in the following: In the initialization part a curve of discrete equilibria according to Assumption 6.5.1 is derived. From that stored curve a reference equilibrium is selected as initial temporary equilibrium $\mathbf{x}_t^*(0)$, that can be

Algorithm 6.5.1 Unified, iteratively updating of $\mathbf{x}_t^*$

```
 1: Initialization:
 2: curve of discrete equilibria $\mathbf{x}_l^*$: discretization according to Assumption 6.5.1
 3: select $\mathbf{x}_r^*$ s. t. $\mathbf{x}_0 \in \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$
 4: set $\mathbf{x}_t^* = \mathbf{x}_r^*$, $\mathbf{u}_t^* = \mathbf{u}_r^*$
 5: set $k = 0$, $\mathbf{x}_t^*(k) = \mathbf{x}_t^*$, $\mathbf{u}_t^*(k) = \mathbf{u}_t^*$
 6: Real-Time:
 7: while termination = false do
 8:   if $V_{\mathbf{x}_d^*} \leq \eta_{\mathbf{u}_d^*}$ = true then
 9:     $\mathbf{x}_t^* = \mathbf{x}_d^*$, $\mathbf{u}_t^* = \mathbf{u}_d^*$
10:     exit: set termination = true
11:   else
12:     set $\mathbf{x}_t^*(k+1) = \mathbf{x}_t^+(k)$, calculate $\mathbf{u}_t^*(k+1)$
13:     if $V_{\mathbf{x}_t^*}(k+1) \leq \eta_{\mathbf{u}_t^*}(k+1)$ = true then
14:       $\mathbf{x}_t^* = \mathbf{x}_t^*(k+1)$
15:     else
16:       if $V_{\mathbf{x}_t^*}(k) \leq \eta_{\mathbf{u}_t^*}(k)$ = false then
17:         reselect $\mathbf{x}_r^*$ s. t. $\mathbf{x} \in \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$
18:         set $\mathbf{x}_t^*(k) = \mathbf{x}_t^* = \mathbf{x}_r^*$, $\mathbf{u}_t^*(k) = \mathbf{u}_t^* = \mathbf{u}_r^*$
19:       else
20:         exit: set termination = true
```

stabilized concerning the initial state vector $\mathbf{x}_0$. That operation task (selecting the reference equilibrium in line 3 and also in line 18) can be solved in several ways. Either by applying one of the strategies described in the previous sections and selecting the closest $\mathbf{x}_l^*$ to the calculated $\mathbf{x}_r^*$. Or based on the Appendix B. Analogous to Algorithm 6.4.1 a discretization index k and the variables $\mathbf{x}_t^*(k)$, $\mathbf{u}_t^*(k)$ for a candidate concerning an update of the temporary equilibrium are introduced. Both are initially set equal to the values of the stabilizable reference equilibrium. The rest of the algorithm (lines 7 to 20) is executed in real-time. Note that due to line 7 the real-time part is not only executed once per sample but rather as often as possible until termination is required, e.g. due to real-time execution constraints. The objective thereby is to iteratively update the temporary equilibrium. At the beginning of the iterative real-time part it is checked whether the desired equilibrium can already be activated or not. If $\mathbf{x} \notin \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$ the iteration becomes active. We set the candidate equal to the next discrete equilibrium that is closer to the desired one $\mathbf{x}_t^*(k+1) = \mathbf{x}_t^+(k)$. If the candidate can be stabilized (line 13, 14) it is activated by updating the temporary equilibrium $\mathbf{x}_t^* = \mathbf{x}_t^*(k+1)$. Otherwise the prior value is kept and it is checked if it still can still be stabilized. If this is true the iteration is aborted as no other temporary equilibrium can be stabilized at this time. If it fails, a new valid reference equilibrium has to be selected (line 17), subject to the current state vector $\mathbf{x}$. The reference equilibrium is again (analogous to the initialization) considered as new temporary set point. If the execution of the real-time part is terminated the latest valid equilibrium candidate is considered as sub-optimal solution. Du to the computationally not intensive "IF-THEN"-operation, at least a single iteration should be always feasible.

Concerning that unified updating procedure we state the following Theorem for ensuring the asymptotic stability of the desired equilibrium:

Theorem 6.5.1. *Let a system* (6.1) *in T-S form and a corresponding T-S controller* $\mathbf{u} = \mathbf{u}_d^* + \sum_{j=1}^{r} h_j(\mathbf{z}_c)\left(\mathbf{F}_j(\mathbf{x} - \mathbf{x}_d^*)\right)$ *that asymptotically stabilizes* $\mathbf{x}_d^*\ \forall\ \mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$ *be given. Then the GINA controller extended closed-loop system asymptotically stabilizes* $\mathbf{x}_d^*$ *for all* $\mathbf{x} \in \bigcup_{(\forall\, \mathbf{x}_r^*)} \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$ *when updating* $\mathbf{x}_t^*$ *by Algorithm 6.5.1 if a curve of discrete allowed equilibria* (6.5b) *exists that fulfills Assumption 6.5.1 and*

$$|\mathbf{x}_d^* - \mathbf{x}_t^*(k)| - |\mathbf{x}_d^* - \mathbf{x}_t^*(k+1)| > 0 \tag{6.24}$$

holds for all $c(k) \in [0, 1[$.

Proof: Following the argumentation of A.3 the asymptotic stabilization is guaranteed if the temporary equilibrium can be iteratively updated. Assumption 6.5.1 ensures the existence of a set point that enables an updating at least when $\mathbf{x} = \mathbf{x}_t^*(k)$. Thereby, the proof is concluded. ∎

Even in case that a discrete allowed equilibrium (6.5b) exists which does not fulfill (6.24), the attraction of $\mathbf{x}_d^*$ is ensured at least by Theorem 6.5.1:

Corollary 6.5.1. *Let a system* (6.1) *in T-S form and a corresponding T-S controller* $\mathbf{u} = \mathbf{u}_d^* + \sum_{j=1}^{r} h_j(\mathbf{z}_c)\left(\mathbf{F}_j(\mathbf{x} - \mathbf{x}_d^*)\right)$ *that asymptotically stabilizes* $\mathbf{x}_d^*\ \forall\ \mathbf{x} \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$ *be given. Then the GINA controller extended closed-loop system stabilizes* $\mathbf{x}_d^*$ *for all* $\mathbf{x} \in \bigcup_{(\forall\, \mathbf{x}_r^*)} \mathcal{X}_{\mathbf{x}_r^*}(\mathbf{P}, \eta_{\mathbf{u}_r^*})$ *for* $t \to \infty$ *(attraction of* $\mathbf{x}_d^*$*) when updating* $\mathbf{x}_t^*$ *by Algorithm 6.5.1 if a curve* (6.5b) *exists that fulfills Assumption 6.5.1.*

Proof: The discrete scaling factor $c(k) \in [0, 1[$ in (6.5b) will either increase or stay constant due to the key take-away $\dot{c}(k) \geq 0$ according to Case 2 in the proof of Theorem 6.5.1. However, the value of the Lyapunov-like function (A.1) does not necessarily decrease as condition (6.24) is not guaranteed for all relevant $c \in [0, 1[$ due to (A.5). Based on Theorem 6.5.1 it is ensured that $\dot{c}(k) > 0$ at least every time when updating the temporary equilibrium whereby $c(k) = 1$ is guaranteed for $t \to \infty$ at least. As soon as $c(k) = 1$, the GINA controller extended closed-loop system asymptotically stabilizes $\mathbf{x}_\mathbf{d}^*$ by Case 1. Consequently, $\mathbf{x}_d^*$ is attractive which concludes the proof. ∎

6.6 Summary

The main contribution of this chapter has been the novel *GINA controller* (Governor Integrated Nominal-Value Adaptation) which is, up to the author's knowledge, the first reference governor for constrained dynamical systems that exploits the T-S framework. By modifying the set point, the GINA controller enables a stabilization of a desired equilibrium even if the current state vector is not within its estimated domain of attraction (DA). Roughly speaking, its basic principle can be summarized with: Derive an equilibrium which can guaranteed to be stabilized subject to input amplitude, rate and state restrictions. Then shift this equilibrium as fast as possible towards the desired one such that stability is maintained. Thereby, arbitrary set point changes can be realized. We have introduced several algorithms for the practical implementation which is the key step to enable a general applicability. Thereby, we have addressed and solved the following problems: computational effort, optimality of the governor solution, discrete and continuous realization as well as robustness against external disturbances and measurement noise.

Chapter 7

Application and Experimental Results

In this chapter, we demonstrate the practical relevance of the developed methods by four technical examples subject to input amplitude, rate and state constraints. We present both, simulation and experimental results. We start in Section 7.1 with a commonly known benchmark system – inverted pendulum on cart. Subsequently, we recall the Ballbot system of Example 6.4.1. This robot is a complete in-house development (starting from the idea up to its construction) that we built up together with some colleagues which is gratefully acknowledged. In the fourth example, we leave the area of robotics and step over to the field of ground vehicles. We apply our methods to enhance the ride comfort of a simple active cruise control (ACC) which is already an available feature in modern cars. Concerning all examples, the nominal level value is set to $\eta_\mathbf{0} = 1$ and required LMIs are solved by using the YALMIP Toolbox.

7.1 Inverted Pendulum

We experimentally verify our methods based on the nonlinear inverted pendulum on cart system. The equations of motion and our test rig are explained in Section 7.1.1. Thereafter, we investigate three different experimental setups in Section 7.1.2: In the first experiment, a change of the desired set point is focused and the GINA controller is iteratively updated according to Algorithm 6.4.1. Thereafter, we illustrate the robustness and the analytically updating procedure of the GINA controller (Algorithm 6.3.1) concerning a large non-equilibrium initial error of the state vector. Finally, we show a multiple set point change including a continuously moving one (moving target). To this end, we apply a switched controller architecture based on non-nested DAs and recurrent fuzzy systems of the Sections 4.3 and 5.3, respectively. Also here the GINA controller is added to ensure asymptotic stability in the large.

7.1.1 Modeling and Test Rig

The model of the pendulum on cart system is derived within the Lagrangian mechanics framework. The considered schematics of the pendulum system and our test rig are depicted in Fig. 7.1(a) and Fig. 7.1(b), respectively. The cart's position is given by x and ϕ represents the pendulum's tilt angle. The final nonlinear equations of motion are

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \theta_1\theta_2\theta_3 3m_p g & 0 & -\theta_1 4d_x & -\theta_1\theta_4 4m_p a \\ \theta_1\theta_2\frac{3m_g g}{a} & 0 & -\theta_1\theta_3\frac{3d_x}{a} & -\theta_1\theta_3\theta_4 3m_p \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ \theta_1 4c_m \\ \theta_1 f_3 \frac{3}{a} \end{bmatrix} \sigma(u, \dot{u}) \tag{7.1}$$

whereby the state vector is $\mathbf{x} = [\phi, x, \dot{x}, \dot{\phi}]^T$. The system's nonlinearities are

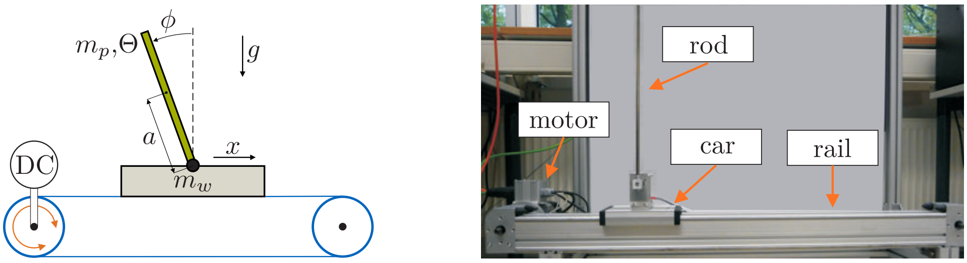
$$\begin{aligned} \theta_1 &= \frac{1}{4m_g - 3m_p \cos(\phi)^2}, & \theta_2 &= \operatorname{sinc}(\phi) \left(= \frac{\sin(\phi)}{\phi} \right) \\ \theta_3 &= \cos(\phi), & \theta_4 &= \dot{\phi}\sin(\phi) \end{aligned} \tag{7.2}$$

and the saturated input u equals to (3.9) (input amplitude and rate saturation) having a dynamic according to (3.10). All relevant system parameters are summarized in Table C.1 of Appendix C.

Due to the four nonlinearities, a SE-NL T-S model (3.11) consists of $r = 16$ linear subsystems with a common input vector $\hat{\mathbf{b}} \in \mathbb{R}^5$ and the augmented state vector $\hat{\mathbf{x}} = [\mathbf{x}, u]^T$ (see (3.12)). The corresponding matrices are given in (C.1). We set the bounds of the universe of discourse (UoD) (2.9) to

$$\phi_{max} = 0.44\,\text{rad}\ (= 25°), \quad \dot{\phi}_{max} = 2\pi\,\frac{\text{rad}}{\text{s}} \tag{7.3}$$

which allow to experimentally highlight the advantages of the GINA controller on our



(a) Schematic

(b) Test rig

Figure 7.1: Inverted pendulum on cart system.

test rig while input amplitude and rate saturation occur. Based on the obtained model, we can state the following facts concerning the design of the GINA controller: The SE-NL T-S model fulfills the rank conditions (6.12). Hence, according to the proof of Lemma 6.3.2 the bounding level value of the estimated DA of each equilibrium is constant and the steady-state input signal vanishes. Additionally, the required reference set point $\hat{\mathbf{x}}_r^*$ is analytically given by (6.13) according to Theorem 6.4.1. The matrix $\mathbf{N}$, which spans the subspace of allowed equilibria, becomes the constant vector $\mathbf{n} = [0, 1, 0, 0, 0]^T$ (only the cart's position coordinate x is allowed to differ from zero at an equilibrium) and thus the scaling vector $\mathbf{z}$ becomes a scalar z. As each location of the cart is a set point, a linear relation (6.14) can be considered as equilibrium functions (6.5a) and (6.10b).

7.1.2 Results

Set point tracking task: First, we consider a set point tracking task from a desired position $x_{d,1}^* = -0.2\,\mathrm{m}$ towards the second one $x_{d,2}^* = 0.45\,\mathrm{m}$ (all other state variables are zero in both equilibria). We design a LQR controller [73]

$$J = \int_0^\infty \hat{\mathbf{x}}^T \mathbf{Q}_{LQR} \hat{\mathbf{x}} + v R_{LQR} v \;\mathrm{dt}, \tag{7.4}$$

with $\mathbf{Q}_{LQR} = \mathrm{diag}[51467, 71162, 0, 0]$ and $R_{LQR} = 10$ for all $r = 16$ subsystems of the SE-NL T-S model of the pendulum system. We estimate a DA subject to the UoD by the first part of Algorithm 3.2.1 while claiming an exponential decay rate of $\alpha = 1.4$. The temporary equilibrium of the GINA controller is iteratively updated by the implicit recurrent fuzzy interpolation according to Algorithm 6.4.1. The core positions of the membership functions (see Fig. 6.5) and the value for reducing the bounding level set are chosen equivalent to [31]:

$$[s_1, s_2] = [0.8, 1.2], \quad [\mathrm{reduce}, \mathrm{enlarge}] = [-0.5, 0.5], \quad \iota = 0.99. \tag{7.5}$$

Fig. 7.2 shows the obtained experimental results concerning a fixed maximum input amplitude of $u_{max} = 16\,\mathrm{V}$ and a rate saturation of $\dot{u}_{max} = 1000\,\mathrm{V/s}$ (both values are within the allowed ranges of Table C.1): The GINA controller works properly and ensures a fast and asymptotic stabilization of the desired set point. Both, the input amplitude and rate saturate and the UoD (7.3) are not violated. The shown set point tracking task destabilizes the pendulum system due to saturation effects if the GINA controller is deactivated. Hence, the asymptotic stability is guaranteed by iteratively

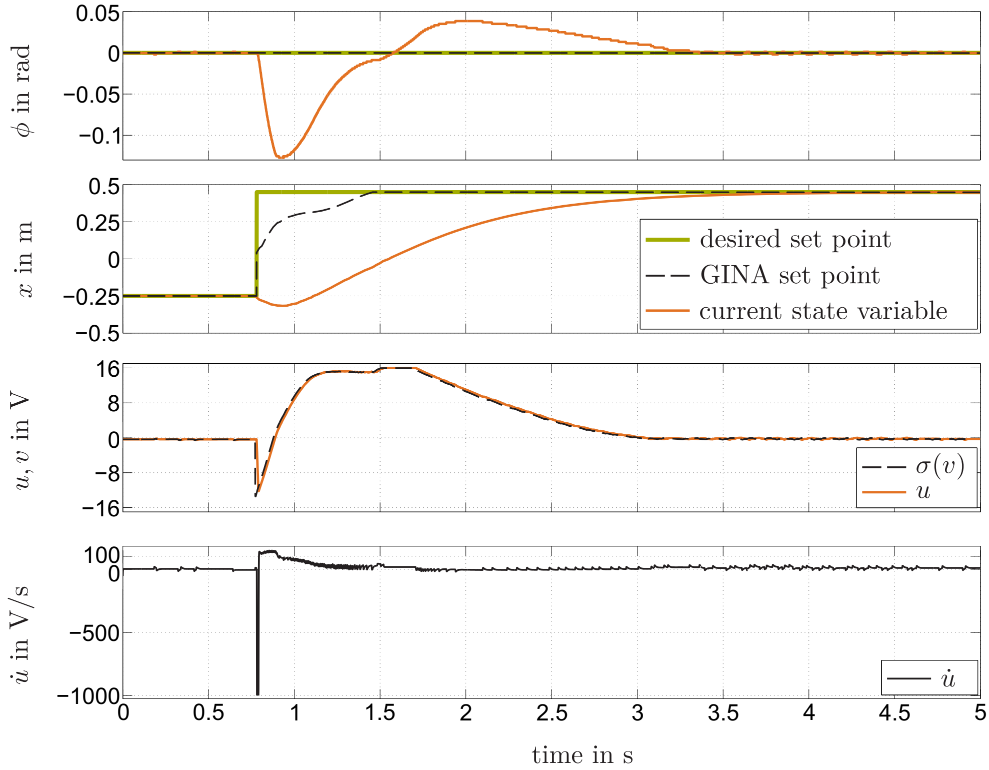


Figure 7.2: Experiment I: Set point change of the pendulum with GINA controller.

updating the set point to be tracked. The temporary equilibrium $\mathbf{x}_t^*$ (called GINA set point in Fig 7.2) is successively shifted towards the desired one $\mathbf{x}_d^*$. Both become equivalent at a time of $t \approx 1.5\,\mathrm{s}$.

Initial error compensation: In the second experimental setup, we consider an initial disturbance (non-equilibrium state vector) of $\mathbf{x}_0 = [0.26, 0, 0, 1.1, 0]^T$. The objective is to stabilize the set point $\mathbf{x}_d^* = [0, 0.5, 0, 0, 0]^T$. The T-S controller according to the prior example is used and the DA is estimated by Algorithm 3.2.1. The initial error is created by stabilizing the pendulum at $\mathbf{x}^* = [0, -0.1, 0, 0, 0]^T$. Then the T-S controller based on (7.4) is deactivated while simultaneously moving the cart to $x = 0\,\mathrm{m}$. The controller as well as the desired set point $x_{d,2}^*$ are activated when the pendulum's tilt angle reaches $\phi = 0.26\,\mathrm{rad}$. The temporary equilibrium of the GINA controller is optimally calculated according to Algorithm 6.3.1. Fig. 7.3 shows the obtained experimental results concerning a maximum input amplitude of $u_{max} = 33\,\mathrm{V}$ and a rate limitation of $\dot{u}_{max} = 1000\,\mathrm{V/s}$. First, the GINA controller calculates a reference equilibrium $\mathbf{x}_r^*$ that can be stabilized. Due to the pendulums's tilt angel the reference equilibrium is placed in the opposite direction of the desired set point. While stabilizing $\mathbf{x}_r^*$, the temporary equilibrium $\mathbf{x}_t^*$ (called GINA set point in Fig 7.3)

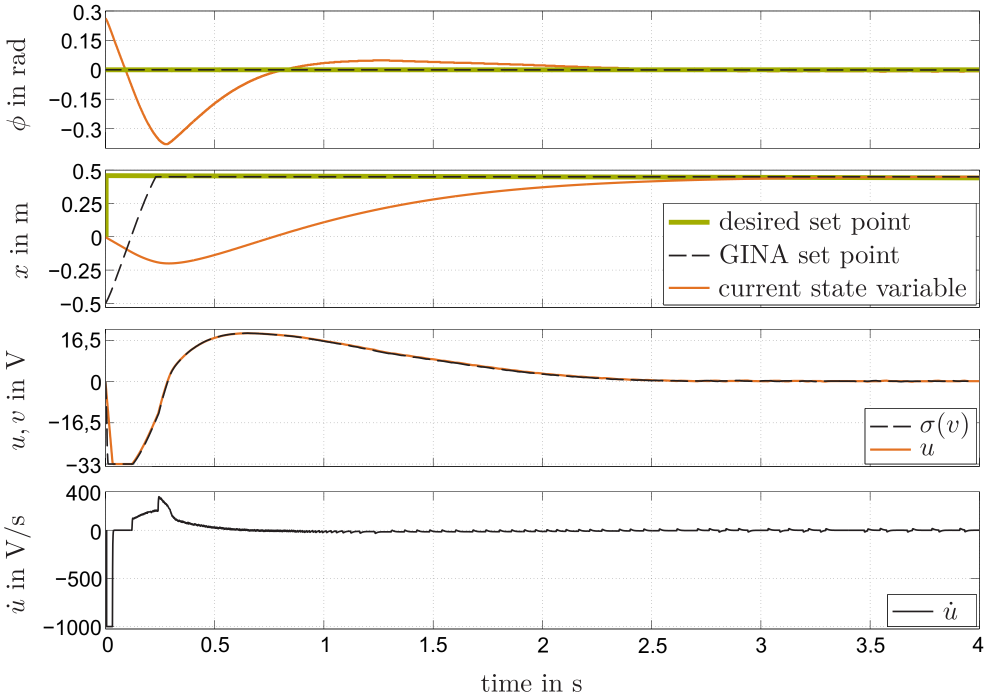


Figure 7.3: Experiment II: Set point tracking of the pendulum system with GINA controller and an initial error.

is continuously shifted from $\mathbf{x}_r^*$ towards $\mathbf{x}_d^*$. The input signals saturate at the beginning of the experiment which illustrates that the input signal is exploited for the considered stabilization task. Due to this, it becomes quite obvious that the experiment becomes unstable if the GINA controller is disabled whereby its effectiveness is confirmed.

Switched controller design: In the third experimental set up, we show a multiple set point change subject to input amplitude saturation $u_{max} = 12\,\mathrm{V}$ based on a switched controller architecture with multiple Lyapunov functions. Three LQR controllers

$$J = \int_0^\infty \mathbf{x}^T \mathbf{Q}_{LQR,i}\mathbf{x} + uR_{LQR}u \;\mathrm{d}t, \tag{7.6}$$

with the weighting matrices $\mathbf{Q}_{LQR,i} = \mathbf{Q}_{LQR,i}^T \geq \mathbf{0}$ and $R_{LQR} > 0$, $i \in \mathbb{N}_{1:3}$, are considered. The weights are presented in Table 7.1. In addition, a LQR with $\check{\mathbf{Q}}_{LQR,4}$ is designed for the augmented system $\check{\mathbf{x}} = [\mathbf{x}, \int x\; dt]$. As no common Lyapunov function for all of the closed-loop T-S system could have been found based on Algorithm 3.2.1, we apply our switching strategy with non-nested DAs (see Section 5.2).

First, we optimize two common Lyapunov functions $V_1 = \mathbf{x}^T\mathbf{P}_1\mathbf{x}$ and $V_2 = \mathbf{x}^T\mathbf{P}_1\mathbf{x}$ (and related DAs) for the closed-loop systems Σ^i, $i \in \mathbb{N}_{1:2}$ and Σ^i, $i \in \mathbb{N}_{2:3}$, respectively. Consequently, a smooth blending between the controllers within each DA is allowed by

Table 7.1: Controller weights $\mathbf{Q}_{LQR,i}$, $R_{LQR} = 10$ for all controllers

$\mathbf{Q}_i$	value	description
$\mathbf{Q}_{LQR,1}$	$\mathrm{diag}[51470, 71160, 0, 0]$	position error oriented
$\mathbf{Q}_{LQR,2}$	$\mathrm{diag}[67680, 24770, 0, 0]$	angle error oriented
$\mathbf{Q}_{LQR,3}$	$\mathrm{diag}[60125, 6720, 0, 0]$	balanced controller
$\check{\mathbf{Q}}_{LQR,4}$	$\mathrm{diag}[2525, 1400, 0, 0, 80800]$	integrator enhanced controller

Corollary 5.2.1. A smooth switching between V_1 and V_2 is realized by (5.1) whereby Σ^1 is equivalent to the blending controller (5.2). In other words, a switching from the closed-loop system Σ^1 towards Σ^3 is only allowed by activating Σ^2 first according to the multi Lyapunov Theorem 5.2.1. Second, a common Lyapunov function $V_3 = \check{\mathbf{x}}^T \mathbf{P}_3 \check{\mathbf{x}}$ has been found for $\check{\Sigma}^4$ and $\check{\Sigma}^1$. The latter matrix is obtained by extending the closed-loop matrix of Σ^1 with the row $\mathbf{h}^T = [0, 1, 0, 0]$ and a zero column at the end. Hence, Σ^1 belongs to the state vector $\check{\mathbf{x}}$. The dynamics of $\check{\Sigma}^1$ and Σ^1 are equivalent, whereby a hard switching between them does not cause discontinuities in the active vector field. As a consequence from that design, smooth switching from a closed-loop system within V_1 towards $\check{\Sigma}^1$ (V_3) can be done by the multi Lyapunov Theorem 5.2.1. For formulating the rule base of the switched RFS according to Section 5.3, we consider the absolute values of the desired position's derivative, $\tilde{u}_1 = |\dot{x}_d^*|$, the derivative of the position error, $\tilde{u}_2 = |\dot{x}_d^* - \dot{x}|$, and the position error $\tilde{u}_3 = |x_d^* - x|$ as elements of the input vector $\tilde{\mathbf{u}}$. Their linguistical characteristics are chosen as

$$\mathbf{L}_{\mathbf{j}}^{\tilde{u}_1} = \mathbf{L}_{\mathbf{j}}^{\tilde{u}_2} = \{\text{small, large}\}, \quad \mathbf{L}_{\mathbf{j}}^{\tilde{u}_3} = \{\text{small, medium, large}\}. \tag{7.7}$$

The switching rules are non-strict and thus trapezoid-shaped membership functions according to Definition 5.3.1 are used. Their core positions are set to

$$\mathbf{s}_{\mathbf{L_{j,i}}}^{\tilde{u}_1} = [0.25, 0.5][\frac{m}{s}], \quad \mathbf{s}_{\mathbf{L_{j,i}}}^{\tilde{u}_2} = [0.005, 0.01][\frac{m}{s}], \quad \mathbf{s}_{\mathbf{L_{j,i}}}^{\tilde{u}_3} = [0.08, 0.16, 0.25, 0.5][m]. \tag{7.8}$$

The fuzzification is done as stated in Definition 5.3.3. The final rule base is summarized in Table D.1 of the Appendix. The GINA controller is designed according to Algorithm 6.5.1 and the experimental results obtained for tracking a sequence of three set points are shown in Fig. 7.4. Thereby, the second set point is slowly moving along a ramp of the position coordinate x whereby a moving target is illustrated. Each desired set point is tracked fast and asymptotically stable subject to saturation and the valid DA is not left. The effect of the integrator enlarged subsystem can be especially

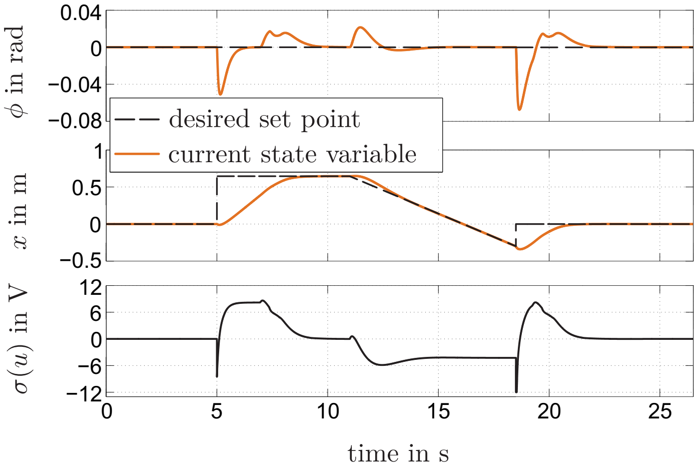


Figure 7.4: Experiment III: Set point tracking of the pendulum system by GINA controller together with switched RFS.

seen while stabilizing the moving set point ($t \in [12, 18]$s. The input signal saturates ($u_{max} = 12\,\mathrm{V}$) while activating the third set point and stability is ensured by the GINA controller.

7.2 Ballbot System

The multi-input Ballbot system of Fig. 6.4 is reconsidered with the control objective of fast and asymptotically stable tracking desired set points. The Ballbot system is a new mobile robot. It is an unstable, underactuated, nonholonomic system of non-minimum phase which in total makes it challenging to be controlled adequately. In Section 7.2.1, we summarize the modeling and the construction of the Ballbot.

Thereafter, we derive a novel motion control architecture based on the developed methods within this thesis. The key idea is to achieve a fast approaching of a desired set point by a velocity control unit and a final stabilization by switching to a position control unit. Both units are expanded with the GINA controller in its optimal form (Algorithm 6.3.1) to guarantee stability. Thereby, we relax the required calculations to be made for each input of the Ballbot to the single input case according to Section 6.4.2. We introduce two ways of realizing the control concept in Section 7.2.2 and Section 7.2.3: First based on linear control law and second by using a T-S controller design. We show the benefit of the T-S controller by experimentally comparing both concepts in Section 7.2.4 where we track a path of several set points.

7.2.1 Modeling and Test Rig

Fig. 7.5 depicts the Ballbot system which has been built-up from scratch. A motivation and a first practical application of Ballbots have been detailed in Example 6.4.1. The robot consists of a 60 cm high aluminum cuboid frame F where the batteries, the *Inertial Measurement Unit* (IMU) and the microcontroller unit for controlling the driving mechanism is placed. An ethernet cable L links the microcontroller to the target PC where the control actions are computed. DC-motors M_i, $i \in \mathbb{N}_{1:3}$, with a maximum input amplitude of $u_{max} = 12\,\text{V}$, drive three omniwheels W_i which are equidistantly positioned (120°) on a rigid ball B. This driving mechanism is called *Inverse Mouse-Ball Drive* [72]. The ball is an aluminum hollow sphere with a plastic coating. Ball clips C_i, $i \in \mathbb{N}_{1:3}$, enlarge the contact pressure between the omniwheels and the ball by three small roles in order to prevent slip. The total height and weight of the current version of our Ballbot are approximately 85 cm and 10.5 kg, respectively.

The nonlinear equations of motion of the Ballbot system at hand are derived in [92] . Due to the complexity of the nonlinear equations of motion, we will focus on the linear state space representation at the upright (unstable) equilibrium point of the Ballbot

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}_{lin}^{-1}\mathbf{K}_{lin} & -\mathbf{M}_{lin}^{-1}\mathbf{D}_{lin} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}}_{\mathbf{x}_{gen}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ -\mathbf{M}_{lin}^{-1}\mathbf{Q}_{lin} \end{bmatrix}}_{\mathbf{B}} \mathbf{u} \tag{7.9}$$

where $\mathbf{M}_{lin}$, $\mathbf{D}_{lin}$, $\mathbf{K}_{lin}$ and $\mathbf{Q}_{lin}$ represents the linearized mass matrix, damping matrix, stiffness matrix and the linearized excitation matrix, respectively. Their calculations are detailed in [92]. The three motors of the omniwheels form the input vector and the

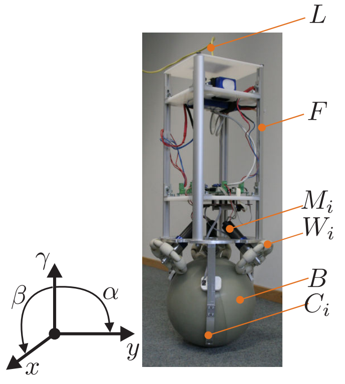


Figure 7.5: Ballbot system of the Institute of Automatic Control at TU München.

minimal coordinate vector

$$\mathbf{q} = \begin{bmatrix} x & y & \psi & \alpha & \beta & \gamma \end{bmatrix}^T, \tag{7.10}$$

and its derivatives form the general state vector $\mathbf{x}_{gen}$. It consists of the planar position coordinates x and y of the ball's center of gravity and its rotational angle ψ. Additionally, the tilt angles α and β around the x and y axis, respectively, and the yaw angle γ around the z axis of the robot's body are part of the state vector. The yaw angle is zero when the robot moves in its x coordinate (one omniwheel is in front). This mentioned physical quantities and their derivatives define the general state vector. As also detailed in [92], the rotational angle ψ has to eliminated from the state space model in order to ensure controllability. Hence, in this thesis the state vector

$$\mathbf{x} = \begin{bmatrix} x & y & \alpha & \beta & \gamma & \dot{x} & \dot{y} & \dot{\psi} & \dot{\alpha} & \dot{\beta} & \dot{\gamma} \end{bmatrix}^T, \tag{7.11}$$

is considered. Remember that a linear system is a special case of a T-S model (having only a single subsystem) and thus our developed methods can be directly applied here. The Ballbot is an updated version (mass, damping, friction, etc.) of the one in [92]. The updated parameters are given in Appendix C.2.

7.2.2 Velocity-Position-Yaw Angle Control: Linear

For our control objective of fast and asymptotically stable moving the Ballbot between set points we propose a novel multi-layer control architecture based on linear control theory according to Fig. 7.6 first. Three control units are needed:

First, moving fast towards a desired set point is done by regulating the speed of the Ballbot within the velocity control unit (VCU). Second, if approaching to a desired set point then the position control unit (PCU) becomes active which is in charge of asymptotically stabilizing the equilibrium. In both units, constraints and disturbances are handled by the GINA controller. Third, the orientation of the Ballbot is controlled by the yaw angle control unit (YACU). An additional transformation unit (TU) is required to adjust the input signal of the PCU/VCU according to the actual yaw angle. Each part of the control architecture is explained in the following. Estimates of the domain of attraction are done based on the first part of Algorithm 3.2.1 without UoD optimization (linear system) and avoiding input saturation to occur.

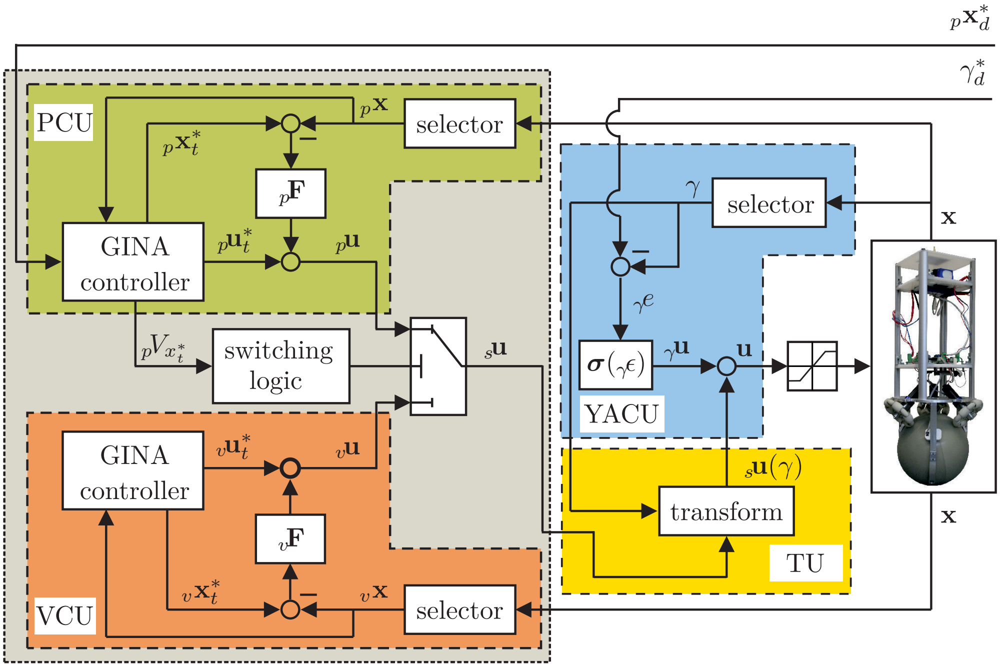


Figure 7.6: Linear motion control architecture of the Ballbot.

Position Control Unit: The PCU is designed for planar position control, i.e. yawing of the Ballbot is ignored. To this end, the state vector (7.11) can be reduced to

$$_p\mathbf{x} = \begin{bmatrix} x & y & \alpha & \beta & \dot{x} & \dot{y} & \dot{\alpha} & \dot{\beta} \end{bmatrix}^T \tag{7.12}$$

where the index p denotes the purpose of position control. The reduced state space model $_p\mathbf{A} \in \mathbb{R}^{8\times 8}$ and $_p\mathbf{B} \in \mathbb{R}^{8\times 3}$ are given in (C.4) and (C.5), respectively, of Appendix C.2. They can be derived by erasing the not required columns and rows from the original model [92].

A valid equilibrium (reference, temporary or desired) is given by

$$_p\mathbf{x}^* = \begin{bmatrix} x^* & y^* & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T . \tag{7.13}$$

The GINA controller, which extents the nominal controller $_p\mathbf{F}$, is designed according to Algorithm 6.3.1 and its real-time computation is relaxed by applying the results of Section 6.4 as follows: First, $_p\mathbf{x}_r^*$ is derived by Theorem 6.4.1 with

$$_p\mathbf{N}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.14}$$

(the position coordinates x and y span the matrix $_p\mathbf{N}$). Second, the updating of the scaling variable c can be reduced to a single level value calculation based on Corollary 6.4.1. Third, the equilibrium functions (6.5a) and (6.10b) are linear (6.14). The final obtained control input is denoted with ${}_p\mathbf{u}$ in Fig. 7.6.

Velocity Control Unit: The VCU is designed for planar velocity control, meaning the robot's position and its yawing movement are ignored. Erasing the corresponding state variables from (7.11) we obtain

$$ {}_v\mathbf{x} = \begin{bmatrix} \alpha & \beta & \dot{x} & \dot{y} & \dot{\alpha} & \dot{\beta} \end{bmatrix}^T , \tag{7.15} $$

where the index v denotes the purpose of velocity control. The corresponding system matrices ${}_v\mathbf{A} \in \mathbb{R}^{6\times 6}$, ${}_v\mathbf{B} \in \mathbb{R}^{6\times 3}$ are given in (C.6) and (C.7), respectively, of Appendix C.2.

A valid equilibrium (reference, temporary or desired) is given by

$$ {}_v\mathbf{x}^* = \begin{bmatrix} 0 & 0 & \dot{x}^* & \dot{y}^* & 0 & 0 \end{bmatrix}^T . \tag{7.16} $$

We decompose a desired velocity into its planar components $\dot{x}_d^*$ and $\dot{y}_d^*$, meaning

$$ \begin{bmatrix} \dot{x}_d^* \\ \dot{y}_d^* \end{bmatrix} = \frac{v_d^*}{\left\| \begin{bmatrix} x_d^* \\ y_d^* \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \right\|} \left(\begin{bmatrix} x_d^* \\ y_d^* \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \right) , \tag{7.17} $$

such that the final movement is towards the actually desired location x_d^*, y_d^*. The GINA controller, which extents the nominal controller ${}_v\mathbf{F}$, is designed according to Algorithm 6.3.1 and its real-time computation is relaxed by applying the results of Section 6.4 as follows: First, the reference equilibrium ${}_v\mathbf{x}_r^*$ is derived by Theorem 6.4.1 with

$$ {}_v\mathbf{N}^T = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{7.18} $$

($\dot{x}$ and $\dot{y}$ span ${}_v\mathbf{N}$). Second, the updating of the scaling variable c can be reduced to a single level value calculation based on Theorem 6.4.2 and Corollary 6.4.2. The final obtained control input is denoted with ${}_v\mathbf{u}$ in Fig. 7.6.

Switching Logic: We will move as long as possible within the VCU mode towards the desired spot before switching to the PCU for stabilization. To this end, we decelerate within the VCU when approaching $\mathbf{x}_d^*$ according to Corollary 6.4.2. The braking process starts at a distance of ϖ_b to the target. As soon as the value of the Lyapunov function is below a defined switching value

$$_{p}V_{x_t^*} \leq \varpi_s \leq_p \tilde{\eta}_0 \tag{7.19}$$

we will activate the PCU. Hence, the GINA controller within the PCU is able to deliver a stabilizable reference equilibrium and the asymptotic stabilization of the desired set point is ensured. The final values ϖ_s and ϖ_b will be experimentally defined. The finally obtained control input is denoted with $_{s}\mathbf{u}$ in Fig. 7.6 which either equals to $_{p}\mathbf{u}$ or $_{v}\mathbf{u}$.

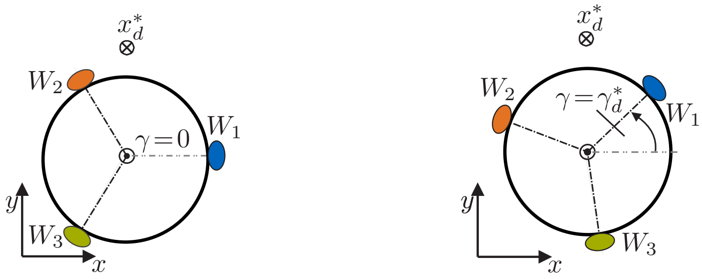
Yaw Angle Control Unit: A desired yaw angle γ_d^* of the robot is separately controlled within the YACU. Remember that the input amplitude are restricted to ±12 V. Of course, one might design a GINA controller as for the PCU and the VCU to effectively handle saturation. However, due to the fact that yawing is a simple and not crucial task concerning stability, we consider a saturated P-controller for regulating a yaw angle error $_{\gamma}e = \gamma_d^* - \gamma$. The controller is given by

$$_{\gamma}u_i = \boldsymbol{\sigma}(_{\gamma}e) = \begin{cases} -_{\gamma}u_{max} & \text{if } |_{\gamma}u_i| \leq u_{max}, \\ +_{\gamma}u_{max} & \text{if } |_{\gamma}u_i| \geq u_{max}, \\ _{\gamma}k \cdot {_{\gamma}e} & \text{else.} \end{cases} \tag{7.20}$$

where the index γ denotes the YACU. The limits of the control input and the proportional coefficient are $_{\gamma}u_{max}$ and $_{\gamma}k$, respectively. Since $_{\gamma}\mathbf{u}$ should only lead to a strict yawing, each of its elements $_{\gamma}u_i$, $i \in \mathbb{N}_{1:3}$, has to have the same absolute value. Note that the input amplitude for PCU/VCU has to be reduced by $_{\gamma}u_{max}$ to

$$_{p}u_{max} = {_{v}u_{max}} = {_{s}u_{max}} = u_{max} - {_{\gamma}u_{max}}. \tag{7.21}$$

Transformation Unit: A constant yaw angle $\gamma_d^* = 0$, as depicted in Fig. 7.7(a), is always present within both planar motion units (PCU/VCU). However, if $\gamma \neq 0$ then the VCU/PCU will lead to an incorrect movement due to the considered linear system. For instance, it is easy to see that x_d^* will not be reached when applying the motors' torques of the robot in Fig. 7.7(a) to the one in Fig. 7.7(b). Consequently, the actual yaw angle has to be taken into account which we will do within the transformation unit (TU). Here the objective is to adapt the control input signal $_{s}\mathbf{u}$ such that the VCU/PCU deliver a proper input signal subject to yawing.



(a) $\gamma_d^* = 0$

(b) $\gamma_d^* \neq 0$

Figure 7.7: Control input subject to yawing.

To this end, our goal is to compute a transformation

$$_s\mathbf{u}(\gamma) = \tilde{\mathbf{A}}(\gamma)_s\mathbf{u} \tag{7.22}$$

such that $_s\mathbf{u} \rightarrow {}_s\mathbf{u}(\gamma)$ leads to a correct motion subject to yawing. The transformation matrix $\tilde{\mathbf{A}}(\gamma)$ is derived based on the actual yaw angle γ:

Proposition 7.2.1. *The transformation matrix* $\tilde{\mathbf{A}}(\gamma)$ *in* (7.22) *is*

$$\tilde{\mathbf{A}}(\gamma) = \left(\mathbf{A}_{IF}^{\mathbf{0},\gamma}\tilde{\mathbf{A}}_{FW}\right)^{-1}\tilde{\mathbf{A}}_{FW}, \tag{7.23}$$

where $\tilde{\mathbf{A}}_{FW}$ *summarizes the first columns* $\mathbf{a}_{FW_i}$ *of the transformation matrices* $\mathbf{A}_{FW_i}$ *from the coordinate frame of the i–th omniwheel,* $i \in \mathbb{N}_{1:3}$*, to the robot's aluminum frame F:* $\tilde{\mathbf{A}}_{FW} = [\mathbf{a}_{FW_1}, \mathbf{a}_{FW_2}, \mathbf{a}_{FW_3}]$*. The matrix* $\mathbf{A}_{IF}^{\mathbf{0},\gamma}$ *transforms from the aluminum frame into the inertial one setting* $\alpha = \beta = 0\,\mathrm{rad}$ *and* $\mathbf{A}_{FW_i}$*,* $\mathbf{A}_{IF}$ *are given in [92].*

Proof: The voltages u_i and the torques M_i at the omniwheels are directly coupled when neglecting the motors' induction terms (see [92]). Both are transformed by $\tilde{\mathbf{A}}_{IW}(\gamma)$, which is derived based on the torques in the following: The torques $[_{W_i}\tilde{M}_i(\gamma), 0, 0]^T$ in the omniwheel's coordinate frame W_i for a yaw angle γ (Fig.7.7(b)) should lead to the same motion as $[_{W_i}M_i, 0, 0]^T$ calculated for the robot with $\gamma = 0\,\mathrm{rad}$ (Fig.7.7(a)). Referring both to the inertial coordinate frame we obtain

$$\underbrace{\mathbf{A}_{IF}^{\mathbf{0}}}_{\mathbf{I}}\sum_{i=1}^{3}\mathbf{A}_{FW_i}\cdot\begin{bmatrix} _{W_i}M_i \\ 0 \\ 0 \end{bmatrix} \overset{!}{=} \mathbf{A}_{IF}^{\mathbf{0},\gamma}\sum_{i=1}^{3}\mathbf{A}_{FW_i}\cdot\begin{bmatrix} _{W_i}\tilde{M}_i \\ 0 \\ 0 \end{bmatrix}. \tag{7.24}$$

The matrix $\mathbf{A}_{IF}^{0} = \mathbf{I}$ is obtained from the matrix $\mathbf{A}_{IF}$, which transforms from the aluminum frame into the inertial coordinates, when setting all angles therein to zero

($\alpha = \beta = \gamma = 0\,\mathrm{rad}$). Analogously, the matrix $\mathbf{A}_{IF}^{\mathbf{0},\gamma}$ results from $\mathbf{A}_{IF}$ setting $\alpha = \beta = 0\,\mathrm{rad}$ and γ equal to its actual value. The tilt angles α and β are both times set to zero as they are already considered in the planar control strategy within the PCU/VCU. Equation (7.24) can be compactly written as

$$\underbrace{\begin{bmatrix}\mathbf{a}_{FW_1}^T\\ \mathbf{a}_{FW_2}^T\\ \mathbf{a}_{FW_3}^T\end{bmatrix}^T}_{\tilde{\mathbf{A}}_{FW}}\underbrace{\begin{bmatrix}{}_{W_1}M_1\\ {}_{W_2}M_2\\ {}_{W_3}M_3\end{bmatrix}}_{{}_{\mathbf{w}}\mathbf{M}} \overset{!}{=} \mathbf{A}_{IF}^{\mathbf{0},\gamma}\cdot\underbrace{\begin{bmatrix}\mathbf{a}_{FW_1}^T\\ \mathbf{a}_{FW_2}^T\\ \mathbf{a}_{FW_3}^T\end{bmatrix}^T}_{\tilde{\mathbf{A}}_{FW}}\underbrace{\begin{bmatrix}{}_{W_1}\tilde{M}_1\\ {}_{W_2}\tilde{M}_2\\ {}_{W_3}\tilde{M}_3\end{bmatrix}}_{{}_{\mathbf{w}}\tilde{\mathbf{M}}}. \tag{7.25}$$

Rearranging (7.25) leads to

$$_{\mathbf{w}}\tilde{\mathbf{M}} = \underbrace{\left(\mathbf{A}^{\mathbf{0},\gamma}\tilde{\mathbf{A}}_{FW}\right)^{-1}\tilde{\mathbf{A}}_{FW}}_{\tilde{\mathbf{A}}_{IW}(\gamma)}\,{}_{\mathbf{w}}\mathbf{M}, \tag{7.26}$$

whereby the desired transformation matrix is obtained and the proof concluded. ■

Remark 7.2.1. Note that ${}_su(\gamma)_{max} \leq u_{max} - {}_\gamma u_{max}$ needs to be fulfilled for $\gamma \in [0, 2\pi]$ whereby the saturation limit ${}_su_{max}$ in (7.21) has to be further reduced.

7.2.3 Velocity-Position-Yaw Angle Control: T-S

In this section we formulate the control architecture of the previous section by considering the T-S framework. The corresponding architecture is depicted in Fig. 7.8. Compared to Fig. 7.6, no transformation unit is needed. The DAs are estimated analogous. An elimination of the transformation unit is achieved as follows: The objective is to consider yawing within the planar control units by several linear systems where each is a linearization at a specific yaw angle. These models are blended depending on the actual yaw angle in terms of a LO T-S formulation without affine terms, meaning

$$_p\dot{\mathbf{x}} = \sum_{i=0}^{r} h_i(\gamma)\left({}_p\mathbf{A}_i\,{}_p\mathbf{x} + {}_p\mathbf{B}_i\,{}_p\mathbf{u}\right) \tag{7.27}$$

for the Ballbot dynamics within the PCU and

$$_v\dot{\mathbf{x}} = \sum_{i=0}^{r} h_i(\gamma)\left({}_v\mathbf{A}_i\,{}_v\mathbf{x} + {}_v\mathbf{B}_i\,{}_v\mathbf{u}\right) \tag{7.28}$$

for the VCU. The corresponding state variables are identical to the linear control case according to (7.12) and (7.15). The interpolation between two surrounding linear sub-

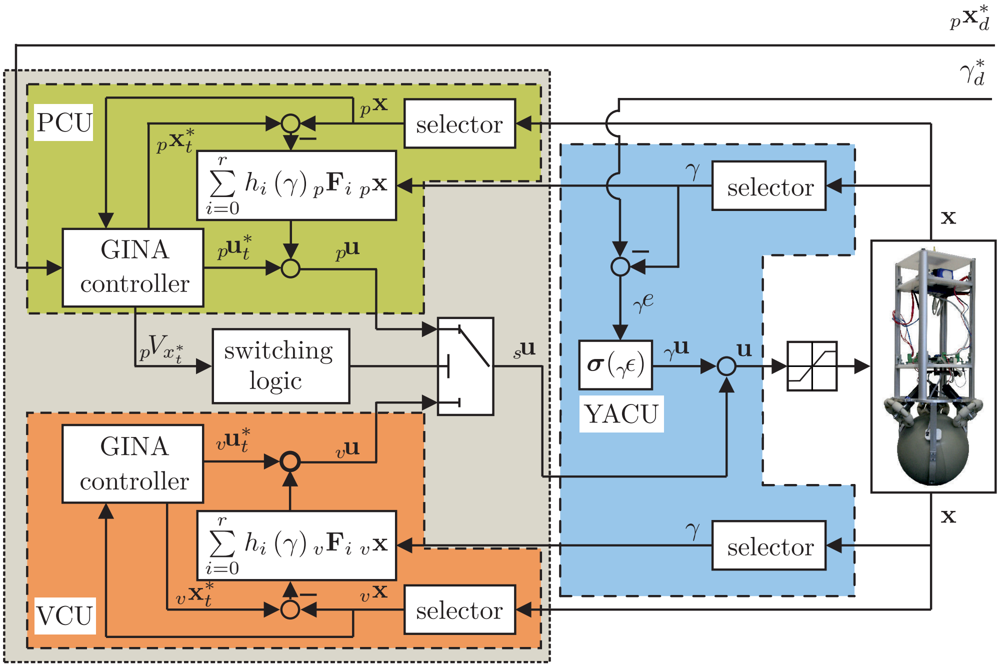


Figure 7.8: T-S motion control architecture of the Ballbot.

systems (if γ is in-between two operation points) is done by triangular membership functions $h_i(\gamma)$. Similar to the models, T-S controllers

$$ {}_p\mathbf{u} = \sum_{i=0}^{r} h_i(\gamma)\,{}_p\mathbf{F}_i\,{}_p\mathbf{x} \tag{7.29} $$

and

$$ {}_v\mathbf{u} = \sum_{i=0}^{r} h_i(\gamma)\,{}_v\mathbf{F}_i\,{}_v\mathbf{x} \tag{7.30} $$

can be designed based on our results of Section 3. The desired velocity is again calculated according to (7.17). Also the GINA controllers for the PCU and the VCU are computed analogous to Section 7.2.2. Note that the Ballbot fulfills Theorem 6.4.1 (see Example 6.4.1). Hence, (7.14) and (7.18) are also valid for the T-S models. The switching logic and the YACU are identical to the linear case.

Remark 7.2.2. Note that in contrast to Section 7.2.2 (Remark 7.2.1), ${}_su_{max}$ does not have to be reduced. In other words, as the controllers (7.29) and (7.30) are already aware of the actual yaw angle the equation (7.21) stays valid.

7.2.4 Results

In this section, we will provide experimental results concerning the introduced motion control architectures of Fig. 7.6 and Fig. 7.8. We will validate and compare both concepts. The proportional control coefficient for controlling the yaw angle (YACU) in (7.20) is set to ${}_{\gamma}k = -1.0\,\mathrm{V/\,rad}$ and the corresponding saturation limit is chosen to be at ${}_{\gamma}u_{max} = 0.7\,\mathrm{V}$. Hence, the maximum input amplitude for the PCU/VCU has to be reduced to ${}_{s}u(\gamma)_{max} = 11.3\,\mathrm{V}$ according to (7.21). This value can be directly considered if planar motion and yawing is linked via T-S blending according to Section 7.2.3 (see Remark 7.2.2). We discretize $\gamma \in [0, 2\pi]$ in $r = 359$ equidistant values and thus linear subsystems, for instance $\gamma_1 = 0.017\,\mathrm{rad}$ (1°) and $\gamma_2 = 0.034\,\mathrm{rad}$ (2°). Concerning the linear control approach with transformation of Section 7.2.2, the maximum input amplitude has to be further reduced as stated in Remark 7.2.1. More precisely, we have to set ${}_{s}u_{max} = 9.75\,\mathrm{V}$ for the case at hand.

For both approaches, we choose LQR controllers (for each linear system) with the weighting matrices

$$
{}_{p}\mathbf{Q}_{LQR} = \mathrm{diag}\,(100,\ 100,\ 100,\ 100,\ 1000,\ 1000,\ 0,\ 0)\,, \quad {}_{p}\mathbf{R}_{LQR} = \mathrm{diag}\,(4,\ 4,\ 4) \tag{7.31}
$$

for the PCV and

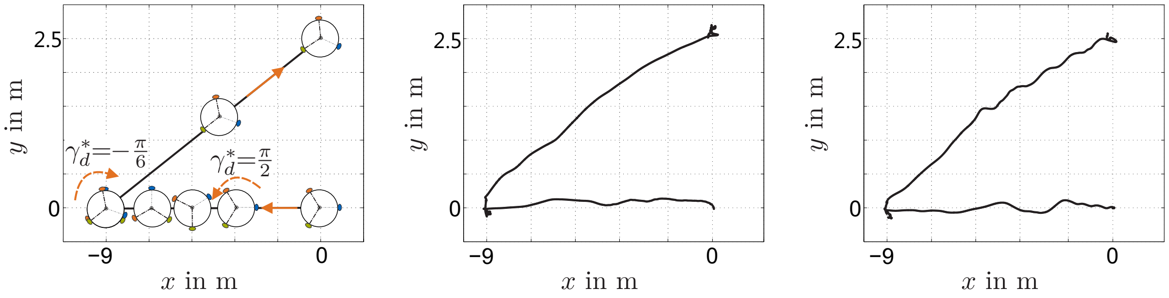
$$
{}_{v}\mathbf{Q}_{LQR} = \mathrm{diag}\,(5,\ 5,\ 0.2,\ 0.2,\ 100,\ 100)\,, \quad {}_{v}\mathbf{R}_{LQR} = \mathrm{diag}\,(1.2,\ 1.2,\ 1.2) \tag{7.32}
$$

concerning the VCU. We set the desired velocity of the VCU to $|{}_{v}\mathbf{v}_{d}^{*}| = 0.77\,\mathrm{m/s}$ which corresponds to a maximum required input value of ${}_{v}u_{d}^{*} = 8\,\mathrm{V}$. The braking process starts at $\varpi_b = 0.3\,\mathrm{m}$ and the PCU is activated as soon as $\varpi_s = 0.2$ (see (7.19)).

Based on that setup, we experimentally validate the control architectures concerning a desired track of set points according to Fig. 7.9(a). It consists of five equilibria

$$
\begin{aligned}
&\mathbf{x}_{d,1}^{*} = [0, 0, \dots, 0]^T, \ \ \mathbf{x}_{d,2}^{*} = [-9, 0, 0, \dots, 0]^T, \ \ \mathbf{x}_{d,3}^{*} = [-9, 0, 0, \frac{\pi}{2}, 0, 0, \dots, 0]^T, \\
&\mathbf{x}_{d,4}^{*} = [-9, 0, 0, -\frac{\pi}{6}, 0, 0, \dots, 0]^T, \ \ \mathbf{x}_{d,5}^{*} = [0, 2.5, 0, -\frac{\pi}{6}, 0, 0, \dots, 0]^T.
\end{aligned} \tag{7.33}
$$

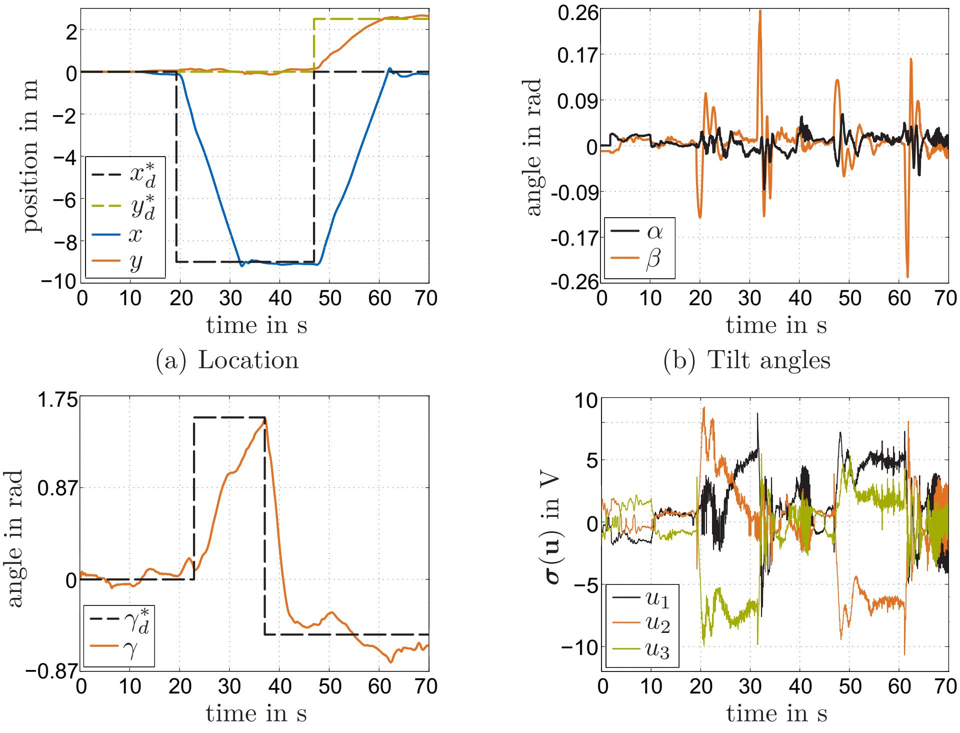
The Ballbot starts at $\mathbf{x}_{d,1}^{*}$. While moving towards $\mathbf{x}_{d,2}^{*}$ the desired yaw angle is changed to $\gamma_{d,3}^{*} = \pi/2\,\mathrm{rad}$ (90°) which is realized by activating $\mathbf{x}_{d,3}^{*}$. Then the robot yaws at the spot towards $\gamma_{d,4}^{*} = -\pi/6\,\mathrm{rad}$ (−30°) and finally it moves without yawing to the final spot which is given by $\mathbf{x}_{d,5}^{*}$. Each set point is activated manually after the Ballbot reaches the prior one.



(a) Desired track of set points (b) Experiment I: Section 7.2.2 (c) Experiment II: Section 7.2.3

Figure 7.9: Experiment: Set point tracking of the Ballbot.

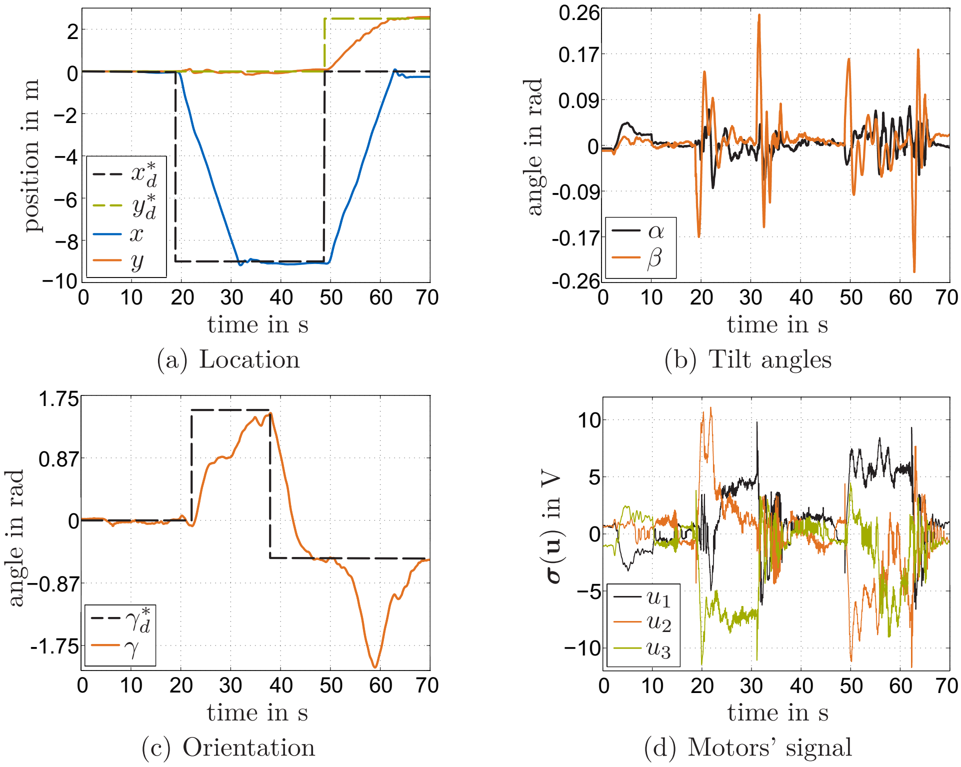
Fig. 7.9(b) and Fig. 7.9(c) depict the final tracking behavior with the linear and the T-S-based approach, respectively. As can be seen, the robot reaches each desired set point and hence the desired track has been realized by both control concepts. Detailed information can be gathered from Fig. 7.10 and Fig. 7.11 where relevant state variables as well as the required input signals are shown for both control concepts. As already known from Fig. 7.9, the Ballbot reaches each desired equilibrium and saturation is prevented due to the GINA controllers. If they are deactivated then both experiments become unstable whereby the relevance of the GINA controller becomes obvious.



(a) Location (b) Tilt angles

(c) Orientation (d) Motors' signal

Figure 7.10: Experiment I: Set point tracking of the Ballbot with linear controller.



(a) Location

(b) Tilt angles

(c) Orientation

(d) Motors' signal

Figure 7.11: Experiment II: Set point tracking of the Ballbot with T-S controller.

According to the Remarks 7.2.1 and 7.2.2, higher voltage amplitudes are expected by the T-S controller which is confirmed when comparing Fig. 7.10(d) and Fig. 7.11(d). This allows a faster acceleration and deceleration of the robot which finally enables a faster shifting of the temporary set point towards the desired one. Consequently, the desired set points should be tracked faster especially while moving with a constant yaw angle. This becomes clear at the last set point change ($\mathbf{x}_{d,4}^* \rightarrow \mathbf{x}_{d,5}^*$) according to (7.33). The T-S controller is with $\delta t = 65\,\mathrm{s} - 48.8\,\mathrm{s}$ approximately 5 s faster than the linear controller ($\delta t = 68\,\mathrm{s} - 46.9\,\mathrm{s}$). Note that the current construction leads to a non-smooth moving of the robot when driving faster which in turn results in an additional yaw angle deflection. Hence, the robot seems to turn itself into a preferred orientation for motion. Despite this, asymptotic stability is guaranteed by the GINA controller.

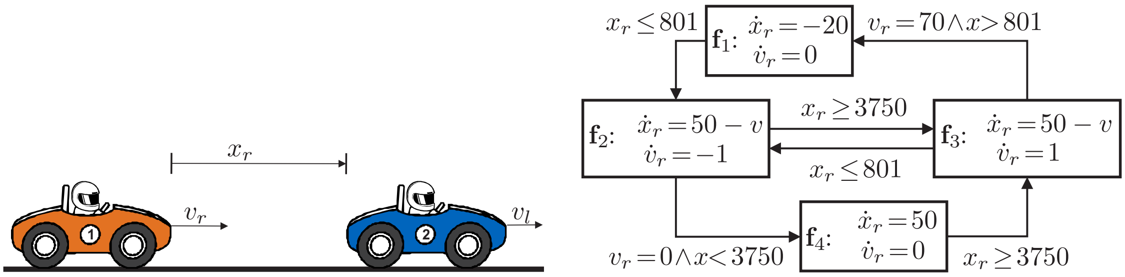
7.3 Car-Distance-Control by Hybrid Automaton

In this section, we investigate an active cruise control (ACC) system which allows to drive a car in a motorcade very comfortable. The ACC automatically accelerates and decelerates trying to keep the distance to the car in front constant.

> In the following we enhance the ride comfort of a simple ACC system which is given in terms of a hybrid automaton. The drawback of the system lies in a large region within which the distance to the car in front and thus the car's velocity vary (large ϵ-region, see Definition 2.3.1). In addition, a change of the velocity is done non-smooth and thus the ride comfort is reduced. Both disadvantages are bypassed by transforming the automaton into a switched recurrent fuzzy system (RFS) formulation according to Section 5.3.

Consider the active cruise control (ACC) illustrated in Fig. 7.12(a). The rear car should follow the leading one which drives with a constant speed of $v_l = 50\,\text{km/h}$. The rear car senses the distance x between the cars and decides whether to increase or decrease its speed v_r. Its maximum speed is assumed to be at $v_r = 70\,\text{km/h}$ and $0\,\text{km/h}$ is the lower speed limit. Fig. 7.12(b) shows the hybrid automaton for realizing the ACC. Its region stability has been proven in [100, 101]. The automaton consists of four linear subsystems $\mathbf{f}_1(x_r, v_r)$ to $\mathbf{f}_4(x_r, v_r)$. The first one describes the dynamics when the rear car drives with its maximal speed and $\mathbf{f}_4(x_r, v_r)$ becomes active for minimum velocity. The second and the third subsystems have to be considered if deceleration and acceleration phase, respectively, is required. The acceleration phase is activated if $x_r \geq 3750\,\text{m}$ and the deceleration phase becomes active if $x \leq 801\,\text{m}$.

In the following we show the transfer of the hybrid automaton into a switched RFS formulation according to Section 5.3: Following Definition 5.3.2, the membership functions concerning v_r, which is the first input $\tilde{u}_1$ of the recurrent fuzzy switching, has



x_r
v_r
v_l
1
2
$x_r \leq 801$
$\mathbf{f}_1$: $\dot{x}_r = -20$
$\dot{v}_r = 0$
$v_r = 70 \wedge x > 801$
$\mathbf{f}_2$: $\dot{x}_r = 50 - v$
$\dot{v}_r = -1$
$x_r \geq 3750$
$x_r \leq 801$
$\mathbf{f}_3$: $\dot{x}_r = 50 - v$
$\dot{v}_r = 1$
$\mathbf{f}_4$: $\dot{x}_r = 50$
$\dot{v}_r = 0$
$v_r = 0 \wedge x < 3750$
$x_r \geq 3750$

(a) Schematic (b) Hybrid automaton

Figure 7.12: Active cruise control via hybrid automaton.

Table 7.2: Rule base of the switched RFS distance controller

$\mathbf{f}_i$	$[Z,S]^T$	$[Z,L]^T$	$[L,S]^T$	$[L,L]^T$	$[S,S]^T$	$[S,L]^T$
$\mathbf{f}_1$	$\mathbf{f}_2$	$\mathbf{f}_1$	$\mathbf{f}_2$	$\mathbf{f}_1$	$\mathbf{f}_2$	$\mathbf{f}_1$
$\mathbf{f}_2$	$\mathbf{f}_4$	$\mathbf{f}_3$	$\mathbf{f}_2$	$\mathbf{f}_3$	$\mathbf{f}_2$	$\mathbf{f}_3$
$\mathbf{f}_3$	$\mathbf{f}_2$	$\mathbf{f}_3$	$\mathbf{f}_2$	$\mathbf{f}_1$	$\mathbf{f}_2$	$\mathbf{f}_3$
$\mathbf{f}_4$	$\mathbf{f}_4$	$\mathbf{f}_3$	$\mathbf{f}_4$	$\mathbf{f}_3$	$\mathbf{f}_4$	$\mathbf{f}_3$

to be a rectangular-shaped membership function. In detail, three linguistic velocity regions need to be considered: $\mathbf{L}_\mathbf{j}^{\tilde{u}_1} = \{\text{zero, small, large}\}$, which are shortened by "Z", "S", and "L", henceforth. The core positions are $\mathbf{s}_{\mathbf{L_{j,1}}}^{\tilde{u}_1} = [0, 70]$km/h. The membership functions for the second input $\tilde{u}_2 = x_r$ are chosen in accordance with Definition 5.3.1. We consider trapezoidal functions as shown in Fig. 2.2(a). Two linguistical characteristics $\mathbf{L}_\mathbf{j}^{\tilde{u}_2} = \{\text{small, large}\}$ are required. Analogous to the velocity, they are shortened by "S" and "L" and their core positions are $\mathbf{s}_{\mathbf{L_{j,1}}}^{\tilde{u}_2} = [801, 3750]$m (see [32]). The resulting rule base of the switched RFS is illustrated in Table 7.2. The currently active subsystem is written in the first column and the linguistical characteristics of the input $\tilde{\mathbf{u}}$ is placed in the first row. Consequently, a linguistic differential equation (5.7) is obtained. For instance, the first rule is: "If $\bar{x}(k)$ is $\mathbf{f}_1$ and $\bar{\mathbf{u}}(k)$ is $[Z,S]^T$ then $\bar{x}(k+1)$ is $\mathbf{f}_2$".

A comparison between the hybrid automaton and the switched RFS is shown in Fig. 7.13. The ϵ-region (see Definition 2.3.1) of the automation is highlighted. The corresponding region of the switched RFS is within the ϵ-region whereby Theorem 5.3.1 is confirmed. Also the smoothing of the signals (e.g. the velocity) become clear.

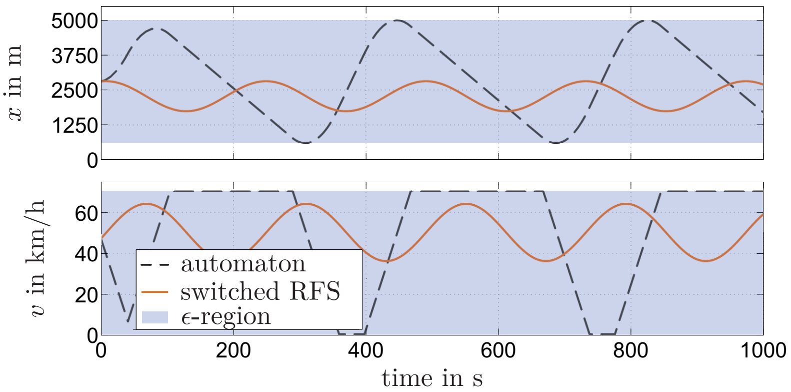


Figure 7.13: Simulation results of the ACC.

7.4 Summary

In this chapter, we have confirmed the performance of the methods that have been developed in the first part of this thesis. First, we have shown experimental results for the nonlinear inverted pendulum on cart in SE-NL T-S form. Due to the GINA controller all desired set points have been asymptotically stabilized subject to constraints. Both, the iterative and the analytical implementation of the GINA controller have been validated. In the second example, we have developed a novel motion control strategy for a complex mobile robot with multi-inputs. The strategy allows a fast and asymptotically stable set point tracking by switching between a velocity and a position control unit. In both units, the GINA controller has been integrated such that stability is ensured even for large set point changes. The required computational effort has been reduced to the single input case. A comparison of a T-S controller with a linear control law illustrates the advantages of the T-S based strategy. Consequently, the experimental results have confirmed the applicability of our methods to complex mechanic systems. Finally, an active cruise controller (ACC) has been investigated based on a hybrid automaton. We have reformulated the model into a switched recurrent fuzzy system (RFS) and have provided simulation results that have shown how the stability region of the hybrid automaton becomes smaller and the switching can be smoothed by the RFS.

Part II

Trajectory Tracking Control

Chapter 8

Flatness-based Trajectory Generation

With this chapter we start the second part of the thesis which is devoted to trajectory tracking. We focus on the *two-degree of freedom* (2-DOF) control structure and develop new methods for both of its parts: the feedforward and error controller. More precisely, the present Chapter 8 aims to design a performant trajectory along predefined waypoints such that the transition time becomes minimal subject to system constraints. That is for instance a major task in motion planing of mobile robotic systems [76]. Subsequently, in the Chapters 9 to 11 the problem of deriving a tracking controller that exploits the system's input amplitude and rate while simultaneously ensuring the asymptotic stabilization of the trajectory is addressed. Thereby, we tackle the problem that the available input signal changes over time depending on the feedforward signal (time-varying constraints). The remainder of this chapter is organized as follows: In Section 8.1 we give a formal statement of the considered trajectory generation problem. Subsequently, we introduce in Section 8.2 an algorithm for designing performant trajectories along waypoints based on differential flatness.

8.1 Problem Formulation

In this section, we propose a strategy based on differential flatness (see Section 2.5) for computing a performant trajectory. In this context performant means that the final transition time is minimized while predefined waypoints are precisely hit and system relevant state and input constraints are not violated. The key idea of the approach is to formulate the trajectory generation task as an optimization problem based on polynomial trajectory pieces such that a stationary movement (movement without acceleration) is obtained at each waypoint. Such a movement can be realized by lots of technical systems, e.g. mobile robots, robotic manipulators, generators or wind turbines [85, 103, 105]. Consequently, the proposed planing strategy can be understood as a sequence of changes between easily manageable (controllable and stable) state configurations. To this end, we consider differential flat input affine dynamical systems

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{8.1}$$

with $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^m$, which fulfills the following assumptions:

Assumption 8.1.1. The state vector of a system (8.1) can be rewritten as

$$\mathbf{x}^T = \begin{bmatrix} \boldsymbol{\Gamma}^T & \dot{\boldsymbol{\Gamma}}^T & \tilde{\boldsymbol{\Gamma}}^T \end{bmatrix}, \tag{8.2}$$

where the vector $\boldsymbol{\Gamma} \in \mathbb{R}^{n_\Gamma}$ summarizes the location variables (position state variables of the system), $\dot{\boldsymbol{\Gamma}}$ denotes the corresponding derivatives (velocity state variables) and $\tilde{\boldsymbol{\Gamma}}$ the remaining state variables.

Assumption 8.1.2. System (8.1) can be operated with a stationary velocity (stationary movement without acceleration), meaning $\ddot{\boldsymbol{\Gamma}} = \mathbf{0}$. In other words, a velocity equilibrium

$$(\mathbf{x}^*_{red})^T = \begin{bmatrix} \left(\dot{\boldsymbol{\Gamma}}^*\right)^T & \left(\tilde{\boldsymbol{\Gamma}}^*\right)^T \end{bmatrix} \tag{8.3}$$

when ignoring $\boldsymbol{\Gamma}$ (erasing from the equations of motion). Thereby, $\mathbf{x}_{red} \in \mathbb{R}^{n-n_\Gamma}$ denotes the obtained reduced state vector.

Obviously, both assumptions are inherently fulfilled by lots of technical systems, e.g. mobile robots, robotic manipulators, generators or wind turbines. If both assumptions are fulfilled then the input signal of a stable stationary movement is given by

$$\mathbf{u}^*_{red} = -\left(\mathbf{g}^T(\mathbf{x}^*_{red})\mathbf{g}(\mathbf{x}^*_{red})\right)^{-1} \mathbf{g}^T(\mathbf{x}^*_{red})\mathbf{f}(\mathbf{x}^*_{red}). \tag{8.4}$$

Based on that the problem that we address is:

Problem 8.1.1. Given a system (8.1) that fulfills the Assumptions 8.1.1 and 8.1.2. Find a sufficiently smooth polynomial trajectory of the flat output such that: First, the transition time $\mathcal{T} = \{t | t_0 = 0 \leq t \leq t_e\}$ becomes as small as possible. Second, input and state depending constraints $\hat{\mathbf{c}}(\mathbf{x}_T(t), \mathbf{u}_T(t)) \leq \mathbf{0}$ of the system are adhered to. Third, predefined waypoints $\boldsymbol{\Gamma}_i$, $i \in \mathbb{N}_{0:w}$, are precisely hit by the trajectory whereby the first and the last waypoint define *rest-to-rest coordinates*, i.e.

$$\mathbf{x}_T(t_i)^T = \begin{bmatrix} \boldsymbol{\Gamma}_i^T & \left(\dot{\boldsymbol{\Gamma}}_i^*\right)^T & \left(\tilde{\boldsymbol{\Gamma}}_i^*\right)^T \end{bmatrix}, \quad i \in \mathbb{N}_{0:w}, \tag{8.5}$$

The key idea of the proposed method is based on defining a stationary movement (movement without acceleration) of the system at each $i \in \mathbb{N}_{1:w-1}$ waypoint. We formulate a two-point boundary value problem (similar to a set point change) between each pair of subsequent waypoints. This allows us to formulate the trajectory generation Problem 8.1.1 as an optimization of trajectory pieces with a stationary movement at each beginning and end. The stationary velocity at each waypoint and the overall transition time form the optimization parameters.

8.2 Piecewise Trajectory Generation Along Waypoints

Assuming that a velocity $\dot{\mathbf{\Gamma}}_i^*$ is defined for all $i \in \mathbb{N}_{1:w-1}$ waypoints, we can state the optimization problem

$$\min_{\left\{t_{e,i},\dot{\mathbf{\Gamma}}_i^*\right\}} \left(\sum_{i=1}^{w} t_{e,i}\right) \quad \text{s. t. } \hat{\mathbf{c}}(\mathbf{x}_T(t), \mathbf{u}_T(t)) \leq \mathbf{0}, \tag{8.6}$$

where the optimization parameters are $t_{e,i}$, which denote the final time of the i-th flat trajectory piece (2.56), and $\dot{\mathbf{\Gamma}}_i^*$ if the initial time of the first trajectory piece $t_{0,i}$, for $i = 0$, is defined. The proposed strategy for solving (8.6) is summarized in Algorithm 8.2.1. Fig. 8.1 clarifies the proposed algorithm based on a trajectory along seven ($w = 6$) waypoints considering a two dimensional position subspace x and y. The waypoints $\mathbf{\Gamma}_i = [x_i, y_i]^T$ are highlighted by black circles and the corresponding velocities $\dot{\mathbf{\Gamma}}_i^* = [\dot{x}_i^*, \dot{y}_i^*]^T$, $i \in \mathbb{N}_{0:6}$, are depicted by orange arrows.

As the first and the last waypoint $\mathbf{\Gamma}_0$ and $\mathbf{\Gamma}_w$, respectively, define rest-to-rest coordinates, the corresponding velocities are set to zero (line 1 of Algorithm 8.2.1). Consequently, $\dot{\mathbf{\Gamma}}_0^*$ and $\dot{\mathbf{\Gamma}}_6^*$ are zero in Fig. 8.1. The velocities at each waypoint in-between

Algorithm 8.2.1 Piecewise Flatness-based Trajectory

```
 1: set $\dot{\mathbf{\Gamma}}_0^* = \dot{\mathbf{\Gamma}}_w^* = \mathbf{0}$
 2: for $i = 1 : 1 : w - 1$ do
 3:     for $l = 1 : 1 : \mathbb{R}^{n_{\dot{\Gamma}}}$ do
 4:         if $(\Gamma_{i-1,l}^* < \Gamma_{i,l}^* < \Gamma_{i+1,l}^*)$ or $(\Gamma_{i-1,l}^* > \Gamma_{i,l}^* > \Gamma_{i+1,l}^*)$ then
 5:             compute auxiliary trajectory: $\Gamma_{i-1,l}^* \to \Gamma_{i+1,l}^*$ with $t_{e,i}$ based on (8.8)
 6:             calculate $\dot{\Gamma}_{i,l}^*$ based on (8.9)
 7:         else
 8:             $\dot{\Gamma}_{i,l}^* = 0$
 9:     calculate $\tilde{\mathbf{\Gamma}}_i^*$ belonging to $\dot{\mathbf{\Gamma}}_i^*$
10:     generate the $i$-th trajectory piece with $t_{e,i}$ based on (8.8)
```

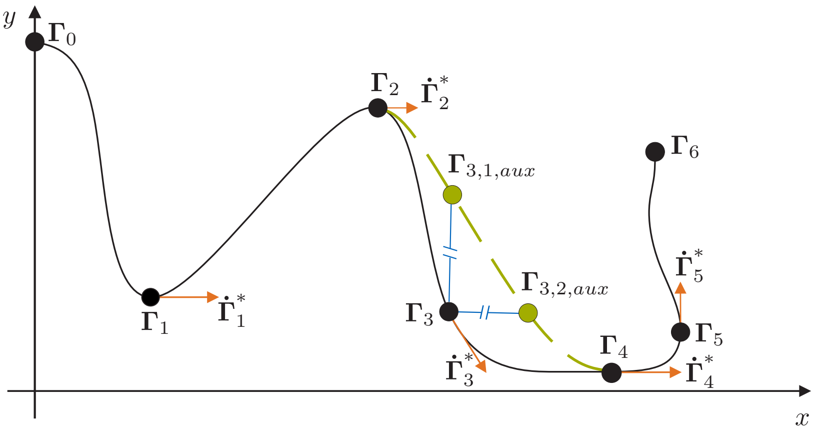

y
$\mathbf{\Gamma}_0$
$\mathbf{\Gamma}_2$
$\dot{\mathbf{\Gamma}}_2^*$
$\mathbf{\Gamma}_{3,1,aux}$
$\mathbf{\Gamma}_6$
$\dot{\mathbf{\Gamma}}_5^*$
$\mathbf{\Gamma}_{3,2,aux}$
$\mathbf{\Gamma}_1$
$\dot{\mathbf{\Gamma}}_1^*$
$\mathbf{\Gamma}_3$
$\mathbf{\Gamma}_5$
$\mathbf{\Gamma}_4$
$\dot{\mathbf{\Gamma}}_3^*$
$\dot{\mathbf{\Gamma}}_4^*$
x

Figure 8.1: Sketch of the proposed Algorithm 8.2.1 concerning a trajectory along the waypoints $\mathbf{\Gamma}_i$, $i \in \mathbb{N}_{0:6}$.

are calculated component-wise for each waypoint $i = 1$ until $i = w - 1$. Therefore, a case differentiation is done for each component l of the considered waypoint $\mathbf{\Gamma}_i$ and its surrounding ones $\mathbf{\Gamma}_{i-1}$, $\mathbf{\Gamma}_{i+1}$ (line 2 until 8): If the l-th component of these waypoints does continuously increase or decrease then the actual waypoint i can be passed with a certain speed (within the component l). Otherwise, the moving direction has to change somewhere in-between these waypoints. The second and simpler case is solved by setting $\dot{\Gamma}^*_{i,l} = 0$ in line 8 (the change of the moving direction is forced to be at $\mathbf{\Gamma}_i$). Concerning the first explained case, we estimate a velocity $\dot{\Gamma}^*_{i,l} \neq 0$ (lines 5 and 6) by generating an *flatness-based auxiliary trajectory* between the l-th component of the neighboring waypoints $\mathbf{\Gamma}_{i-1}$ and $\mathbf{\Gamma}_{i+1}$. To this end, we incorporate the already calculated velocities $\dot{\mathbf{\Gamma}}^*_{i-1}$ (prior iteration) and set $\dot{\mathbf{\Gamma}}^*_{i+1} = \mathbf{0}$. Consequently,

$$\mathbf{x}_T(t_{i-1}) = \begin{bmatrix} \mathbf{\Gamma}^T_{i-1} & \left(\dot{\mathbf{\Gamma}}^*_{i-1}\right)^T & \left(\tilde{\mathbf{\Gamma}}^*_{i-1}\right)^T \end{bmatrix}^T, \quad \mathbf{x}_T(t_{i+1}) = \begin{bmatrix} \mathbf{\Gamma}^T_{i+1} & \mathbf{0}^T & \left(\tilde{\mathbf{\Gamma}}^*_{i+1}\right)^T \end{bmatrix}^T \tag{8.7}$$

define the boundary conditions for the auxiliary trajectory. These conditions can be transformed into boundary conditions of the flat output (see (2.57)) and a flat auxiliary trajectory piece (2.56) can be computed. The required transition time of each piece is optimized subject to input and state constraints $\hat{\mathbf{c}}(\mathbf{x}_T(t), \mathbf{u}_T(t)) \leq \mathbf{0}$. Similar to a bisection approach [22], we iterate the end-time

$$t_{e,i}(k+1) = \begin{cases} t_{e,i}(k) + 1/2\,|t_{e,i}(k) - t_{e,i}(k-1)| & \text{if } \hat{\mathbf{c}}(\mathbf{x}_T(t), \mathbf{u}_T(t)) \leq \mathbf{0} \text{ is violated,} \\ t_{e,i}(k) - 1/2\,|t_{e,i}(k) - t_{e,i}(k-1)| & \text{else,} \\ \text{abort if } k = \chi,\ \chi \in \mathbb{N}, \text{ or a constraint is within } 95\,\% \text{ and } 100\,\%. \end{cases} \tag{8.8}$$

Therein, k is the iteration index and $t_{e,i}(k)$ and $t_{e,i}(k-1)$ denote the solutions of the prior and the second to the last iteration, respectively. Initially, we set $t_{e,i}(0) = 0$ and $t_{e,i}(1) > t_{e,i}(0)$. Thereby, it should be paid attention to the fact that the initial value for $t_{e,i}(1)$ has to be chosen such that the second case in (8.8) occurs for the first iteration for ensuring the convergence of the optimization.

This can be easily guaranteed if, for instance, $t_{e,i}(1)$ is set to an arbitrary value and doubled till all constraints $\hat{\mathbf{c}}(\mathbf{x}_T(t), \mathbf{u}_T(t)) \leq \mathbf{0}$ are fulfilled. According to the third case in (8.8), the iteration is aborted if k reaches a predefined upper bound χ or one of the constraints is within 95 % and 100 % of its maximum (before violation). Such a designed auxiliary trajectory is exemplarily shown in Fig. 8.1 (dashed green line) between the waypoints $\mathbf{\Gamma}_2$ and $\mathbf{\Gamma}_4$. The velocity component $\dot{\Gamma}^*_{i,l}$ is finally estimated to

$$\dot{\Gamma}^*_{i,l} = \dot{\Gamma}_{i,l,aux}(\Gamma_{i,l,aux}). \tag{8.9}$$

In other words, $\dot{\Gamma}^*_{i,l}$ is set equal to the velocity at the waypoint component $\Gamma_{i,l,aux}$ along the auxiliary trajectory. The explained estimation principle is illustrated for $\mathbf{\Gamma}_3$ in Fig. 8.1. Hence, the velocities $\dot{\Gamma}_{3,1,aux}(\Gamma_{3,1,aux})$ and $\dot{\Gamma}_{3,2,aux}(\Gamma_{3,2,aux})$ of the auxiliary trajectory between the waypoints $\mathbf{\Gamma}_2$ and $\mathbf{\Gamma}_4$ form the velocity vector at the third waypoint.

If the velocity vector for a waypoint $\mathbf{\Gamma}_i$ has been estimated, the corresponding values of the missing state variable $\tilde{\mathbf{\Gamma}}^*_{\boldsymbol{i}}$ have to be calculated (line 9 of the algorithm) based on Assumption 8.1.2. Finally, the desired state vectors of two subsequent waypoints (8.7) are known and thus the i-th trajectory piece can be derived. Analogous to the flatness-based auxiliary trajectory, the transition time is optimized by the bisection approach subject to state and input constraints according to (8.8).

8.3 Summary

In this chapter, we have dealt with the problem of generating performant trajectories along waypoints subject to constraints. We have proposed a piecewise polynomial trajectory approach based on differential flatness. The key idea of the strategy is to formulate the trajectory generation task as an optimization problem such that a stationary movement (movement without acceleration) is obtained at each waypoint and the overall transition time becomes minimal. We have developed a new numerical algorithm that allows a numerically efficient implementation. In other words, the algorithm has been designed such that its real-time applicability of the approach is enhanced.

Chapter 9

From Set Point to Trajectory Tracking

Within this chapter, we transfer our results of Part I of this thesis (set point tracking control) to trajectory tracking based on a two-degree of freedom (2-DOF) control structure. The key idea is to formulate the error dynamics of a nonlinear system in terms of a T-S model whereby the results of Part I are directly applicable. After a formal problem statement in Section 9.1 we derive the error dynamics of the trajectory tracking problem in form of a LO T-S formulation. Based on that, our results concerning set point tracking (estimating a DA, controller design, GINA controller) are directly applicable to trajectory tracking if each system constraint is subdivided into two parts (see Section 2.4).

9.1 Problem Formulation

Suppose that a desired input and state trajectory $\mathbf{u}_T$ and $\mathbf{x}_T$, respectively, is given for a nonlinear system. For designing an appropriate tracking controller, we have to investigate the system's error dynamics. That is most often done by linearizing the system around a desired trajectory which results in linear time-variant error dynamics

$$\dot{\mathbf{e}} = \mathbf{A}(\mathbf{x}_T, \mathbf{u}_T)\mathbf{e} + \mathbf{B}(\mathbf{x}_T, \mathbf{u}_T)\mathbf{u}_e \tag{9.1}$$

where the time dependent dynamics is directly related to the desired trajectory (see Chapter 5.1.2 in [38] for a detailed derivation). The error $\mathbf{e}$ and the input vector $\mathbf{u}_e$ are

$$\mathbf{e} = \mathbf{x}_T - \mathbf{x}, \quad \mathbf{u}_e = \mathbf{u}_T - \mathbf{u}. \tag{9.2}$$

The time dependence in (9.1) prevents the applicability of the T-S and LMI-based results from Part I of the thesis. The sole exception are linear systems where the error dynamics are known to be a time-invariant linear system.

One might think of solving that dilemma based on a SE-NL T-S model (see Definition 2.1.1). This leads to a desired trajectory in T-S notation as well:

$$\dot{\mathbf{x}}_T = \sum_{i=1}^{r} h_i(\mathbf{z}_{s,T})\left(\mathbf{A}_i\mathbf{x}_T + \mathbf{B}_i\mathbf{u}_T\right). \tag{9.3}$$

Remember that the premise vector $\mathbf{z}_s(\mathbf{x},\mathbf{u})$ depends on the state and input variables that nonlinearly affect the system (see Definition 2.1.1). Hence, the index T at the system's premise vector $\mathbf{z}_{s,T}$ denotes that its values are defined by $\mathbf{u}_T$ and $\mathbf{x}_T$ along the trajectory. The corresponding error dynamics are given by

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_T - \dot{\mathbf{x}} = \sum_{i=1}^{r} h_i(\mathbf{z}_{s,T})\left(\mathbf{A}_i\mathbf{x}_T + \mathbf{B}_i\mathbf{u}_T\right) - h_i(\mathbf{z}_s)\left(\mathbf{A}_i\mathbf{x} + \mathbf{B}_i\mathbf{u}\right). \tag{9.4}$$

Partly replacing $\mathbf{x}$ and $\mathbf{u}$ by (9.2), we obtain

$$\dot{\mathbf{e}} = \sum_{i=1}^{r}(h_i(\mathbf{z}_{s,T}) - h_i(\mathbf{z}_s))\left(\mathbf{A}_i\mathbf{x}_T + \mathbf{B}_i\mathbf{u}_T\right) + h_i(\mathbf{z}_s)\left(\mathbf{A}_i\mathbf{e} + \mathbf{B}_i\mathbf{u}_e\right) \tag{9.5a}$$

$$= \underbrace{\sum_{i=1}^{r}(h_i(\mathbf{z}_{s,T}) - h_i(\mathbf{z}_s))\left(\mathbf{A}_i\mathbf{x}_T + \mathbf{B}_i\mathbf{u}_T\right)}_{\dot{\mathbf{e}}_1} + \underbrace{\sum_{i=1}^{r} h_i(\mathbf{z}_s)\left(\mathbf{A}_i\mathbf{e} + \mathbf{B}_i\mathbf{u}_e\right)}_{\dot{\mathbf{e}}_2}. \tag{9.5b}$$

The second part $\dot{\mathbf{e}}_2$ represents a time-invariant error system in T-S notation that could be handled via LMIs. However, that does not hold true for the fist part $\dot{\mathbf{e}}_1$ which amounts to the following question:

Problem 9.1.1. Given a system (2.6) and a desired input and state trajectory $\mathbf{u}_T$ and $\mathbf{x}_T$, respectively. Then the problem is to transform the linear time-variant error dynamics (9.1) into a T-S system such that a tracking controller can be designed based on linear matrix inequalities (LMIs) even if constraints have to be considered.

In the following, two possible ways for solving Problem 9.1.1 are proposed: First, we formulate a local (LO) T-S system without affine terms by locally linearizing the linear time-variant error dynamics along the trajectory. Second we transform the linear time-variant error dynamics into a sector nonlinearity (SE-NL) T-S model. Both T-S representations are obtained by considering the transition time of the trajectory as premise variable. Based on the obtained T-S systems, the developed methods concerning set point tracking (estimating a DA, controller design, GINA controller) are directly applicable to trajectory tracking if a static allocation of each constraint is considered (see Assumption 2.4.1).

9.2 Formulating the Error Dynamics in T-S Notation

One possibility for solving Problem 9.1.1 is to transfer the time-variant linear error dynamics (9.1) into a LO T-S system without affine terms:

Theorem 9.2.1. *Let a desired input and state trajectory $\mathbf{u}_T$ and $\mathbf{x}_T$, respectively, for a nonlinear system be given. Then there exists a LO T-S formulation without affine terms of the tracking error dynamics* (9.1)*.*

Proof: We define r local linear systems $(\mathbf{A}_i, \mathbf{B}_i)$, $i \in \mathbb{N}_{1:r}$ along the desired trajectory by inserting equidistant time instances $t_i \in [t_0, t_e]$, $i \in \mathbb{N}_{1:r}$, of the transition time into (9.1). Based on that the LO T-S formulation of the error dynamics is obtained as

$$\dot{\mathbf{e}} = \sum_{i=1}^{r} h_i(z_{s,T}) \left(\mathbf{A}_i \mathbf{e} + \mathbf{B}_i \mathbf{u}_e\right) \tag{9.6}$$

where the scalar premise variable is $z_{s,T} = t$ with $t \in [t_0, t_e]$. The nonlinear blending functions $h_i(z_{s,T})$ can be chosen according to Definition 2.1.2 whereby the proof is completed. ■

In contrast to Theorem 9.2.1, Problem 9.1.1 can also be solved by establishing a SE-NL T-S model of the error dynamics (instead of a LO T-S system):

Theorem 9.2.2. *Let a desired input and state trajectory $\mathbf{u}_T$ and $\mathbf{x}_T$, respectively, be given. Then there exists a SE-NL T-S formulation of the error dynamics* (9.1)*.*

Proof: Following the sector nonlinearity approach (see Section 2.1.2) for a system (9.1) results in time-invariant error dynamics in form of (9.6), where the scalar premise variable $z_s = t$ equals to time. The extreme values of the time-dependent functions $\theta_k(t)$, $k \in \mathbb{N}_{1:r}$, within the SE-NL T-S formulation (see (2.10)) are given by

$$\overline{\theta}_k(t) = \max(\theta_k(t)), \quad \underline{\theta}_k(t) = \min(\theta_k(t)). \tag{9.7}$$

Thereby, the proof is concluded. ■

Example 9.2.1. In this example, we illustrate Theorem 9.2.2: Assume that a desired state and input trajectory $\mathbf{x}_T$ and u_T, respectively is given. The corresponding system's error dynamics (9.1) is

$$\dot{\mathbf{e}} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & \theta_1(t) \end{bmatrix}}_{A(t)} \mathbf{e} + \underbrace{\begin{bmatrix} 0 \\ \theta(t) \end{bmatrix}}_{b(t)} u_e \tag{9.8}$$

with the time-varying nonlinearities $\theta_1(t) = x_{1,T}^2(t) + 1$ and $\theta_2(t) = x_{1,T}(t) + x_{2,T}(t)$. Let the elements of the state trajectory be within the range $x_{1,T} \in [0, 1]$, $x_{2,T} \in [1, 2]$ for $t \in [t_0, t_e]$. Then the corresponding SE-NL T-S formulation consists of $r = 4$ linear subsystems which are given by the possible compilations of

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ 0 & \underline{\theta}_1(t) \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 0 & \overline{\theta}_1(t) \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 0 & \underline{\theta}_2(t) \end{bmatrix}^T, \quad \mathbf{b}_2 = \begin{bmatrix} 0 & \overline{\theta}_2(t) \end{bmatrix}.^T \tag{9.9}$$

with $\underline{\theta}_1(t) = 1$, $\overline{\theta}_1(t) = 2$ and $\underline{\theta}_2(t) = 1$, $\overline{\theta}_2(t) = 3$. The blending functions are given accordingly.

The obtained T-S models are in the error domain but structurally identical to a conventional T-S system in the state space domain. Hence, it becomes obvious that our results concerning set point tracking (estimating a DA, controller design, GINA controller) are directly applicable to trajectory tracking if each system constraint is subdivided into two parts (see Section 2.4).

9.3 Summary

In this chapter, we have studied the problem of transferring the methods and algorithms derived in the Part I of this thesis (set point racking) to trajectory tracking based on a 2-DOF control structure. To this end, we have first formulated the error dynamics of a nonlinear system in terms of a conventional T-S model considering time as premise variable. It has been shown that both, a LO T-S as well as a SE-NL T-S model can be realized. Thereby, the estimation of a DA and a tracking controller design becomes trivial if a static allocation of the system constraints is considered.

Chapter 10

Limits of Inputs and States are Allocated: LISA Condition

Due to the previous chapter, our results from Part I of this thesis can be directly applied to trajectory tracking problems. However, the required a priori subdivision of system constraints (see Assumption 2.4.1) might lead to reduced performance. This is especially the case concerning input saturation. For example, consider that some time instances exist during tracking where the feedforward part does only require an input amplitude which is much smaller than the allowed maximum, meaning $|\mathbf{u}_T| << |\mathbf{u}_{T,max}|$. The tracking controller however is only allowed to use a defined maximum input amplitude according to (2.50) whereby the input amplitude is not exploited and thus performance for error compensation remains unused.

In this chapter we solve the explained draw back. The main result is the novel *LISA* (Limits of Inputs and States are Allocated) condition which allows an adaptation of the allocated input and state limits within Assumption 2.4.1 depending on the currently required values of the feedforward part. Thereby, we are able to handle time-varying constraints while ensuring the asymptotic stabilization of the desired trajectory.

After a formal problem statement in Section 10.1 we formulate conditions in terms of Lyapunov decay rates which allow to handle the time-varying input and state constraints in Section 10.2. These conditions are relaxed in Section 10.3 which leads directly to the formulation of the final LISA condition is Section 10.4.

10.1 Problem Formulation

Taking the algebraic sign of system constraints according to Assumption 2.4.1 into account each constraint can be formulated in an asymmetric and time-variant manner:

$$\varpi_{e,max}(t) = \varpi_{max} - \varpi_T(t), \tag{10.1a}$$

$$\varpi_{e,min}(t) = -\varpi_{max} - \varpi_T(t) \tag{10.1b}$$

Therein, ϖ_{max} denotes the maximum allowed absolute value of the constraint and $\varpi_T(t)$ represents the currently (time-dependent) required amplitudes of the feedforward part. Consequently, the saturation limits become time-variant:

$$\boldsymbol{\sigma}(\varpi_e, t) = \begin{cases} \varpi_{e,min}(t) & \text{if } \varpi_e \leq \varpi_{e,min}(t), \\ \varpi_{e,max}(t) & \text{if } \varpi_e \geq \varpi_{e,max}(t), \\ \varpi_e & \text{else,} \end{cases} . \tag{10.2}$$

This leads to a change of the size of an estimated domain of attraction (DA)

$$\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t)) = \left\{ \mathbf{e} \in \mathbb{R}^n : V_{\mathbf{0}} = \mathbf{e}^T \mathbf{P} \mathbf{e} \leq \eta_{\mathbf{0}}(t) \right\} \tag{10.3}$$

for error compensation which clearly can cause stability problems. Hence:

Problem 10.1.1. Let the error dynamics in T-S notation (based on Theorem 9.2.1 or Theorem 9.2.2) and a desired trajectory be given such that the system constraints fulfill Assumption 2.4.1 as well as (2.51). Then the problem is to ensure the asymptotic stabilization of the desired trajectory with the time-dependent saturation function (10.2).

The key concept for solving Problem 10.1.1 is sketched in Fig. 10.1 which depicts two DAs according to (10.3) at a time t_1 and $t_2 > t_1$ around the desired trajectory ($\mathbf{e}_d^* = \mathbf{0}$) for a two-dimensional tracking error domain: If the time-varying DA (10.3) shrinks from $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t_1))$ to $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t_2))$ due to the signals required by the feedforward controller, the tracking error has to decrease faster than the DA does such that $\mathbf{e}(t_2) \in \mathcal{E}(\mathbf{P}, \eta_{\mathbf{0}}(2))$. The DAs are depicted by their bounding level values $\partial\mathcal{E}(\mathbf{P}, \eta_{\mathbf{0}}(t_1))$ and $\partial\mathcal{E}(\mathbf{P}, \eta_{\mathbf{0}}(t_2))$. In order to ensure that the tracking error decreases fast enough, we derive the novel LISA condition (Limits of Inputs and States are Allocated) which leads to a required decay rate of the Lyapunov function $V_{\mathbf{0}} = \mathbf{e}^T \mathbf{P} \mathbf{e}$ by analyzing the desired trajectory.

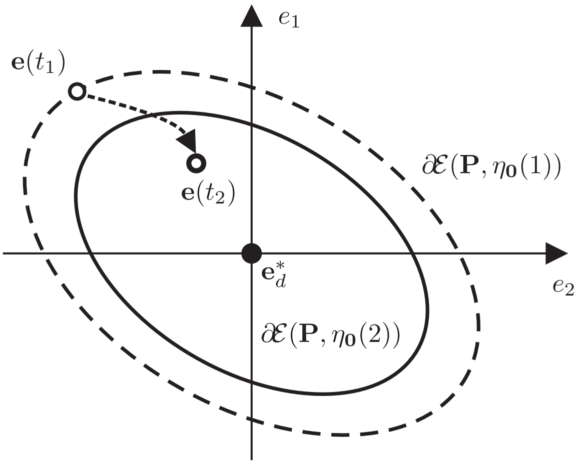

e_1
$\mathbf{e}(t_1)$
$\mathbf{e}(t_2)$
$\partial\mathcal{E}(\mathbf{P}, \eta_{\mathbf{0}}(1))$
$\mathbf{e}_d^*$
e_2
$\partial\mathcal{E}(\mathbf{P}, \eta_{\mathbf{0}}(2))$

Figure 10.1: Operation principle of the LISA condition.

10.2 Decay Rate Conditions

In this section, we do a first step towards an adaptation according to (10.2) by considering the symmetric but time-dependent bounds for the saturation function

$$\varpi_{e,max}(t) = \varpi_{max} - |\varpi_T(t)|\,, \quad \varpi_{e,min}(t) = -\varpi_{e,max}(t). \tag{10.4}$$

Note, equation (2.51) ensures that $\varpi_{e,max}(t)$ is always greater than zero. In Section 10.2.1 we derive conditions such that an input amplitude allocation according to (10.4) is possible. In Section 10.2.2, analogous conditions for the input rate adaptation are computed. These conditions are reformulated in Section 10.2.3 such that a state limit allocation is enabled.

10.2.1 Input Amplitude Allocation

Considering an allocation of the input amplitude according to (10.4), the time-variation of the DA (10.3) depends only on the actual feedforward signal $\mathbf{u}_T(t)$, i.e.

$$\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t)) = \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\mathbf{u}_T(t))) = \left\{\mathbf{e} \in \mathbb{R}^n : V_{\mathbf{0}} = \mathbf{e}^T\mathbf{P}\mathbf{e} \leq \eta_{\mathbf{0}}(\mathbf{u}_T(t))\right\}. \tag{10.5}$$

Consequently, a tracking error $\mathbf{e}$ has to stay within $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\mathbf{u}_T(t)))$ for all $t \in \mathcal{T}$ for ensuring asymptotic stability. Therefore, the following assumption, which is naturally fulfilled by a smooth trajectory design, is stated:

Assumption 10.2.1. The feedforward signal $\mathbf{u}_T$ is a C^1-function and thus differentiable at least.

Based on that assumption, we formulate the following lemma:

Lemma 10.2.1. *Let a system subject to input amplitude saturation* (2.21) *be given for which a bounding level value $\eta_{\mathbf{u}^*}$ at an equilibrium $(\mathbf{x}^*, \mathbf{u}^*)$ can be calculated based on* (6.9)*. Let further a desired trajectory $\mathbf{x}_T$, $\mathbf{u}_T$ of the system (in accordance with* (2.51) *and Assumption 10.2.1) and a T-S formulation of the tracking error dynamics be given. Then there exists an exponential decay rate $\alpha_{\mathbf{u}} > 0$ for $V_{\mathbf{0}}$ in* (10.5)*, meaning*

$$\dot{V}_{\mathbf{0}} \leq -\alpha_{\mathbf{u}} V_{\mathbf{0}}, \quad \alpha_{\mathbf{u}} > 0, \tag{10.6}$$

of the closed-loop error system such that $\mathbf{e}_d^* = \mathbf{0}$ is asymptotically stable for each tracking error $\mathbf{e}(t_1) \in \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\mathbf{u}_T(t_1)))$ occurring at a time $t_1 \in \mathcal{T}$ if the maximal input amplitude is adapted according to (10.4)*.*

Proof: See the Appendix A.4. ■

Concerning closed-loop systems where the bounding level value $\eta_{\mathbf{u}^*}$ of an equilibrium cannot be determined based on (6.9) or a trajectory which does not fulfill Assumption 10.2.1, we propose a different strategy. To this end, we approximate the variation of the bounding level set $\eta_{\mathbf{0}}(\mathbf{u}_T(t))$ and its time derivative $\dot{\eta}_{\mathbf{0}}(\mathbf{u}_T(t))$ similar to the approximation of a nonlinear system by a LO T-S system as follows: First, we discretize the feedforward signal to

$$\mathbf{u}_T(k) := \{\mathbf{u}_T(t) : t = k\Xi\}, \quad t \in \mathcal{T}, \tag{10.7}$$

where $k \in \mathbb{N}_{0:\hat{k}}$ denotes the discretization variable and $\Xi > 0$ is the sampling period. Based on that, we estimate $\eta_{\mathbf{0}}(\mathbf{u}_T(k))$ for each discretized $\mathbf{u}_T(k)$. This can be simply done based on the LMIs developed in Section 3.2. Note, the maximum input amplitude has to be reduced to $\zeta = u_{max,i} - |u_{T,i}(k)|$, $i \in \mathbb{N}_{1:m}$, in the LMI (2.41) in order to account for the fact that $|u_{T,i}(k)|$ is currently not available. The time derivative of $\eta_{\mathbf{0}}(\mathbf{u}_T(k))$ can be approximated by using the central differential quotient to

$$\dot{\eta}_{\mathbf{0}}(u_{T,i}(k)) = \frac{\eta_{\mathbf{0}}(u_{T,i}(k+1)) - \eta_{\mathbf{0}}(u_{T,i}(k-1))}{2\Xi}, \quad 0 < k < \hat{k}. \tag{10.8}$$

The two derivatives $k = 0$ and $k = \hat{k}$ which can not be calculated this way are either approximated by the forward and backward differential quotient, respectively, i.e.

$$\dot{\eta}_{\mathbf{0}}(u_{T,i}(k)) = \begin{cases} \frac{\eta_{\mathbf{0}}(u_{T,i}(k+1)) - \eta_{\mathbf{0}}(u_{T,i}(k))}{\Xi} & \text{if } k = 0, \\ \frac{\eta_{\mathbf{0}}(u_{T,i}(k)) - \eta_{\mathbf{0}}(u_{T,i}(k-1))}{\Xi} & \text{if } k = \hat{k}, \end{cases} \tag{10.9}$$

or defined by the following assumption:

Assumption 10.2.2. The time derivatives of the level values at the beginning and the end of a trajectory are $\dot{\eta}_{\mathbf{0}}(\mathbf{u}_T(0)) = 0$ and $\dot{\eta}_{\mathbf{0}}(\mathbf{u}_T(\hat{k})) = 0$, respectively.

This assumption is equivalent with the statement that $\mathbf{u}_T(t)$ stays constant at the beginning and the end of the trajectory which is the case for lots of practically reasonable trajectories. Most often a trajectory should start and end in a defined equilibrium or a stable set point configuration at least, e.g. each trajectory designed according to the introduced method in Section 8. Based on that, we approximate (A.16) and (A.17) by

$$\eta_{\mathbf{0}}(u_{T,i}(t)) = \mu_\eta(t)\eta_{\mathbf{0}}(u_{T,i}(k)) + (1-\mu_\eta(t))\eta_{\mathbf{0}}(u_{T,i}(k+1)), \tag{10.10a}$$

$$\dot{\eta}_{\mathbf{0}}(u_{T,i}(t)) = \mu_{\dot{\eta}}(t)\dot{\eta}_{\mathbf{0}}(u_{T,i}(k)) + (1-\mu_{\dot{\eta}}(t))\dot{\eta}_{\mathbf{0}}(u_{T,i}(k+1)) \tag{10.10b}$$

for $t \in [k\Xi, (k+1)\Xi]$, $i \in \mathbb{N}_{1:m}$. Thereby, the interpolation parameters $\mu_\eta(t) \in [0,1]$ and $\mu_{\dot{\eta}}(t) \in [0,1]$ grow linearly from $\mu_\eta(k\Xi) = 0$ towards $\mu_\eta((k+1)\Xi) = 1$, and thus similar to the interpolation variables shown in Fig. 5.1. That is analogous to the interpolation between the local linear subsystems of a LO T-S system which enables us to state the following lemma:

Lemma 10.2.2. *Let a system subject to input amplitude saturation* (2.21)*, a desired trajectory* $\mathbf{x}_T$*,* $\mathbf{u}_T$ *(in accordance with* (2.51)*) and a T-S formulation of the tracking error dynamics be given. Then there exists an exponential decay rate* (10.6) *for* $V_{\mathbf{0}}$ *in* (10.5) *concerning the closed-loop error system such that* $\mathbf{e}_d^* = \mathbf{0}$ *is asymptotically stable for each tracking error* $\mathbf{e}(t_1) \in \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\mathbf{u}_T(t_1)))$ *occurring at a time* $t_1 \in \mathcal{T}$ *if the maximal input amplitude is adapted according to* (10.4)*.*

Proof: Reformulating (A.16) and (A.17) based on (10.10) leads to

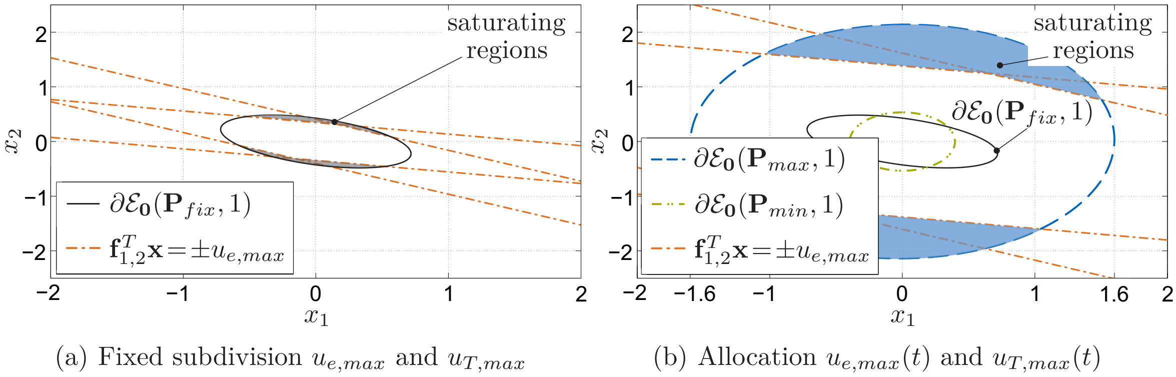
$$\eta_{\mathbf{0}}(\mathbf{u}_T(t)) = \eta_{\mathbf{0}}(\mathbf{u}_{T,\mathbf{p}}(t)) = \min_i \left(\mu_\eta(t)\eta_{\mathbf{0}}(u_{T,i}(k)) + (1-\mu_\eta(t))\eta_{\mathbf{0}}(u_{T,i}(k+1))\right), \tag{10.11}$$

$$\dot{\eta}_{\mathbf{0}}(\mathbf{u}_T(t)) = \min_p \left(\mu_{\dot{\eta}}(t)\dot{\eta}_{\mathbf{0}}(\mathbf{u}_{T,p}(k)) + (1-\mu_{\dot{\eta}}(t))\dot{\eta}_{\mathbf{0}}(\mathbf{u}_{T,p}(k+1))\right), \tag{10.12}$$

respectively. Following the remaining proof of Lemma 10.2.1 results in $\alpha_{\mathbf{u}}$ according to (A.20) which concludes the proof. ■

Example 10.2.1 (Example 3.2.1 cont'd)**.** Assume an error dynamics

$$\dot{\mathbf{e}} = \begin{bmatrix} 0 & 1 \\ 0 & \theta_1(z_s) \end{bmatrix} \mathbf{e} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \sigma(u_e) \tag{10.13}$$



(a) Fixed subdivision $u_{e,max}$ and $u_{T,max}$

(b) Allocation $u_{e,max}(t)$ and $u_{T,max}(t)$

Figure 10.2: Comparison of estimated DAs for trajectory tracking.

with $\theta_1(z_s) = e_1^2 + 1$, a UoD of $e_{1,max} = 1.6$ and to $u_{e,max} = 7$. We also consider the T-S controller (2.47) of the example for compensating a tracking error. Let the required decay rate in (10.6) be $\alpha_u = 1.0$ and the desired input trajectory fulfills $\rho_{\varpi} = 0.75$, which means that according to (2.51) $u_{e,max} \geq 0.25 \cdot u_{max}$ is ensured for the whole trajectory. Analogous to example 3.2.1, we search for an over-saturating DA which are depicted in Fig. 10.2. The bounding level set $\partial\mathcal{E}_{\mathbf{0}}(\mathbf{P}_{fix}, 1)$ in Fig. 10.2(a) has been obtained by statically subdividing the input amplitude between feedforward and error tracking part according to (2.50). Here, $u_{e,max} = 0.25 \cdot u_{max}$ has to be considered in the LMI conditions. Compared to that, the estimation and variation (10.5) of the DA concerning Lemma 10.2.2 is depicted in Fig. 10.2(b). As this DA varies over time, we show its largest and its smallest bounding level set $\partial\mathcal{E}_{\mathbf{0}}(\mathbf{P}_{max}, 1)$ and $\partial\mathcal{E}_{\mathbf{0}}(\mathbf{P}_{min}, 1)$, respectively. Thereby, $\partial\mathcal{E}_{\mathbf{0}}(\mathbf{P}_{max}, 1)$ is obtained setting $u_{e,max} = u_{max}$ and $\partial\mathcal{E}_{\mathbf{0}}(\mathbf{P}_{min}, 1)$ is given by (A.16) with $u_T(t) = 0.75 \cdot u_{max}$ (its maximum value). In order to ease a comparison of the results, we print $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_{fix}, 1)$ in Fig. 10.2(b) additionally. We can see that $\partial\mathcal{E}_{\mathbf{0}}(\mathbf{P}_{max}, 1)$ is noticeably larger than $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_{fix}, 1)$. If the signal of the desired input trajectory comes close its maximum $u_T(t) = 0.75 \cdot u_{max}$ the DA shirks towards $\partial\mathcal{E}_{\mathbf{0}}(\mathbf{P}_{min}, 1)$. In that case, the DA is in some regions smaller than the one calculated with a fixed subdivision of the input amplitude. Consequently, this example clarifies that an allocation of the input amplitude results in a larger DA in general. However, there might exist single situations where a static subdivision of the input leads to a larger DA. In such a event, $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}_{fix}, 1)$ can be activated by switching the active DA.

10.2.2 Input Rate Allocation

In the sequel, we extend the results the previous section to input rate limitations by adjusting the Lemmas 10.2.1 and 10.2.2. Considering an allocation of the input rate according to (10.4) the estimated domain of attraction DA (10.3) varies over time dependent on the feedforward input rate $\dot{\mathbf{u}}_T(t)$. Hence, we write

$$\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t)) = \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\dot{\mathbf{u}}_T(t))) = \left\{ \mathbf{e} \in \mathbb{R}^n : V_{\mathbf{0}} = \mathbf{e}^T \mathbf{P} \mathbf{e} \leq \eta_{\mathbf{0}}(\dot{\mathbf{u}}_T(t)) \right\}. \tag{10.14}$$

For guaranteeing asymptotically stable tracking of the desired trajectory, the tracking error $\mathbf{e}$ has to stay within (10.14) for all $t \in \mathcal{T}$. To this end, we state the following assumption which is naturally fulfilled by a smooth trajectory design:

Assumption 10.2.3. The feedforward signal $\mathbf{u}_T$ is a C^2-function and thus differentiable twice at least.

However, before we are able to adjust Lemma 10.2.1, we state the following corollary for ensuring an adaptation of the DA analogous to (6.9):

Corollary 10.2.1. *Let a system in T-S notation (without affine terms) which is subject to input rate saturation* (3.9) *be given. Let further a desired trajectory* $\mathbf{x}_T$, $\mathbf{u}_T$ *of the system (in accordance with* (2.51) *and Assumption 10.2.1) be given. Then the variation of the DA* (10.14) *depending on* $\dot{\mathbf{u}}_T(t)$*is analogous to* (A.16) *given by*

$$\eta_{\mathbf{0}}(\dot{\mathbf{u}}_T(t)) = \eta_{\mathbf{0}}(\dot{\mathbf{u}}_{T,\mathbf{p}}(t)) = \min_i \left(\underbrace{\eta_{\mathbf{0}} \cdot \frac{(\overbrace{\dot{u}_{max,i} - |\dot{u}_{T,i}(t)|}^{\dot{u}_{e,max,i}(t)})^2}{\dot{u}^2_{max,i}}}_{\eta_{\mathbf{0}}(\dot{u}_{T,i}(t))} \right), \tag{10.15}$$

for all $t \in \mathcal{T}$ *if a linear state feedback law is considered. The index* $i \in \mathbb{N}_{1:m}$ *denotes the elements of the input vector and* $\eta_{\mathbf{0}}$ *is the bounding level value estimated for* $\dot{\mathbf{u}}_T(t) = \mathbf{0}$, *meaning* $\dot{u}_{e,max,i}(t) = \dot{u}_{max,i}$. *The index vector* $\mathbf{p}$ *summarizes the element* $\dot{u}_{T,i}(t)$ *of the input vector that lead to the smallest level value at a certain time, e.g. if* $\eta_{\mathbf{0}}(\dot{\mathbf{u}}_{T,\mathbf{p}}(t_1)) = \eta_{\mathbf{0}}(\dot{u}_{T,2}(t_1)) = \eta_{\mathbf{0}}(\dot{u}_{T,4}(t_1))$ *then* $\mathbf{p} = [2, 4]$.

Proof: The considered T-S model is given by

$$\dot{\hat{\mathbf{x}}} = \sum_{i=1}^{r} h_i(\mathbf{z}_s) \left(\hat{\mathbf{A}}_i \hat{\mathbf{x}} + \hat{\mathbf{B}}\, \boldsymbol{\sigma} \left(\mathbf{K}\hat{\mathbf{x}} + \mathbf{T}\mathbf{v} \right) \right) \tag{10.16}$$

which is obtained when ignoring the input amplitude saturation in (3.11). Based on a linear state feedback law $\mathbf{v} = \mathbf{F}\hat{\mathbf{x}}$, the optimization problem for estimating a DA depending on the available input rate is

$$\eta_{\mathbf{0}}(\dot{u}_{T,i}(t)) = \min_{\hat{\mathbf{x}}} \left(\hat{\mathbf{x}}^T \mathbf{P} \hat{\mathbf{x}} \right), \quad \text{subject to } \mathbf{h}_i^T \hat{\mathbf{x}} = \pm(\dot{u}_{max,i} - |\dot{u}_{T,i}(t)|) \tag{10.17}$$

for every input $i \in \mathbb{N}_{1:m}$. Therein, $\mathbf{h}_i^T$ is the i-th row of the auxiliary controller $\mathbf{H} = \mathbf{K} + \mathbf{TF}$. The solutions of (10.17), which among the extreme values of the related Lagrange function, are

$$\eta_{\mathbf{0}}(\dot{u}_{T,i}(t)) = \eta_{\mathbf{0}} \frac{(\dot{u}_{max,i} - |\dot{u}_{T,i}(t)|)^2}{u_{max}^2}, \quad i \in \mathbb{N}_{1:m}. \tag{10.18}$$

Consequently, a valid estimation of the related DA is bounded by the smallest of these level values whereby (10.15) is obtained and the proof concluded. ■

Based on that result, we can adjust Lemma 10.2.1:

Lemma 10.2.3. *Let a system subject to input rate saturation* (3.9) *be given for which Corollary 10.2.1 holds. Let further a desired trajectory* $\mathbf{x}_T$*,* $\mathbf{u}_T$ *of the system (in accordance with* (2.51) *and Assumption 10.2.3) and a T-S formulation of the tracking error dynamics be given. Then there exists an exponential decay rate* $\alpha_{\dot{\mathbf{u}}} > 0$ *for* $V_{\mathbf{0}}$ *in* (10.14)*, meaning*

$$\dot{V}_{\mathbf{0}} \leq -\alpha_{\dot{\mathbf{u}}} V_{\mathbf{0}}, \quad \alpha_{\dot{\mathbf{u}}} > 0 \tag{10.19}$$

of the closed-loop error system such that $\mathbf{e}_d^* = \mathbf{0}$ *is asymptotically stable for each tracking error* $\mathbf{e}(t_1) \in \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\dot{\mathbf{u}}_T(t_1)))$ *occurring at a time* $t_1 \in \mathcal{T}$ *if the input rate limitation is adapted according to* (10.4)*.*

Proof: The time derivative of all elements in (10.15) is

$$\dot{\eta}_{\mathbf{0}}(\dot{u}_{T,i}(t)) = \frac{2\eta_{\mathbf{0}} \ddot{u}_{T,i}(t)(|\dot{u}_{T,i}(t)| - \dot{u}_{max,i})}{\dot{u}_{max,i}^2} \operatorname{sgn}(\dot{u}_{T,i}(t)), \tag{10.20}$$

whereby the signum function occurs due to the weak differentiability of the absolute value function. From that point, following the proof of Lemma 10.2.1 from equation (A.18) results in

$$\alpha_{\dot{\mathbf{u}}} \geq \underbrace{\frac{-\dot{\eta}_{\mathbf{0}}(\dot{\mathbf{u}}_T(t))}{\eta_{\mathbf{0}}(\dot{\mathbf{u}}_T(t))}}_{\alpha_{\dot{\mathbf{u}}}(t)}, \quad \forall\, t \in \mathcal{T}, \tag{10.21a}$$

$$\alpha_{\dot{\mathbf{u}}} \geq \max\left(\beta, \max_{t \in \mathcal{T}} (\alpha_{\dot{\mathbf{u}}}(t))\right), \tag{10.21b}$$

and thus in a required decay rate which concludes the proof. ■

Concerning the case that Corollary 10.2.1 is not valid (for instance if a T-S controller is considered), a numerical approximation of the bounding level value can be derived analogous to the input amplitude case according to Lemma 10.2.2:

Lemma 10.2.4. *Let a system subject to input amplitude saturation* (2.21)*, a desired trajectory* $\mathbf{x}_T$*,* $\mathbf{u}_T$ *(in accordance with* (2.51)*) and a T-S formulation of the tracking error dynamics be given. Then there exists an exponential decay rate* (10.19) *for* $V_\mathbf{0}$ *in* (10.14) *concerning the closed-loop error system such that* $\mathbf{e}_d^* = \mathbf{0}$ *is asymptotically stable for each tracking error* $\mathbf{e}(t_1) \in \mathcal{E}_\mathbf{0}(\mathbf{P}, \eta_\mathbf{0}(\dot{\mathbf{u}}_T(t_1)))$ *occurring at a time* $t_1 \in \mathcal{T}$ *if the input rate limitation is adapted according to* (10.4)*.*

Proof: First we replace $u_{T,i}$ in (10.10) by $\dot{u}_{T,i}$. In other words, the approximation of the bounding level value (from (10.7) to (10.10)) is transferred from discretizing $\mathbf{u}_T$ to a discretized $\dot{\mathbf{u}}_T$. Following the proof of Lemma 10.2.2 based on that, meaning (10.11) is reformulated to

$$\eta_\mathbf{0}(\dot{\mathbf{u}}_T(t)) = \eta_\mathbf{0}(\dot{\mathbf{u}}_{T,\mathbf{p}}(t)) = \min_i \left(\mu_\eta(t)\eta_\mathbf{0}(\dot{u}_{T,i}(k)) + (1-\mu_\eta(t))\eta_\mathbf{0}(\dot{u}_{T,i}(k+1))\right), \tag{10.22a}$$

$$\dot{\eta}_\mathbf{0}(\dot{\mathbf{u}}_T(t)) = \min_p \left(\mu_{\dot{\eta}}(t)\dot{\eta}_\mathbf{0}(\dot{\mathbf{u}}_{T,p}(k)) + (1-\mu_{\dot{\eta}}(t))\dot{\eta}_\mathbf{0}(\dot{\mathbf{u}}_{T,p}(k+1))\right), \tag{10.22b}$$

results in the required $\alpha_{\dot{\mathbf{u}}}$ which concludes the proof. ■

10.2.3 State Limit Allocation

Considering an allocation of the state limitations according to (10.4) the time-variation of the DA (10.3) depends on the actual state vector of the desired trajectory $\mathbf{x}_T(t)$. Hence, we write

$$\mathcal{E}_\mathbf{0}(\mathbf{P}, \eta_\mathbf{0}(t)) = \mathcal{E}_\mathbf{0}(\mathbf{P}, \eta_\mathbf{0}(\mathbf{x}_T(t))) = \left\{\mathbf{e} \in \mathbb{R}^n : V_\mathbf{0} = \mathbf{e}^T\mathbf{P}\mathbf{e} \leq \eta_\mathbf{0}(\mathbf{x}_T(t))\right\}. \tag{10.23}$$

Consequently, a tracking error $\mathbf{e}$ has to stay within $\mathcal{E}_\mathbf{0}(\mathbf{P}, \eta_\mathbf{0}(\mathbf{x}_T(t)))$ for all $t \in \mathcal{T}$ for ensuring asymptotically stable tracking. To this end, we transfer the results concerning input rate saturation of Section 10.2.2:

Corollary 10.2.2. *Let a system in T-S notation (without affine terms) which is subject to state limitations be given. Let further a desired trajectory* $\mathbf{x}_T$*,* $\mathbf{u}_T$ *of the system (in accordance with* (2.51) *and Assumption 10.2.1) be given. Then the variation of the*

DA (10.23) *depending on* $\mathbf{x}_T(t)$ *is analogous to* (A.16) *given by*

$$\eta_{\mathbf{0}}(\mathbf{x}_T(t)) = \eta_{\mathbf{0}}(\mathbf{x}_{T,\mathbf{p}}(t)) = \min_i \left(\underbrace{\eta_{\mathbf{0}} \cdot \frac{(\overbrace{x_{max,i} - |x_{T,i}(t)|}^{x_{e,max,i}(t)})^2}{x_{max,i}^2}}_{\eta_{\mathbf{0}}(x_{T,i}(t))} \right), \tag{10.24}$$

for all $t \in \mathcal{T}$ *if a linear state feedback law is considered. The index* $i \in \mathbb{N}_{1:n}$ *denotes the elements of the state vector and* $\eta_{\mathbf{0}}$ *is the bounding level value estimated for* $\mathbf{x}_T(t) = \mathbf{0}$*, meaning* $x_{e,max,i}(t) = x_{max,i}$*. The index vector* $\mathbf{p}$ *summarizes the element* $x_{T,i}(t)$ *of the state vector which lead to the smallest level value at a certain time, e.g. if* $\eta_{\mathbf{0}}(\mathbf{x}_{T,\mathbf{p}}(t_1)) = \eta_{\mathbf{0}}(x_{T,2}(t_1)) = \eta_{\mathbf{0}}(x_{T,4}(t_1))$ *then* $\mathbf{p} = [2, 4]$.

Proof: The optimization problem

$$\eta_{\mathbf{0}}(x_{T,i}(t)) = \min_{\mathbf{x}} \left(\mathbf{x}^T \mathbf{P} \mathbf{x} \right), \quad \text{subject to } \mathbf{g}_i^T \mathbf{x} = \pm(x_{max,i} - |x_{T,i}(t)|) \tag{10.25}$$

for every state $i \in \mathbb{N}_{1:n}$ defines the bounding level value of the related DA depending on the variation of the required state amplitude of the state trajectory. Therein, $\mathbf{g}_i^T$ equals to (3.6). The solutions of (10.25), which among the extreme values of the related Lagrange function, are

$$\eta_{\mathbf{0}}(x_{T,i}(t)) = \eta_{\mathbf{0}} \frac{(x_{max,i} - |x_{T,i}(t)|)^2}{x_{max}^2}, \quad i \in \mathbb{N}_{1:n} \tag{10.26}$$

whereby the proof is completed. ■

This result allows us to adjust Lemma 10.2.1 (analogous to Lemma 10.2.3):

Lemma 10.2.5. *Let a system subject to state limitations be given for which Corollary 10.2.2 holds. Let further a desired trajectory* $\mathbf{x}_T$*,* $\mathbf{u}_T$ *of the system (in accordance with* (2.51) *and Assumption 10.2.1) and a T-S formulation of the tracking error dynamics be given. Then there exists an exponential decay rate* $\alpha_{\mathbf{x}} > 0$ *for* $V_{\mathbf{0}}$ *in* (10.23)*, meaning*

$$\dot{V}_{\mathbf{0}} \leq -\alpha_{\mathbf{x}} V_{\mathbf{0}}, \quad \alpha_{\mathbf{x}} > 0 \tag{10.27}$$

of the closed-loop error system such that $\mathbf{e}_d^* = \mathbf{0}$ *is asymptotically stable for each tracking error* $\mathbf{e}(t_1) \in \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\mathbf{x}_T(t_1)))$ *occurring at a time* $t_1 \in \mathcal{T}$ *if the state constraints are adapted according to* (10.4).

Proof: The time derivative of all elements in (10.24) is

$$\dot{\eta}_{\mathbf{0}}(x_{T,i}(t)) = \frac{2\eta_{\mathbf{0}}\dot{x}_{T,i}(t)(|x_{T,i}(t)| - x_{max,i})}{x_{max,i}^2}\,\mathrm{sgn}(x_{T,i}(t)), \tag{10.28}$$

whereby the signum function occurs due to the weak differentiability of the absolute value function. Note that due to Assumption 10.2.1 the input signal and thus the state vector is differentiable. From that point, following the proof of Lemma 10.2.1 from equation (A.18) results in

$$\alpha_{\mathbf{x}} \geq \underbrace{\frac{-\dot{\eta}_{\mathbf{0}}(\mathbf{x}_T(t))}{\eta_{\mathbf{0}}(\mathbf{x}_T(t))}}_{\alpha_{\mathbf{x}}(t)}, \quad \forall\, t \in \mathcal{T}, \tag{10.29a}$$

$$\alpha_{\mathbf{x}} \geq \max\left(\beta, \max_{t\in\mathcal{T}}\left(\alpha_{\mathbf{x}}(t)\right)\right), \tag{10.29b}$$

and thus in a required decay rate which concludes the proof. ■

Concerning the case that Corollary 10.2.2 is not valid (for instance if a T-S controller is considered), a numerical approximation of the bounding level value can be derived analogous to input amplitude case according to Lemma 10.2.2:

Lemma 10.2.6. *Let a system subject to state limitations, a desired trajectory* $\mathbf{x}_T$*,* $\mathbf{u}_T$ *(in accordance with* (2.51)*) and a T-S formulation of the tracking error dynamics be given. Then there exists an exponential decay rate* (10.27) *for* $V_{\mathbf{0}}$ *in* (10.23) *concerning the closed-loop error system such that* $\mathbf{e}_d^* = \mathbf{0}$ *is asymptotically stable for each tracking error* $\mathbf{e}(t_1) \in \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\dot{\mathbf{u}}_T(t_1)))$ *occurring at a time* $t_1 \in \mathcal{T}$ *if the state constraints are adapted according to* (10.4).

Proof: First we replace $u_{T,i}$ in (10.10) by $x_{T,i}$. In other words, the approximation of the bounding level value (from (10.7) to (10.10)) is transferred from discretizing $\mathbf{u}_T$ to a discretized $\mathbf{x}_T$. Following the proof of Lemma 10.2.2 based on that, meaning (10.11) is reformulated to

$$\eta_{\mathbf{0}}(\mathbf{x}_T(t)) = \eta_{\mathbf{0}}(\mathbf{x}_{T,\mathbf{p}}(t)) = \min_i\left(\mu_\eta(t)\eta_{\mathbf{0}}(x_{T,i}(k)) + (1-\mu_\eta(t))\eta_{\mathbf{0}}(x_{T,i}(k+1))\right), \tag{10.30a}$$

$$\dot{\eta}_{\mathbf{0}}(\mathbf{x}_T(t)) = \min_p\left(\mu_{\dot{\eta}}(t)\dot{\eta}_{\mathbf{0}}(\mathbf{x}_{T,p}(k)) + (1-\mu_{\dot{\eta}}(t))\dot{\eta}_{\mathbf{0}}(\mathbf{x}_{T,p}(k+1))\right), \tag{10.30b}$$

results in the required $\alpha_{\mathbf{x}}$ which concludes the proof. ■

10.3 Relaxing the Allocation Conditions

If the nominal bounding level value $\eta_{\mathbf{0}}$ of $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t))$ of the time-varying DA is estimated such that stable over-saturation is possible then the allocation logic (10.4) can be adjusted to the theoretically possible maximum (10.1). Indeed, over-saturation makes no sense at all in case of state constraints and thus we consider only input amplitude and rate saturation:

Theorem 10.3.1. *Let a system subject to input amplitude and rate saturation be given. Let further an over-saturating DA* (10.5) *and* (10.14) *which either fulfills Lemma 10.2.1 and 10.2.3 or Lemma 10.2.2 and 10.2.4, respectively, be given. Then the asymptotic stability of* $\mathbf{e}_d^* = \mathbf{0}$ *is still guaranteed if the allocation* (10.4) *is relaxed to* (10.1)*.*

Proof: In case of an over-saturating DA,

$$|\mathbf{u}_e| \geq \mathbf{u}_{max} - |\mathbf{u}_T(t)|\,, \tag{10.31a}$$

$$|\dot{\mathbf{u}}_e| \geq \dot{\mathbf{u}}_{max} - |\dot{\mathbf{u}}_T(t)| \tag{10.31b}$$

is fulfilled if saturation is ignored, at least in some regions within the DA. Obviously, the asymptotic stabilization of $\mathbf{e}_d^* = \mathbf{0}$ is still guaranteed for every error $\mathbf{e}$ within the DA. More precisely, as long as the relaxed bounds (10.2) are within the DA, meaning

$$|\mathbf{u}_e| \geq \boldsymbol{\sigma}(\mathbf{u}_e, t) \geq \mathbf{u}_{max} - |\mathbf{u}_T(t)|\,, \tag{10.32a}$$

$$|\dot{\mathbf{u}}_e| \geq \boldsymbol{\sigma}(\dot{\mathbf{u}}_e, t) \geq \dot{\mathbf{u}}_{max} - |\dot{\mathbf{u}}_T(t)| \tag{10.32b}$$

$\mathbf{e}_d^* = \mathbf{0}$ is asymptotic stabilized. If the bounds are not within the DA and thus

$$\boldsymbol{\sigma}(\mathbf{u}_e, t) \geq |\mathbf{u}_e| \geq \mathbf{u}_{max} - |\mathbf{u}_T(t)|\,, \tag{10.33a}$$

$$\boldsymbol{\sigma}(\dot{\mathbf{u}}_e, t) \geq |\dot{\mathbf{u}}_e| \geq \dot{\mathbf{u}}_{max} - |\dot{\mathbf{u}}_T(t)| \tag{10.33b}$$

is the case then the asymptotic stabilization of $\mathbf{e}_d^* = \mathbf{0}$ is still guaranteed as for every $\mathbf{e}$ within the DA. In other words, if the bounding level value of the DA does not allow the maximum allocation the over-saturating will get us as close as possible towards that allocation which concludes the proof. ■

10.4 The LISA Condition

In this section, we merge all of the considered variation possibilities of an estimated DA $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t))$ to a single decay rate condition – the *LISA* condition (Limits of Inputs and States are Allocated):

Theorem 10.4.1. *Let a system subject to input amplitude and/or input rate and/or state limitations be given. Let further a desired trajectory* $\mathbf{x}_T$, $\mathbf{u}_T$ *and a T-S formulation of the tracking error dynamics be given. If the asymptotic stability of* $\mathbf{e}_d^* = \mathbf{0}$ *is guaranteed for each limitation separability, meaning by one of the Lemmas 10.2.5 and 10.2.6 concerning only state limitation and Theorem 10.3.1 for input amplitude or rate saturation. Then there exists an exponential decay rate* $\alpha_L > 0$ *for* $V_{\mathbf{0}}$ *in* (10.3) *of the closed-loop error system such that*

$$\dot{V}_{\mathbf{0}} \leq -\alpha_L V_{\mathbf{0}}, \quad \alpha_L > 0 \tag{10.34}$$

and $\mathbf{e}_d^* = \mathbf{0}$ *is asymptotically stabilized for the combination of all constraints.*

Proof: The required decay rates are: $\alpha_{\mathbf{u}}$ for input amplitude saturation (according to one of the Lemmas 10.2.1 and 10.2.2), $\alpha_{\dot{\mathbf{u}}}$ for input rate saturation (according to one of the Lemmas 10.2.3 and 10.2.4) and $\alpha_{\mathbf{x}}$ for state saturation (according to one of the Lemmas 10.2.5 and 10.2.6). Hence, the largest of them (highest decay rate)

$$\alpha_L = \max\left(\alpha_{\mathbf{u}}, \alpha_{\dot{\mathbf{u}}}, \alpha_{\mathbf{x}}\right) \tag{10.35}$$

fulfills all of the required restrictions whereby the proof is completed. ■

10.5 Summary

The main contribution of this chapter has been the *LISA* condition (Limits of Inputs and States are Allocated) which is, up to the author's knowledge, the first approach that allows a dynamical shifting of system constraints between the feedforward and the feedback part of a 2-DOF controller. More precisely, we have developed a condition in form of a required exponential decay rate of a Lyapunov function which ensures the asymptotic stabilization of the desired trajectory when adapting the saturation limits for error compensation depending on the current feedforward signal. The required decay rate and a related domain of attraction (DA) has been shown to be numerically computable by LMI conditions based on our results of Section 3.

Chapter 11

The LISA-GINA Control Framework

In this chapter, we extend the LISA condition of the previous chapter such that asymptotic stability of the desired trajectory can be ensured even if the state vector is outside of the estimated DA. To this end, the GINA controller of Chapter 6 is considered for trajectory tracking. The novel *LISA-GINA control framework* allows thus an adaptation of the saturation functions (input amplitude, rate and state limit) for error compensation depending on the required feedforward signals. In addition, an asymptotic stabilization of the desired trajectory (LISA condition) is guaranteed for every error that can be stabilized based on the GINA controller. Moreover, we establish a switching between different LISA-GINA controllers to relax the required LMI conditions and to increase the control performance.

After stating the problem in Section 11.1, the framework is derived in Section 11.2. The switching and smooth switching conditions that we have devolved for the purpose of set point tracking (see the Chapters 4 and 5) can be also beneficially integrated in the LISA-GINA control framework, which is shown in Section 11.3.

11.1 Problem Formulation

The DA within which the asymptotic stabilization of a desired trajectory can be ensured based on the LISA condition (see Section 10.4) is restricted to a time-varying ellipsoid (10.3) around the desired equilibrium $\mathbf{e}_d^* = \mathbf{0}$. This fact brings us to the following main problem to be investigated in this chapter:

Problem 11.1.1. Asymptotic stabilization of a desired trajectory based on the LISA condition of Section 10.4 if the tracking error is outside of the estimated DA, meaning $\mathbf{e} \notin \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t))$.

The key idea for solving Problem 11.1.1 is in attaching the GINA controller to the closed-loop trajectory tracking controller analogous to the set point case (see Section 6). In other words, we extend the domain of attraction (DA) by adding the GINA controller. Hence, every error that can be stabilized based on the GINA controller is asymptotically shifted to the desired trajectory. This leads to the novel LISA-GINA control framework which combines the advantages of the LISA condition and the GINA controller for trajectory tracking tasks.

Beside the size of the DA, the required number of LMI conditions for designing a tracking controller based on the LISA condition might lead to problems. This is due to the fact that the number of LMIs depends on the desired trajectory: First, if the trajectory transition time is long then the amount of linear subsystems in the T-S error formulation according to Section 9.2 that are required to approximate the error dynamics increases. Second, a desired trajectory might lead to subsystems of the T-S error formulation which differ such that the solvability of the LMI optimization is not guaranteed. Hence, a second problem to be addressed is:

Problem 11.1.2. Relaxation of the LMI conditions for deriving a tracking controller based on the LISA condition.

We tackle this problem by transferring our results concerning switched and smoothly switched controller design from Chapter 4 and 5. Putting the major benefit in a nutshell: The number of LMIs is subdivided into several convex optimization problems. Each of these problems results in a valid domain of attraction (DA) for a piece of the desired trajectory. The stabilization of the desired trajectory is ensured by switching between the DAs in a certain manner.

11.2 The Framework

In this section, we combine the LISA condition with the GINA controller to solve Problem 11.1.1. The final *LISA-GINA control framework* is summarized in Framework 11.2.1. Each step of the framework will be explained in the following:

Framework 11.2.1 LISA-GINA Control Framework

```
1: generate trajectory (feedforward control)
2: compute LISA condition
3: estimate DA and design tracking controller
4: compute GINA controller
```

11.2.1 Generate Trajectory and Compute LISA Condition

Initially in step 1 of the LISA-GINA control framework 11.2.1, a desired state and input trajectory $(\mathbf{x}_T(t), \mathbf{u}_T(t))$ hat to be generated. Note that depending on the relevant system constraints, the feedforward signal $\mathbf{u}_T(t)$ has to fulfill the corresponding Assumptions 2.4.1, 10.2.1 and 10.2.3. In other words, every commonly known feedforward design method can be used for designing a proper trajectory, e.g. flatness-based design (see Chapter 8). In step 2 of the framework, the LISA condition (see Theorem 10.4.1) has to be calculated to obtain the required decay rate α_L of the Lyapunov function.

11.2.2 Estimate DA and Design Tracking Controller

Based on step 2, an efficiently estimate of a DA $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t))$ that fulfills the LISA condition is done in step 3 by convex optimization subject to LMI constraints. An appropriate tracking controller needs either to be a priori designed, e.g. by LMI optimization, pole placement, or together with the estimate of the DA (see Example 2.2.3).

Remark 11.2.1. Note that a pre-designed tracking controller does not necessarily means that a Lyapunov function with the required decay rate (10.19) exists. This can be bypassed by designing a LMI-based LQR controller with a pre-defined decay rate such that a common Lyapunov function exists [12, 40]. For instance, in paper [34] such a controller has been derived for linear systems. The developed Riccati equation can be directly used for T-S systems within a LMI optimization.

Due to the required α_L, the estimated DA (10.3) can become smaller at a certain $t \in \mathcal{T}$ than a specific $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}})$ with a static allocation of the constraints. This disadvantage can be solved by combining both control concepts and thus combining their resulting domains of attraction. This has been already investigated in Fig. 10.2 of Example 10.2.1. For the sake of completeness, we summarize the results in form of a proposition in the following:

Proposition 11.2.1. *Let estimated DAs $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}})$ and $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t))$ for tracking error compensation with a static allocation of the constraints according to* (2.50) *and Theorem 10.4.1, respectively, be given. Then a valid DA is given by the switching condition*

$$\mathcal{E}_{\mathbf{0}}(\mathbf{P}, t) = \begin{cases} \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t)) & \textit{if } \mathbf{e}(t) \in \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t)), \\ \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}) & \textit{else} \end{cases} \tag{11.1}$$

if saturation limitations are switched simultaneously to the active controllers.

Proof: Both DAs in (11.1) ensure the asymptotically stable compensation of a tracking error. Switching between them, and thus switching between two different tracking controllers, will only happen once, namely from $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}})$ towards $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t))$ as soon as the tracking error is within the DA obtained by the LISA condition. In fact, the asymptotic stability of the switched error system is guaranteed because a single switching between asymptotically stable systems results in an asymptotically stable overall behavior [80] whereby the proof is completed. ■

Obviously, the DA (11.1) is at least as large as both individual DAs.

11.2.3 Compute GINA Controller

Concerning a trajectory tracking controller design according to the LISA condition, stability can not be ensured if $\mathbf{e} \notin \mathcal{E}_{\mathbf{0}}(\mathbf{P}\eta_{\mathbf{0}}(t))$. We will tackle that problem by adding the GINA controller in step 4 of the framework. To this end, we state the following assumption:

Assumption 11.2.1. If the system is subject to input amplitude and state constraints with $\mathbf{u}^*_{e,r} \neq \mathbf{u}^*_{e,t} \neq \mathbf{u}^*_{e,d} = \mathbf{0}$ and $\mathbf{e}^*_r \neq \mathbf{e}^*_t \neq \mathbf{e}^*_d = \mathbf{0}$, respectively, then the operating space of the GINA controller is defined by a set $\mathcal{M}^*_r$ of reference equilibria that fulfills Assumption 2.4.1.

The reason therefore is that $u_{e,max,i}$ in equation (A.16) and $x_{e,max,i}$ in (10.24) needs to be updated according to the following proposition:

Proposition 11.2.2. *If the set $\mathcal{M}^*_r$ according to Assumption 11.2.1 is not empty then*

$$u_{e,max,i} = u_{max,i} - \left(|u_{T,i}(t)| + \left|u^*_{e,r,max,i}\right|\right), \tag{11.2a}$$

$$x_{e,max,i} = x_{max,i} - \left(|x_{T,i}(t)| + \left|x^*_{e,r,max,i}\right|\right) \tag{11.2b}$$

has to be considered in (A.16) *and* (10.24)*, respectively, for calculating the exponential decay rates. In* (11.2)*, $u^*_{e,r,max,i}$ and $x^*_{e,r,max,i}$ are the i-th components of the steady-state input and state values of all $\mathbf{e}^*_r \in \mathcal{M}^*_r$ such that $u_{e,max,i}$ and $x_{e,max,i}$ become minimal.*

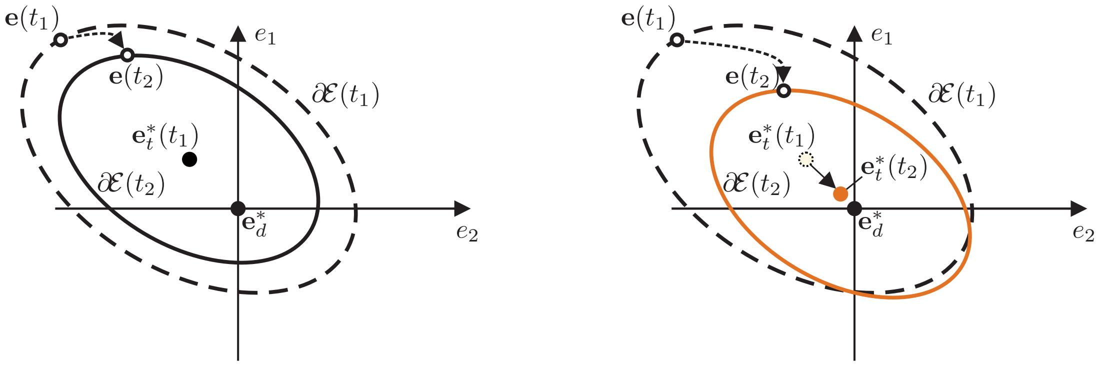
Proof: According to (A.16) the minimal bounding level value has to be considered for deriving the required exponential decay rate. The required steady-state input signal for stabilizing a reference equilibrium $\mathbf{e}^*_r$ reduces the available input amplitude and thus the DA. Consequently, this reduction is taken into account by (11.2a). The same argumentation holds for (10.24) and thus (11.2b) for state restriction which concludes the proof. ■

Note that Assumption 11.2.1 is fulfilled concerning input rate limitation. Based on Proposition 11.2.2, we can state the following theorem:

Theorem 11.2.1. *Let a trajectory tracking controller according to the LISA condition (Theorem 10.4.1) be given. Then extending the closed-loop error system with a GINA controller asymptotically stabilizes* $\mathbf{e}_d^* = \mathbf{0}\ \forall\ \mathbf{e} \in \bigcup_{\mathbf{e}^*} \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(t))$ *if Proposition 11.2.2 is taken into account for deriving the exponential decay rate* α_L *such that*

$$\dot{V}_{\mathbf{0}} < -\alpha_L V_{\mathbf{0}}, \quad \alpha_L > 0. \tag{11.3}$$

Proof: A temporary equilibrium in the error domain generally requires a state-state input signal $\mathbf{u}_{e,t}^* \neq \mathbf{0}$ and also $\mathbf{e}_t^* \neq \mathbf{0}$ differs from zero. Thus, the smallest DA within a pre-defined operation space of the GINA controller is given for a maximal considered error according to Proposition 11.2.2. The reason why "$\leq$" in equation (10.34) of Theorem 10.4.1 has to be replaced by a strict " $<$ " (see (11.3)) can be easily understood based on Fig. 11.1 which shows the most critical case concerning stability: The tracking error is on the bounding level set $\partial\mathcal{E}(t_1)$ of a temporary equilibrium $\mathbf{e}_t^*(t_1)$ calculated by the GINA controller at a time $t_1 \in \mathcal{T}$. Suppose the feedforward signal currently changes in a way such that the DA shrinks, meaning $\partial\mathcal{E}(t_2) < \partial\mathcal{E}(t_1)$, with $t_2 > t_1$. Fig. 11.1(a) illustrates the case if $\dot{V}_{\mathbf{0}} = -\alpha_L V_{\mathbf{0}}$ which might happen if "$\leq$" is considered. Here, $\mathbf{e}(t_2)$ will be again at the bounding level value. Hence, the GINA controller cannot shift the temporary equilibrium towards the desired one $\mathbf{e}_d^* = \mathbf{0}$ whereby asymptotic stability is lost. Roughly speaking, the DA shrinks as much as the error gets closer to $\mathbf{e}_t^*$. If however, (11.3) is considered and thus the decay rate is smaller than α_L then as depicted in Fig. 11.1(b) the error shrink faster as the DA does. Thus, $\mathbf{e}_t^*$ can be scheduled towards the desired equilibrium ($\mathbf{e}_t^*(t_1) \rightarrow \mathbf{e}_t^*(t_2)$) which completes the proof. ■



(a) $\dot{V}_{\mathbf{0}} = -\alpha_L V_{\mathbf{0}}$

(b) $\dot{V}_{\mathbf{0}} < -\alpha_L V_{\mathbf{0}}$

Figure 11.1: Most critical stability case for LISA-GINA.

11.3 Extension to Switched Control

Subsequently, we will tackle Problem 11.1.2 by transferring our results concerning switched and smoothly switched controller design for set point tracking (see the Chapters 4 and 5). The main idea behind the following approach is to subdivide a single trajectory tracking problem into several ones and switch between them:

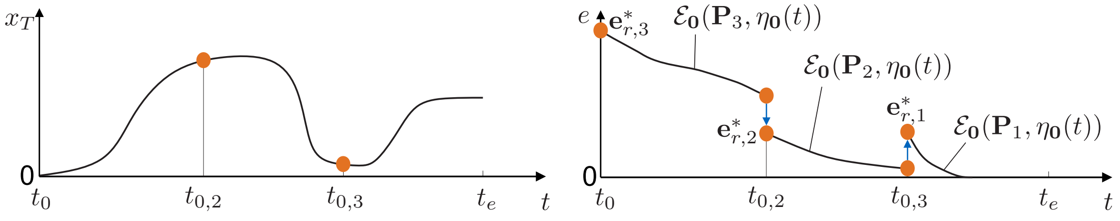
Corollary 11.3.1. *Let a system and a desired trajectory be given such that a GINA-LISA trajectory tracking controller (Theorem 11.2.1) is theoretically possible. Then* $\mathbf{e}_d^* = \mathbf{0}$ *is attractive if* $q \in \mathbb{N}_{1:y}$ *switched LISA-GINA controllers are designed for the error dynamics based on our switching and smooth switching conditions from the Chapters 4 and 5, respectively. The switching signal* l *is:*

$$l = q \begin{cases} \mathcal{T}_q \in t \in [t_{0,q}, t_{0,q+1}[& \text{if } q =\in \mathbb{N}_{2:y}, \\ \mathcal{T}_q \in t \in [t_{0,q}, t_e] & \text{if } q = 1, \end{cases} \tag{11.4}$$

with $t_{0,y} = t_0$ *and* $\mathcal{T}_q$ *denotes the time interval of the desired trajectory within which a certain LISA-GINA controller and its corresponding DA* $\mathcal{E}_\mathbf{0}(\mathbf{P}_q, \eta_\mathbf{0}(t))$ *is active.*

Proof: The switching condition (11.4) implies that the LISA-GINA controllers and their corresponding DAs are ordered along the trajectory in the reverse direction of q. Each $\mathcal{E}_\mathbf{0}(\mathbf{P}_q, \eta_\mathbf{0}(t))$ is either valid for an amount of subsequent linear subsystems (local linear T-S formulation, see Theorem 9.2.1) or for subsequent SE-NL T-S models (see Theorem 9.2.2) along the trajectory. Consequently, analogous to the Chapters 4 and 5, (smooth) switching switching, respectively, do only take place between two subsequent DAs $\mathcal{E}_\mathbf{0}(\mathbf{P}_q, \eta_\mathbf{0}(t))$ and $\mathcal{E}_\mathbf{0}(\mathbf{P}_{q+1}, \eta_\mathbf{0}(t))$. However, contrary to the theorems, corollaries and propositions of the Chapters 4 and 5, a manual design of the switching signal is not allowed here as the activation of the subsystems (and thus the controllers) is scheduled by the transition time. Hence, the GINA controller might have to calculate a new reference equilibrium $\mathbf{e}_{r,q}^*$ after each switch whereby the error $\mathbf{e}$ can increase. For that reason, the desired trajectory is still attractive whereby the proof is concluded. ■

The argumentation of the proof is illustrated in Fig. 11.2 considering a scalar system and three switched LISA-GINA controllers which are denoted by their corresponding DAs $\mathcal{E}_\mathbf{0}(\mathbf{P}_q, \eta_\mathbf{0}(t))$, $q \in \mathbb{N}_{1:3}$. Fig. 11.2(a) sketches a desired state trajectory x_T and Fig. 11.2(b) shows the corresponding tracking error. In both figures the switching intervals (11.4) are depicted. Starting from an initial tracking error, the first active LISA-GINA controller $\mathcal{E}_\mathbf{0}(\mathbf{P}_3, \eta_\mathbf{0}(t))$ asymptotically decreases the error from a corresponding reference equilibrium $\mathbf{e}_{r,3}^*$ towards zero. This is due to the LISA-GINA controller design



(a) Desired trajectory x_T

(b) Value of the error $x_T - x$

Figure 11.2: Attraction of $\mathbf{e}_d^* = \mathbf{0}$ by a switched LISA-GINA tracking controller.

based on Theorem 11.2.1. Due to a switching event to the next LISA-GINA controller with $\mathcal{E}_\mathbf{0}(\mathbf{P}_2, \eta_\mathbf{0}(t))$ at a time $t_{0,2}$, a new reference equilibrium might be required due to the change of the active controller. Hence, a jump in the error signal can occur whereby $\mathbf{e}$ either optimally decreases (see $t_{0,2}$) or increases as shown for the final switch at $t_{0,3}$. After each switch the tracking error decreases again asymptotically (Theorem 11.2.1) and thus the attraction of $\mathbf{e}_d^* = \mathbf{0}$ is ensured as the amount of switching is bounded.

11.4 Summary

In this chapter, the *LISA-GINA* controller framework has been developed whereby a trajectory tracking controller has been derived that allows an adaption of the input amplitude, rate and state limitations based on the LISA condition (Section 10.4) within a large domain of attraction (DA) (the GINA controller). The framework consists of five steps, starting from the generation of the desired trajectory to the design of the GINA controller for ensuring the asymptotic stabilization of the trajectory based on the LISA condition in the large. In form of an optional extension, a switched LISA-GINA trajectory tracking controller has been established. This controller allows to split up the LMI based optimization for fulfilling the LISA condition along the whole trajectory into several optimization problems each for a part of the trajectory. Each of them requires a reduced number of LMI conditions compared to the original problem. Thereby, the general feasibility of the original optimization problem has been enlarged on the one hand and the required numerical effort for finally solving it has been reduced on the other hand.

Chapter 12

Application and Experimental Results

In this chapter, we apply the methods developed in Part II of this thesis to three technical systems subject to input amplitude, rate and state constraints. We present both, simulation as well as experimental results in order to highlight the advantages and the practical applicability of our methods. In the Sections 12.1 and 12.2 we recall the inverted pendulum on cart and the Ballbot system, respectively, which have been already investigated in Chapter 7. In the third example (Section 12.3), we focus on a so-called *VTOL* (vertical take-off and landing) aircraft system.

Concerning all examples, the nominal level value is set to $\eta_{\mathbf{0}} = 1$ and the LMIs are solved by using the YALMIP Toolbox. The GINA controller is always implemented in its optimal form as explained in Section 6.3.

12.1 Inverted Pendulum

> In this section, the LISA-GINA control framework is verified (in simulation and experiment) based on the well-known inverted pendulum on cart system subject to input amplitude saturation. We highlight the advantage of the LISA condition by a comparison with a conventional tracking controller. In addition, the robustness of the GINA controller by shifting a desired trajectory while already tracking it is demonstrated. Noticeable over-saturation is achieved even in the experiment.

The GINA controller is optimally implemented according to Algorithm 6.3.1. The applied calculation for estimating an over-saturating DA for tracking error compensation (according to Section 3.2.2) is detailed in [34]. We consider the linearized pendulum system (linearized around the upper equilibrium $\mathbf{x}^* = [0, 0, 0, 0]^T$)

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \frac{3gm}{m+4M} & 0 & -\frac{4d_x}{m+4M} & 0 \\ \frac{3gm+3gM}{a(m+4M)} & 0 & -\frac{3d_x}{a(m+4M)} & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ \frac{4c_m}{m+4M} \\ \frac{3c_m}{a(m+4M)} \end{bmatrix} \sigma(u), \tag{12.1}$$

subject to input amplitude saturation. For this system, we define the polynomial desired trajectory

$$y_f(t) = \sum_{i=0}^{9} p_i \cdot t^i \tag{12.2}$$

based on the flat output $y_f = x_2 - \frac{4}{3} a x_1$.

Simulation – Benefit of the LISA-GINA Control Framework: According to the LISA-GINA control framework a desired trajectory is generated first. We consider a movement of the flat output from $y_f(t = 0) = 0$ towards $y_f(t = 8) = 1$ and backwards again. Setting the input limitation to $u_{max} = 12\,\text{V}$, the design trajectory fulfills Assumption 2.51 with $\rho_{\varpi} = 0.7$. The second step of the LISA-GINA control framework leads to a required decay rate rate (10.6) of $\alpha_L = 1$ for fulfilling the LISA condition. We choose a LQR with the performance measure

$$J = \int_0^\infty \mathbf{e}^T \mathbf{Q}_{LQR,2} \mathbf{e} + u_e R_{LQR} u_e \text{ dt} \tag{12.3}$$

for controlling the error dynamics of the pendulum system (step 3 of the Framework 11.2.1). The matrices $\mathbf{Q}_{LQR,2}$ and R are given in Table 7.1. The obtained LISA-GINA controller is compared to a static input allocation ($u_{e,max} = 3.6\,\text{V}$) considering the same LQR. However, the simulation became unstable due to the high considered initial position error of 1.2 m, which leads to input amplitudes over 30 V ($u_{max} = 12\,\text{V}$). Hence, we had to integrate the GINA controller also to the LQR with static input allocation in order to regain stability. Fig. 12.1 shows the final obtained simulation results:

The LISA-GINA controller is able to compensate the tracking error in almost 50% of the time required by the controller with static input allocation. The reason for that becomes clear when analyzing the first 8 s of the simulation. The controller with static input allocation is only able to compensate the tracking error with its allocated input amplitude. Compared to that, the LISA-GINA controller adapts the allocated input

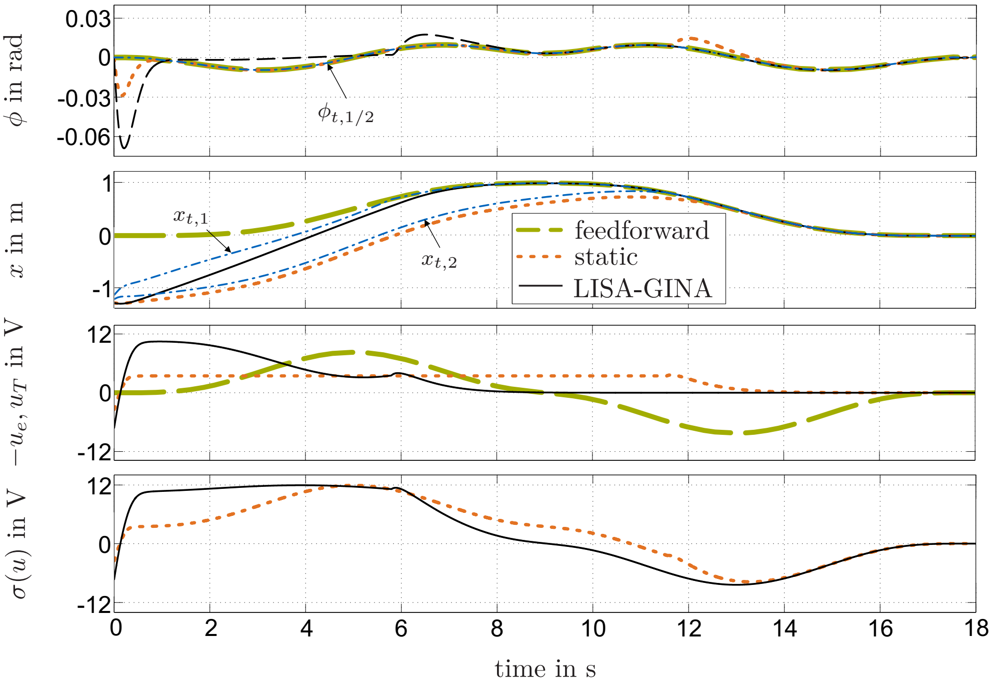


Figure 12.1: Comparison of LISA-GINA controller with a static input allocation.

amplitude depending on the current feedforward signal such that u_{max} is exploited during the whole error compensation. The continuous shifting of the temporary equilibrium $\mathbf{e}_t^*$ is visualized by the position and tilt angle coordinates

$$x_{t,s} = x_T - e^*_{t,x,s}, \quad \phi_{t,s} = \phi_T - e^*_{t,\phi,s} = \phi_T, \tag{12.4}$$

with $s \in \{1, 2\}$, in the state domain. They are marked in Fig. 12.1: $x_{t,1}$ and $\phi_{t,1}$ illustrate the shifting for the LISA-GINA control framework while $x_{t,2}$ and $\phi_{t,2}$ show the shifting concerning the static input allocation with GINA extension. Note, only $e^*_{t,x,s}$ differs from zero (cart position), whereby it becomes clear that the calculated $\mathbf{e}_t^*$ is a stabilizable equilibrium of the system. Consequently, the effectiveness of the LISA-GINA control framework becomes clear concerning performance.

Experiment – Validation of the LISA-GINA Control Framework: Due to the length of our test rig's rail, the cart's position is restricted to $x \in [-0.25, 0.45]$m. Thus, we reduce the maximum motor voltage from $u_{max} = 12\,\text{V}$ to $8\,\text{V}$ and choose a different desired trajectory than for simulation. Both changes are made in order to force input saturation to occur and to ensure the activation of the GINA controller during the experiment. In addition, the trajectory is shifted during the experiment whereby the robustness of the GINA controller can be seen once more.

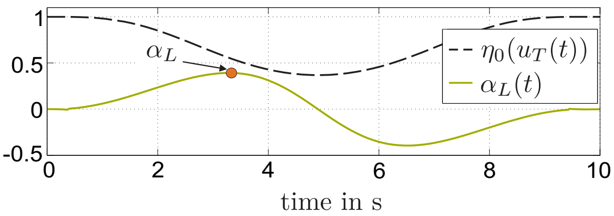


Figure 12.2: Variation of the decay rate and the bounding level value of the DA.

Fig. 12.2 shows the variation of the required decay rate of the Lyapunov function and the variation of the bounding level value $\eta_0(u_T(t))$ which equals to the changing of the DA according to 10.5 along the trajectory. The required decay rate $\alpha_L = \max(\alpha_L(t))$ is highlighted. The corresponding desired trajectory is depicted in Fig. 12.3 together with the obtained experimental results: We force a repeatable tracking error by shifting the position coordinate of the desired trajectory with Δx_T at $t = 2.85$s (like switching between two trajectories). According to Remark 11.2.1, we constructively design a controller which guarantees the existence of a quadratic Lyapunov function with the required decay rate. The so-called α-control law

$$\mathbf{u}_e = -\mathbf{b}^T\mathbf{P}_\alpha\mathbf{e} = \mathbf{f}^T\mathbf{e}, \tag{12.5}$$

is obtained by solving the algebraic Riccati equation

$$\left(\mathbf{A} + \frac{\alpha}{2}\mathbf{I}\right)^T \mathbf{P}_\alpha + \mathbf{P}_\alpha \left(\mathbf{A} + \frac{\alpha}{2}\mathbf{I}\right) - 2\mathbf{P}_\alpha\mathbf{b}\mathbf{B}^T\mathbf{P}_\alpha = \mathbf{0}, \tag{12.6}$$

defining a decay rate $\alpha > 0$. As proven in [20, 34] the controller (12.5) ensures the existence of a quadratic Lyapounov function with a decay rate of $\dot{V}_\mathbf{0} \leq -\alpha V_\mathbf{0}$. We choose $\alpha = 4 > \alpha_L$ (see Fig. 12.2).

As can be seen in Fig. 12.3, the α-controller exploits the possible overall system's input amplitude, resulting in a fast tracking behavior. For instance, at the time 5.43 s the input amplitude would be $u(5.43\,\mathrm{s}) = 11.6\,\mathrm{V}$ if saturation is ignored. However, the system stays asymptotically stable despite this noticeable over-saturation.

To sum up: The LISA-GINA control framework ensures a fast compensation of the tracking error (LISA condition) while guaranteeing the system's asymptotic stability (GINA controller). The overall input amplitude is exploited and noticeable over-saturation has been handled even in experiment which confirms the performance of the LISA-GINA controller and its practical applicability.

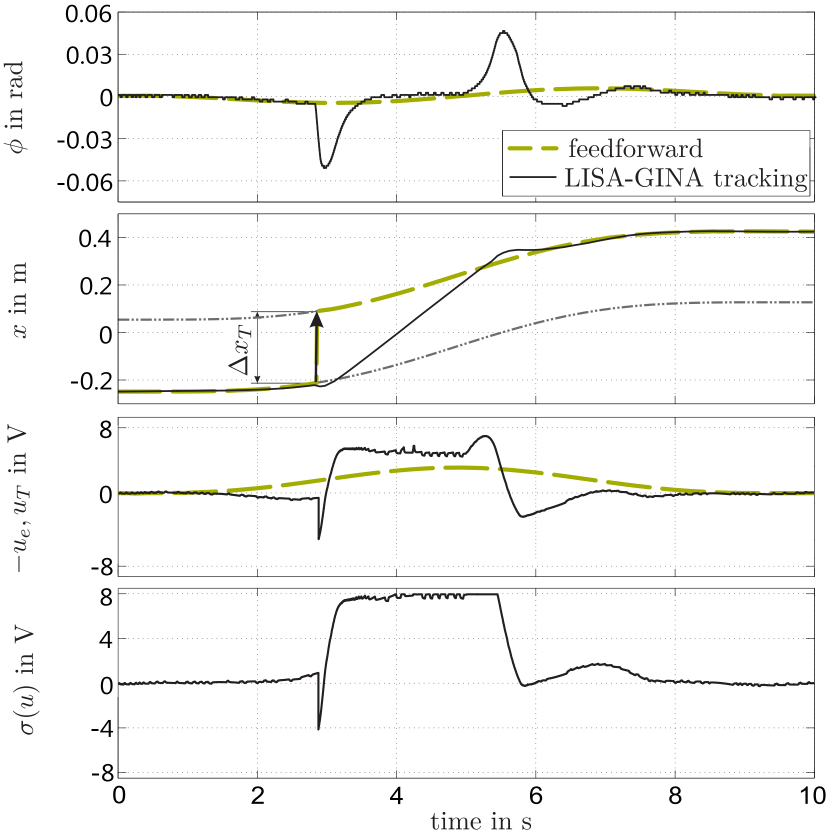


Figure 12.3: Experimental results: LISA-GINA Control Framework with α-controller.

12.2 Ballbot

In this section, the flatness-based trajectory design along defined waypoints according to Chapter 8 will be applied to the Ballbot system (C.4), (C.5). The Ballbot fulfills the Assumptions 8.1.1 and 8.1.2 whereby the piecewise trajectory generation procedure of Section 8.2 can be directly applied. Thereby, input and state limitations are considered. Finally, it is shown that a replanning of trajectories is feasible based on a comparison with a conventional optimization algorithm.

The state space model of the Ballbot can be decoupled into three single input systems

$$\dot{\mathbf{x}}_k = \mathbf{A}_k \mathbf{x}_k + \mathbf{b}_k \tilde{u}_k, \quad k \in \mathbb{N}_{1:3} \tag{12.7}$$

with the state vectors

$$\mathbf{x}_1 = \begin{bmatrix} x & \beta & \dot{x} & \dot{\beta} \end{bmatrix}^T, \ \mathbf{x}_2 = \begin{bmatrix} y & \alpha & \dot{y} & \dot{\alpha} \end{bmatrix}^T, \ \mathbf{x}_3 = \begin{bmatrix} \gamma & \dot{\psi} & \dot{\gamma} \end{bmatrix}^T. \tag{12.8}$$

The system matrices $\mathbf{A}_k$ are obtained by selecting the corresponding lines and rows from (C.4). The original input matrix of the Ballbot system is linked to the new (fictive) scalar inputs $\tilde{u}_k$ by

$$\tilde{\mathbf{u}} = \underbrace{\begin{bmatrix} 0 & 1 & -1 \\ 2 & -1 & -1 \\ 1 & 1 & 1 \end{bmatrix}}_{\mathbf{T}_{\tilde{u}}} \mathbf{u}, \tag{12.9}$$

whereby the decoupled input vectors $\mathbf{b}_k$ are given accordingly. The constructive computation of a flat output for systems of the form (12.7) is explained in Section 2.5 and detailed for the Ballbot system at hand in [33]. An easy to interpret flat output is

$$\begin{aligned} y_{f,1} &= x + 0.35\beta, && (12.10a) \\ y_{f,2} &= y - 0.35\alpha, && (12.10b) \\ y_{f,3} &= \gamma - 0.05\dot{\gamma} - 0.01\dot{\psi}. && (12.10c) \end{aligned}$$

Its first two components are structurally identical to the flat output of an inverted pendulum on cart system [33] and describe thus the planar movement of a point which is 0.35 m above the ball's center of gravity within the aluminum frame. The third component (12.10c) depends only on state variables γ, $\dot{\gamma}$ and $\dot{\psi}$. Hence, it represents the yawing of the robot.

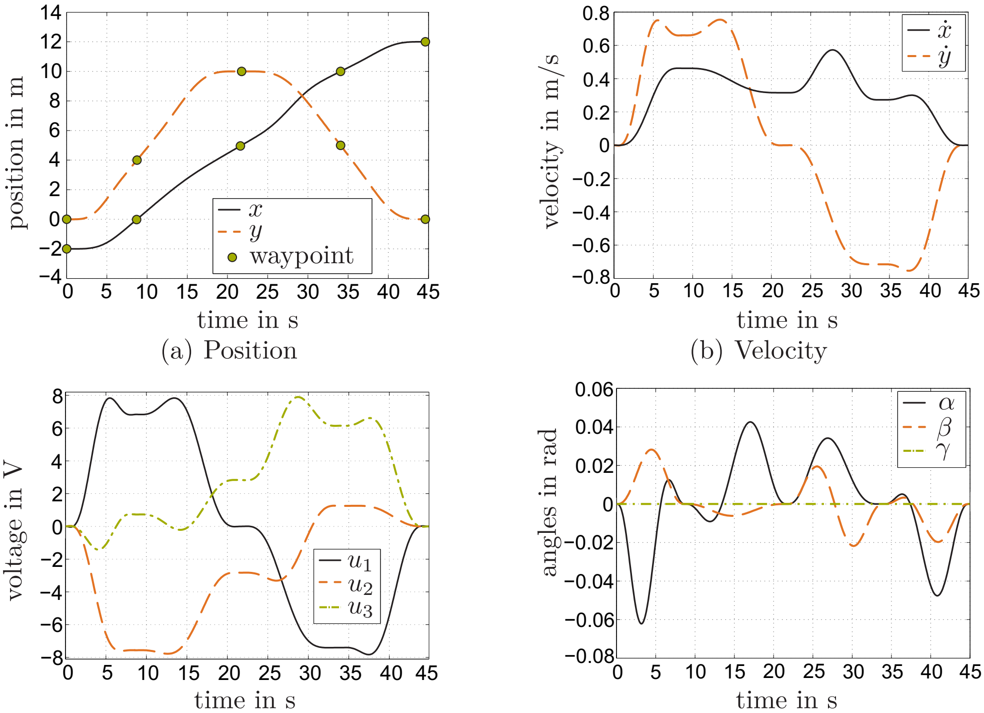
In the following we will generate a flatness-based trajectory according to Algorithm 8.2.1 along the waypoints (given by the coordinates x, y and γ)

$$\begin{aligned} &\boldsymbol{\Gamma}_0 = \begin{bmatrix} -2 & 0 & 0 \end{bmatrix}^T \text{m}, \quad \boldsymbol{\Gamma}_1 = \begin{bmatrix} 0 & 4 & 0 \end{bmatrix}^T \text{m}, \quad \boldsymbol{\Gamma}_2 = \begin{bmatrix} 5 & 10 & 0 \end{bmatrix}^T \text{m}, \\ &\boldsymbol{\Gamma}_3 = \begin{bmatrix} 10 & 5 & 0 \end{bmatrix}^T \text{m}, \quad \boldsymbol{\Gamma}_4 = \begin{bmatrix} 12 & 0 & 0 \end{bmatrix}^T \text{m}. \end{aligned} \tag{12.11}$$

Due to the advantage that yawing is solely defined by (12.10c), we are able to force $\gamma_T \overset{!}{=} 0$ for the whole trajectory by fixing $\tilde{u}_3 = 0$. The transition time in the bisection approach (8.8) is initially set to $t_{e,i}(0) = 0$, $t_{e,i}(1) = 20\,\text{s}$ for all $i \in \mathbb{N}_{1:4}$. We consider the following state and input constraints

$$\begin{aligned} \hat{c}_1 &:= |\alpha_T| \leq 10\,^\circ, \quad \hat{c}_2 := |\beta_T| \leq 10\,^\circ && (12.12a) \\ \hat{c}_{3,k} &:= |u_{k,T}| \leq 8\,\text{V}, \qquad k \in \mathbb{N}_{1:3}. && (12.12b) \end{aligned}$$

Therein, equation (12.12a) limits the tilt angles of the trajectory and (12.12b) restricts the allowed input amplitude to $\frac{2}{3}\mathbf{u}_{max}$.



(a) Position

(b) Velocity

(c) Input signal

(d) Tilt and yaw angles

Figure 12.4: Trajectory along five waypoints according to Algorithm 8.2.1.

Fig. 12.4 depicts the trajectory obtained by Algorithm 8.2.1. All waypoints are precisely hit (Fig. 12.4(a)) and the allowed input amplitude is exploited while the state limits are adhered to. Table 12.1 summarizes the final obtained overall transition time as well as the required computation time. The results are compared to an optimization by using the pattern search algorithm of the MATLAB *Global Optimization Toolbox* [87]. The pattern search algorithm has been initialized with the trajectory according to our developed strategy and the variation of the velocity at the waypoints $i \in \mathbb{N}_{1:3}$ has been additionally bounded by

$$\begin{aligned} \hat{\mathbf{c}}_4 &:= \left|\dot{\mathbf{\Gamma}}_i\right| < \left|\dot{\mathbf{\Gamma}}_{i,ini}\right| + \dot{\mathbf{\Gamma}}_{i,max}, \\ \hat{\mathbf{c}}_5 &:= \left|\dot{\mathbf{\Gamma}}_i\right| > \left|\dot{\mathbf{\Gamma}}_{i,ini}\right| - \dot{\mathbf{\Gamma}}_{i,max} \end{aligned} \tag{12.13}$$

where $\dot{\mathbf{\Gamma}}_{i,max} = [0.5, 0.5, 0]^T \mathrm{m/s}$ is the allowed variation region of the velocity at the waypoints and $\dot{\mathbf{\Gamma}}_{i,ini}$ denotes the velocity obtained by Algorithm 8.2.1. Comparing

Table 12.1: Trajectory design using an Intel Core $i5 - 3230\,\mathrm{M}$, 8 GBRAM, Win7 x64

trajectory design	transition time in s	computation time in s
Algorithm 8.2.1	44.93	0.76
Pattern search algorithm	43.25	279.92

both results, it can be seen that the transition time of the optimized trajectory is 1.68 s shorter than the result obtained according to our approach. However, the required computation time is 368.35 times larger. Algorithm 8.2.1 requires only 0.76 s whereby a real-time computation of a new trajectory is possible. For instance, when the robot moves in-between two waypoints a new trajectory can be generated for the subsequent trajectory pieces. Consequently, the benefit of the developed trajectory generation procedure becomes clear based on Table 12.1 and Fig. 12.4.

12.3 VTOL Aircraft

In this section, we apply the LISA-GINA control framework for tracking a trajectory with a nonlinear *vertical take-off and landing* (VTOL) aircraft subject to input amplitude and rate saturation. The time-variant error system of the aircraft is written in T-S notation according to Section 9. Simulation results highlight then advantages of the control framework: first, a large tracking error is fast and asymptotically compensated (GINA controller) and second, constraints are exploited (LISA condition).

The VTOL model

$$\ddot{x} = -u_1 \sin(\theta) + \varepsilon u_2 \cos(\theta) \tag{12.14a}$$

$$\ddot{y} = u_1 \cos(\theta) + \varepsilon u_2 \sin(\theta) - g \tag{12.14b}$$

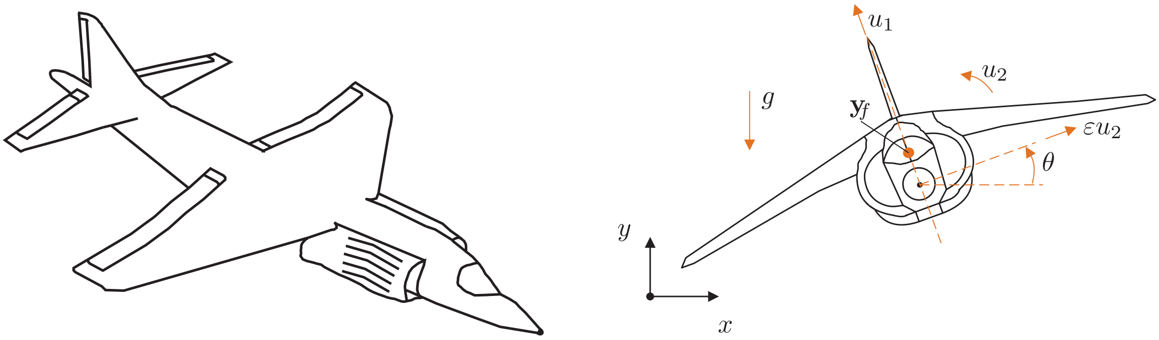
$$\ddot{\theta} = u_2 \tag{12.14c}$$

has been derived in [51]. Fig. 12.5 depicts the corresponding schematic of the aircraft. The VTOL aircraft describes a jet-borne operation (e.g. hovering) in a vertical-lateral plane. Its state variables are the position coordinates x and y, its roll angle θ and the corresponding velocities ($\dot{x}$, $\dot{y}$ and $\dot{\theta}$). The parameter g denotes the gravitational acceleration. The control inputs u_1 and u_2 are the thrust and the rolling moment, respectively. The small coefficient ε defines the coupling between the rolling moment and the lateral acceleration.

The center of thrust with the two components

$$\begin{aligned} y_{f,1} &= x - \varepsilon \sin(\theta) \\ y_{f,2} &= y + \varepsilon \cos(\theta). \end{aligned} \tag{12.15}$$

is a known flat output $\mathbf{y}_f$ of the system [8, 86]. Typical for an aircraft, we consider the inputs to be restricted in their amplitude and rate. To this end, we add the actuator



(a) Geometric view

(b) Planar systems view

Figure 12.5: Vertical take off and landing aircraft.

model (3.10) to the VTOL system. All parameter values of the finally considered VTOL aircraft are summarized in Table 12.2.

We generate a desired trajectory (2.56) of polynomial degree $q_k = 12$, $k \in \mathbb{N}_{1:2}$, for the flat output using Algorithm 8.2.1. The trajectory should precisely hit the waypoints

$$\mathbf{\Gamma}_0 = \begin{bmatrix}100 & 100\end{bmatrix}^T, \quad \mathbf{\Gamma}_1 = \begin{bmatrix}200 & 100\end{bmatrix}^T, \tag{12.16}$$

(e.g. a laterally evasive maneuver) with $\mathbf{\Gamma}_i = [x, y]^T$, $i \in \mathbb{N}_{0:1}$, (the remaining state variables of the VTOL aircraft are zero), subject to the input amplitude constraints

$$\hat{c}_1 := |v_{1,T}| \leq 15\,\mathrm{N/kg}, \quad \hat{c}_2 := |v_{2,T}| \leq 15\,1/\mathrm{s}. \tag{12.17}$$

of the actuator input signal $\mathbf{v}$ (see the actuator model (3.10)). The input rate constraints are set to

$$\hat{c}_3 := |\dot{v}_{1,T}| \leq 30\,\mathrm{N/kg}, \quad \hat{c}_4 := |\dot{v}_{2,T}| \leq 30\,1/\mathrm{s}. \tag{12.18}$$

Table 12.2: Parameters of the VTOL aircraft

description	symbol	value	unit
coupling parameter	ε	0.5	m
motors' inverse time constant	τ	10	1/s
thrust: input amplitude range	$u_{1,max}$	25	N/kg
thrust: input rate	$\dot{u}_{1,max}$	45	N/kgs
roll: input amplitude range	$u_{2,max}$	25	1/s
roll: input rate	$\dot{u}_{2,max}$	45	1/s^2
gravitation constant	g	9.81	N/kg

Fig. 12.6 depicts the final obtained input and state trajectory. The waypoints are precisely hit and the input constraints (12.17) are exploited but not violated. The required rate of the actuator input signal stays within a range of $|\dot{v}_{1,T}| \leq 5\,\mathrm{N/kgs}$, $|\dot{v}_{2,T}| \leq 5\,1/\mathrm{s}^2$ and thus restriction (12.18) is far from being violated. Analyzing the desired trajectory according to step 2 of the LISA-GINA control framework 11.2.1 results in a required decay rate of $\alpha_L = 0.33$. In order to design a related tracking controller (step 3 and 4 of the LISA-GINA control framework), we first derive the time-variant linear error dynamics by linearizing (12.14) along the desired trajectory. In a second step, we follow Theorem 9.2.1 to formulate a LO T-S formulation of the tracking error dynamics with $r = 11$ equidistant subsystems along the trajectory. Finally, we obtain an augmented error dynamics (according to 3.11)

$$\dot{\hat{\mathbf{e}}} = \sum_{i=1}^{r=11} h_i(\mathbf{z}_s)\left(\hat{\mathbf{A}}_i\hat{\mathbf{e}} + \hat{\mathbf{B}}\,\boldsymbol{\sigma}\left(\mathbf{K}\mathbf{e} + \mathbf{T}\boldsymbol{\sigma}(\mathbf{v})\right)\right) \tag{12.19}$$

with the augmented error vector $\hat{\mathbf{e}} = [\mathbf{e}, \mathbf{u}_e]^T \in \mathbb{R}^8$ whereby

$$\mathbf{e} = \begin{bmatrix} x_T - x & \dot{x}_T - \dot{x} & y_T - y & \dot{y}_T - \dot{y} & \theta_T - \theta & \dot{\theta}_T - \dot{\theta} \end{bmatrix}^T. \tag{12.20}$$

We estimate an over-saturating DA according to (3.24) and Remark 3.2.2 subject to the input constraints (amplitude and rate) within Table 12.2. In addition, we force a decay rate of $\alpha = 2\alpha_L$ and we search for a linear control law which is finally obtained

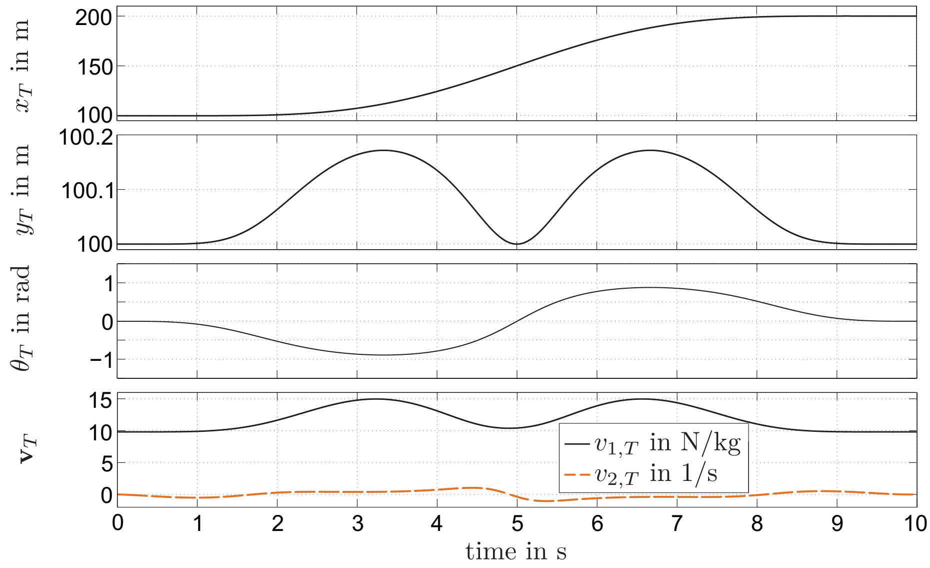


Figure 12.6: Desired state and input trajectory of the VTOL aircraft.

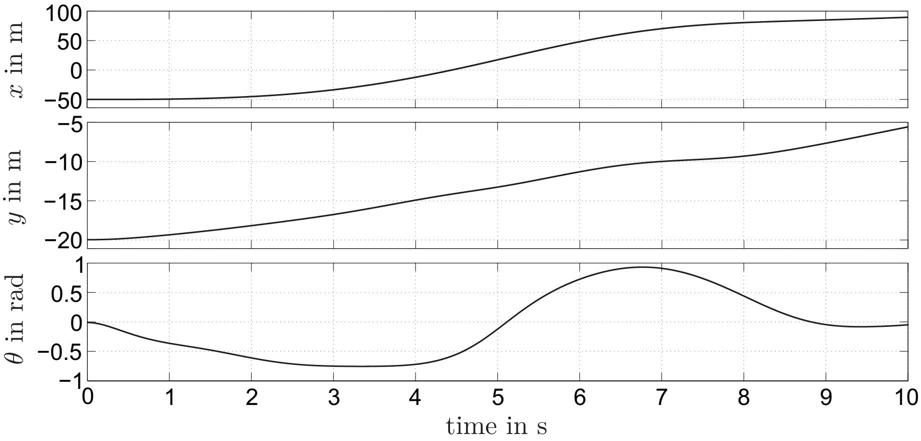


Figure 12.7: State trajectory of the VTOL aircraft.

to

$$F = \begin{bmatrix} 0.74 & 1.72 & 10.76 & 28.27 & -26.42 & -5.81 & 16.31 & -0.36 \\ -14.66 & -48.63 & -3.76 & -12.19 & 569.09 & 237.60 & -0.356 & 42.83 \end{bmatrix}. \tag{12.21}$$

Fig. 12.7 depicts the simulation results. Due to a quite large initial tracking error of

$$\hat{\mathbf{e}} = \begin{bmatrix} -50 & 0 & -20 & 0 & \dots & 0 \end{bmatrix}^T \tag{12.22}$$

the state trajectory does not match with the desired one. However, the LISA-GINA control framework prevents destabilization and ensures a fast compensation of the tracking error which is initially not within the DA of the desired trajectory. Without the reference governor the simulation became unstable. The necessity of the GINA controller is clarified by Fig. 12.8 which illustrates the asymptotic stabilization of the desired trajectory by the continuous shifting of the scaling parameter $c \in [0, 1]$ of the GINA controller. The value $c = 1$ is reached approximately at $t = 9.6\,\text{s}$. Here, the GINA controller is deactivated as the tracking error is within the DA of the desired trajectory. At this point the tracking task become trivial. Fig. 12.9 depicts the saturated input signals. While both actuator input signals v_1 and v_2 saturates, the rate saturation of the actuator prevents saturation of $\mathbf{u}$. Table 12.3 summarizes all relevant absolute values

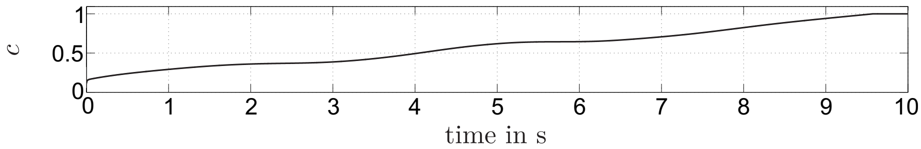


Figure 12.8: Scaling parameter c of the GINA controller towards the desired trajectory.

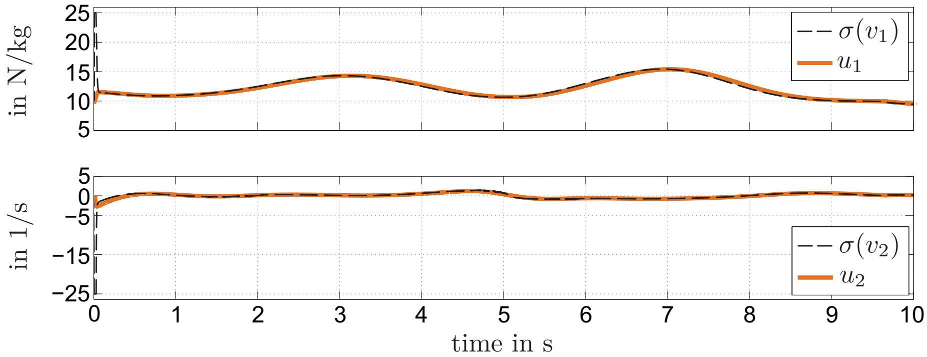


Figure 12.9: Input trajectory of the VTOL aircraft.

concerning input amplitude and rate saturation of the tracking controller: It can be seen that the maximum non-saturated input amplitude and rate values are larger than the corresponding bounds of the VTOL aircraft according to Table 12.2. Thereby, the over-saturation of the estimated DA and the exploitation of the input signal becomes obvious. Especially, the input rate is significantly over-saturating.

For the sake of completeness, we show that a smaller, but still worth mentioning, initial error can be totally compensated within the trajectory transition time. To this end, we set the initial error to

$$\hat{\mathbf{e}} = \begin{bmatrix} -5 & 0 & -0.5 & 0 & \dots & 0 \end{bmatrix}^T . \tag{12.23}$$

The final obtained trajectory is compared to the desired one in Fig. 12.10. The initial error is asymptotically compensated by exploiting the input range, e.g. v_2 saturates at the beginning of the tracking.

Table 12.3: Control input values for error compensation in the VTOL simulation

description	maximum absolute values
Amplitude	$[30, 93]^T$
Saturated amplitude	$[25, 25]^T$
Rate	$[152, 250]^T$
Saturated rate	$[45, 45]^T$

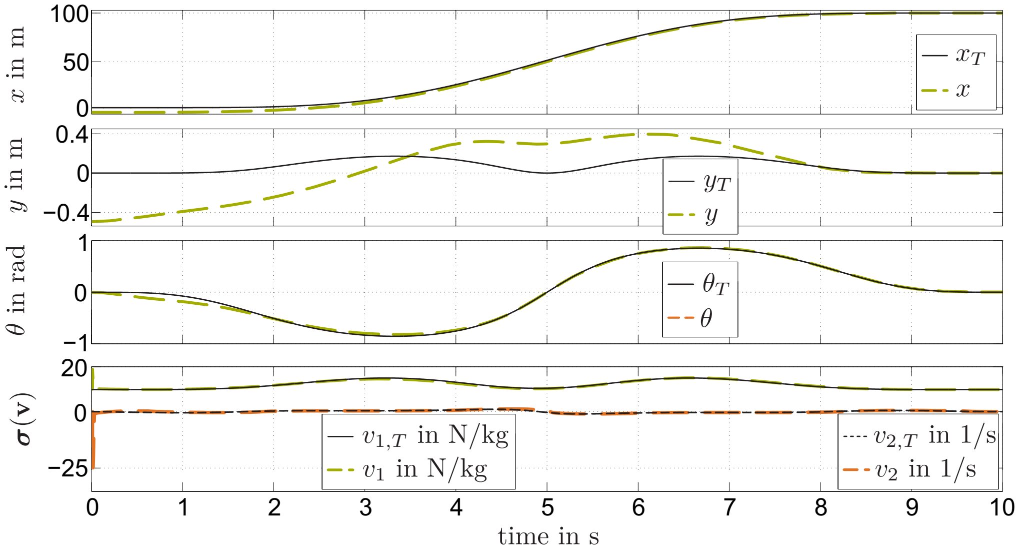


Figure 12.10: Trajectory tracking of the VTOL for smaller initial errors.

12.4 Summary

The practical applicability and the performance benefit gained by the results of the Chapters 8 to 11 within this thesis have been demonstrated. Two benchmark system have been recalled: the inverted pendulum on a cart and the Ballbot system which have been already investigated in the Sections 7.1 and 7.2, respectively. In addition, we have investigated a nonlinear vertical take-off and landing (VTOL) aircraft system.

We have applied the LISA-GINA control framework (see Section 11.2) to the inverted pendulum system subject to saturation in simulation and experiment. A comparison with a conventional tracking controller has shown that an enlarged domain of attraction has been realized and the tracking error has been compensated within half of the time. In addition, noticeable over-saturation has been handled even in the experiment. The Ballbot system has been considered to illustrate the performance of the trajectory generation procedure of Section 8. Thereby, input and state limitations have been considered.The required computational cost has been rather low compared to a conventional optimization algorithm (pattern search) such that a real-time replanning of trajectories is feasible. Concerning the VTOL aircraft system, we have tracked a trajectory subject to input amplitude and rate saturation. The error system has been written in T-S notation according to Chapter 9. The simulation results have highlighted the major advantages of the developed LISA-GINA control framework for T-S systems: large tracking errors can be fast and asymptotically compensated (GINA controller) while simultaneously exploiting the available input amplitude and rate (LISA condition). Each example has become unstable if the GINA controller has been deactivated.

Chapter 13

Conclusions

Today's control engineers are often facing two categories of problems — set point and trajectory tracking of dynamical systems. Both objectives have to be tackled subject to constraints, e.g. actuator and state limits, performance and safety requirements or computational effort. This thesis is primarily devoted to the development of new control methods for tackling the mentioned problems. To this end, we exploit the *Takagi-Sugeno* (T-S) modeling which allows to represent a quite general class of dynamical systems by a convex combination of a finite amount of linear or affine submodels. Consequently, observer and controller design as well as related optimization tasks can be done based on linear control theory.

In the first part of the thesis we have introduced new approaches concerning set point tracking while the second part is devoted to trajectory tracking. We have demonstrated the benefit of the methods as well as their practical relevance in a wide range of applicability in simulation and experiment. The examples origin from the field of robotics, aircraft and ground vehicle control. The following key points have been addressed in this thesis:

- efficient estimation and enlargement of the domain of attraction (DA),
- fast stabilization of a desired set point, even if the current state vector is not within the nominal estimated DA,
- computing performant trajectories along predefined waypoints,
- fast stabilization of a desired trajectory, even if the current tracking error is not within the nominal estimated DA and
- demonstrate the benefit and wide range of practical applicability of the methods.

Concerning the first key point, we have developed *set invariance conditions* in terms of linear matrix inequalities (LMIs) which allow the estimation of an ellipsoidal DA subject to *state, input amplitude and rate constraints.* The LMIs are formulated in a way such that the DA is allowed to include *over-saturating regions.* This means that an initial error within the DA might lead to an input signal which is above the saturation limit. However, the asymptotic stabilization of the considered equilibrium point is guaranteed despite saturation. By integrating the developed set invariance conditions in a novel numerical algorithm a *constructive computation of a large bounded sublevel set* of an ellipsoidal DA has been established. Thereby, the control law can be either predefined or considered as an optimization parameter subject to performance constraints.

The second key point has been addressed in two ways: First, we have extended the algorithm such that *nested invariant sets*, each having a maximized volume, can be estimated. Each DA is related to an *individual designed controller* whereby larger DAs with lower and smaller DAs with larger control performance have been obtained. *Switching and smooth switching conditions* allow to benefit from all individually designed controllers. The conditions have been derived based on multi Lyapunov function theory such that an asymptotically stable scheduling between the control laws is obtained. Consequently, the *trade-off between control performance and the size of the estimated DA has been relaxed.* Finally, the switching conditions have been extended to the *general case of not-nested DAs.* In the second route, we have investigated the case if *no DA has been estimated within which the current state vector is located in.* To handle such situations, the so-called *GINA controller* (Governor Integrated Nominal-Value Adaptation) has been devolved. Its basic operation principle is to compute a set point that is as close as possible to the actual desired one but can be stabilized subject to constraints. While this happens, the set point is shifted towards the desired one until it can be finally stabilized itself. *Several ways of implementation* have been shown for balancing the computational costs, performance and its general applicability.

With the third key point, computation of performant trajectories along predefined waypoints, we have started into the second part of the thesis where the focus is shifted from set point towards trajectory tracking. Here the focus have been to provide an *optimization-based approach for generating a flatness-based trajectory* such that a stationary movement (movement without acceleration) is obtained at each desired waypoint. To this end, the trajectory is subdivided into polynomial pieces in-between subsequent waypoints. The final transition time of the trajectory is minimized while the predefined waypoints are precisely hit and system relevant state and input constraints are not violated. Additionally, the approach is based on simple optimization methods, e.g. the bisection approach, which eases its implementation and its use while maintaining the *real-time applicability* for replanning a trajectory.

The forth key point is devoted to the *adaptation and the extension of our results* concerning set point tracking such that they can be beneficially applied *for compensating a trajectory tracking error.* With this in mind, the *tracking error dynamics of a system in T-S notation* have been formulated. Based on that, we have proved that all of the results obtained for set point tracking are applicable to trajectory tracking if each *system constraint is a priori subdivided into two parts* (input amplitude, input rate and state): one part forms the restriction for the trajectory design and the other part for compensating a tracking error. In order to improve the control performance, we have established a *condition for dynamically allocating the saturation limits* of the feedforward and the feedback part. More precisely, the *LISA condition* (Limits of Inputs and States are Allocated) leads to a required decay rate of the quadratic Lyapunov function which defines the DA. If this decay rate is ensured then the restrictions concerning error compensation can be dynamically adapted based on the corresponding current values of the trajectory. A *combination of the LISA condition and the GINA controller* has led to *a new trajectory tracking control framework that guarantees a fast stabilization of a desired trajectory even if the current tracking error is not within the estimated DA of the trajectory.*

The last key point has been addressed by simple *numerical examples* within several sections of this thesis in order to *clarify the devolved methods.* Further, *simulation and experimental results* have *demonstrated their benefits and their practical applicability* to a wide range of technical systems: the well-known nonlinear benchmark system *inverted pendulum on cart*, an *omnidirectionally movable unstable robot*, a *nonlinear vertical take-off and landing aircraft* and an *active cruise control system* for a car.

Based on the results in this thesis, the following expedient and interesting issues arise and should be addressed in future research:

Deriving a flat output In Appendix E, first results are derived for constructively determining a flat output for a nonlinear system with a single input based on its T-S notation. It would be very useful to extend this procedure to multiple inputs.

Estimating the DA for Nonlinear Controlled Systems Computing a DA for nonlinear controlled systems subject to constraints is not an easy task and numerically expensive. In [26], a new possibility is discussed for easing the problem in general. Therein, a closed-loop nonlinear system (consisting of a nonlinear plant and a nonlinear controller) is written in SE-NL T-S notation in order to compute a related DA based on LMIs. An automation of the approach would be of interest.

It is also worth to investigate the applicability of the results for estimating a valid DA (Chapter 3) for designing T-S observers. For instance, in [98] the benefit of a T-S observer for a nonlinear suspension system is shown. Thereby, the DA of the observer is estimated subject to the universe of discourse of the system.

Extension to Non-Quadratic Lyapunov Functions The methods in this thesis are based on quadratic Lyapunov functions. A naturally rising research question is how the results can be applied to non-quadratic Lyapunov functions. As explained in Section 1.1 non-quadratic approaches (see [77, 95, 120]) reduce the conservatism of the estimate but their applicability is limited. Problems like computational complexity, integrating system constraints and handling T-S formulation with affine terms have to be addressed.

Combining the GINA controller with MPC Model Predictive Control (MPC) seems to be a promising extension of our GINA controller approach for further enlarging its operation region. To this end, two possible combinations would be of interest: First, if no stabilizable equilibrium can be found by the GINA controller then MPC can be used to drive the system towards the nearest set point. As soon as the state vector is within the DA of this equilibrium, the GINA controller is activated again. Thereby, the dual problem of computational cost and stability in MPC (see Section 1.1) can be relaxed as the stable terminal region is placed as close as possible to the current state vector. Hence, a short horizon length is achievable.

While in the first proposed combination MPC acts more or less as a fail-safe mode, the GINA controller takes over that role in the second approach: The primary goal is to compute an equilibrium based on the GINA controller that is as close as possible towards the desired one but can be stabilized by the MPC controller within a single horizon. In other words, the GINA controller determines the stable terminal region for the MPC and shifts that region towards the tracking target.

Fault Tolerant GINA Controller A future-oriented topic is fault detection and fault-tolerant control [90, 123]. Thereby, a system continues to operate in case of a fault, e.g. of a sensor or an actuator, by locating the error and adapting the control law accordingly. The GINA controller can be beneficially integrated in such a control architecture. The advantage might be to change parameters within the GINA controller instead of changing the controller. For instance, by adapting the desired set point or the saturation limits. Thereby, issues like performance guarantees and stability proofs can become simpler. This should be deeper investigated in future research.

Appendix A

Technical Proofs

A.1 Proof of Theorem 6.3.1

The convergence of the temporary equilibrium towards $\mathbf{e}_d = \mathbf{x}_t^* - \mathbf{x}_d^*$ as well as the control error $\mathbf{e}_t = \mathbf{x} - \mathbf{x}_t^*$ must be considered. To this end, we consider the Lyapunov-like function

$$V_{\mathbf{e}_t,\mathbf{e}_d} = \frac{\eta_{\mathbf{0}}}{\eta_{\mathbf{u}_t^*}} V_{\mathbf{x}_t^*} + \underbrace{\mathbf{e}_d^T \mathbf{e}_d}_{V_{\mathbf{e}_d}} \tag{A.1}$$

with $V_{\mathbf{x}_t^*}$ being the Lyapunov function (6.6) at $\mathbf{x}_t^*$ and $\eta_{\mathbf{u}_t^*} > 0$ is the related bounding level value (e.g. (6.9)). The rear term $V_{\mathbf{e}_d}$ in (A.1) is zero if $\mathbf{x}_t^* = \mathbf{x}_d^*$ whereby we have to differ between two cases:

Case A.1.1 ($\mathbf{x}_t^* = \mathbf{x}_d^*$)**.** *The GINA controller is inactive and the derivative of* (A.1) *is*

$$\dot{V}_{\mathbf{e}_t,\mathbf{e}_d} = \frac{\eta_{\mathbf{0}}}{\eta_{\mathbf{u}_t^*}} \dot{V}_{\mathbf{x}_d^*} < 0 \tag{A.2}$$

as $\dot{V}_{\mathbf{x}_d^*} < 0$ *due to the considered asymptotically stabilizing control law for* $x \in \mathcal{X}_{\mathbf{x}_d^*}(\mathbf{P}, \eta_{\mathbf{u}_d^*})$*.*

Case A.1.2 ($\mathbf{x}_t^* \neq \mathbf{x}_d^*$)**.** *According to Algorithm 6.3.1, the scaling variable c is updated such that* $\mathbf{x}$ *is at the border of the DA of the actual* $\mathbf{x}_t^*$*, meaning* $\mathbf{x} \in \partial\mathcal{X}_{\mathbf{x}_t^*}$ *(see also Fig. 6.3). Thereby, the first term in* (A.1) *remains constant*

$$\frac{\eta_{\mathbf{0}}}{\eta_{\mathbf{u}_t^*}} V_{\mathbf{x}_t^*} = \eta_{\mathbf{0}}. \tag{A.3}$$

The derivative of (A.1) *becomes*

$$\dot{V}_{\mathbf{e}_t,\mathbf{e}_d} = \dot{V}_{\mathbf{e}_d} = 2\left(\mathbf{x}_t^* - \mathbf{x}_d^*\right)^T \dot{\mathbf{x}}_t^*, \tag{A.4}$$

which can be reformulated to

$$\dot{V}_{\mathbf{e}_t,\mathbf{e}_d} = \dot{V}_{\mathbf{e}_d} = 2\dot{c}\left(\hat{\mathbf{f}}(\mathbf{x}_r^*, \mathbf{x}_d^*, c) - \mathbf{x}_d^*\right)^T \frac{\partial\left(\hat{\mathbf{f}}(\mathbf{x}_r^*, \mathbf{x}_d^*, c)\right)}{\partial c} \tag{A.5}$$

replacing $\mathbf{x}_t^*$ *and* $\dot{\mathbf{x}}_t^*$ *with* (6.5a) *and its derivative, respectively. Due to* (6.11), $\dot{V}_{\mathbf{e}_t,\mathbf{e}_d} < 0$ *within the relevant range* $c \in [0, 1[$ *if* $\dot{c} > 0$ *which is proven in the following: Suppose a solution* $c(t_1)$ *at a time* t_1 *is fixed, meaning that* $\mathbf{x}_t^*$ *is kept constant and thus* $\dot{c} = 0$ *whereby* $\dot{V}_{\mathbf{e}_t,\mathbf{e}_d} = 0$ *according to* (A.5). *Then the Lyapunov-like function* (A.1) *will decrease based on* (A.2) *(setting* $\mathbf{x}_d^* = \mathbf{x}_t^*$*). However, equation* (A.3) *will be violated an infinitesimal time instance* $t_2 = t_1 + \delta t$*, with* $\delta t << 1$*, later as*

$$\frac{\eta_{\mathbf{0}}}{\eta_{\mathbf{u}_t^*}} V_{\mathbf{x}_t^*}(t_2) < \frac{\eta_{\mathbf{0}}}{\eta_{\mathbf{u}_t^*}} V_{\mathbf{x}_t^*}(t_1) = \eta_{\mathbf{0}}. \tag{A.6}$$

Consequently, c *can be shifted along each direction of* $\hat{\mathbf{f}}$ *at the time* t_2 *until* (A.3) *is fulfilled again. Based on Algorithm 6.3.1* $c(t_2) > c(t_1)$ *is selected, whereby* $\mathbf{x}_t^*$ *gets closer toward* $\mathbf{x}_d^*$*. Comparing* (A.1) *for* $c(t_1)$ *and* $c(t_2)$ *results in*

$$\frac{\eta_{\mathbf{0}}}{\eta_{\mathbf{u}_t^*}} V_{\mathbf{x}_t^*}(t_2) = \frac{\eta_{\mathbf{0}}}{\eta_{\mathbf{u}_t^*}} V_{\mathbf{x}_t^*}(t_1) = \eta_{\mathbf{0}}, \quad V_{\mathbf{e}_d}(t_2) < V_{\mathbf{e}_d}(t_1) \tag{A.7}$$

and thus $V_{\mathbf{e}_t,\mathbf{e}_d}(t_2) < V_{\mathbf{e}_t,\mathbf{e}_d}(t_1)$*. In view of that and remembering the continuity (*$\mathcal{C}^0$*) of* c *(Remark 6.3.1), the integral*

$$\int_{t_1}^{t_2} = \max_i (\dot{c})\, dt = c(t_2) - c(t_1) > 0 \tag{A.8}$$

exists whereby $\dot{c} > 0$ *is ensured and* $\dot{V}_{\mathbf{e}_t,\mathbf{e}_d} < 0$ *proven for the considered case.*

According to the Cases A.1.1 and A.1.2, the asymptotic stability of the GINA controller extended close-loop system is ensured which concludes the proof.

A.2 Proof of Theorem 6.4.2

Based on the functions (6.14) we reformulate (6.7), (6.9) for every input i to

$$\begin{aligned} V_{\mathbf{x}_t^*} = {} & c_i^2 \left[(\mathbf{x}_r^* - \mathbf{x}_d^*)^T \mathbf{P}\,(\mathbf{x}_r^* - \mathbf{x}_d^*)\right] + c_i \left[2\,(\mathbf{x}_r^* - \mathbf{x}_d^*)^T\right] \cdot \left[\mathbf{P}\,(\mathbf{x} - \mathbf{x}_r^*)\right] + \\ & + (\mathbf{x} - \mathbf{x}_r^*)^T \mathbf{P}\,(\mathbf{x} - \mathbf{x}_r^*) \end{aligned} \tag{A.9}$$

for the left-hand side and

$$\eta_{u^*_{t,i}} = \frac{\eta_{\mathbf{0}}}{u^2_{max,i}} \cdot \begin{cases} \Big[u_{max,i} - u^*_{r,i} + c_i \cdot (u^*_{r,i} - u^*_{d,i})\Big]^2 & \text{if } u^*_{t,i} \geq 0 \\ \Big[u_{max,i} + u^*_{r,i} + c_i \cdot (u^*_{d,i} - u^*_{r,i})\Big]^2 & \text{if } u^*_{t,i} < 0 \end{cases} \tag{A.10}$$

for the right-hand side of the equation. Regarding to the trivial reference equilibrium $\mathbf{x}^*_r = \mathbf{0}$ the quadratic functions (A.10) can be further written in vertex form of a parabola

$$\eta_{u^*_{t,i}} = a^*_{t,i} \left(c_i - s_{t,i}\right)^2, \quad i \in \{1, 2, ...\} \tag{A.11}$$

with $a^*_{t,i} = \eta_{\mathbf{0}} \frac{{u^*_{d,i}}^2}{u^2_{max,i}}$ and

$$s_{t,i} = \begin{cases} \frac{u_{max,i}}{u^*_{d,i}} & \text{if } u^*_{t,i} \geq 0, \\ -\frac{u_{max,i}}{u^*_{d,i}} & \text{if } u^*_{t,i} < 0, \end{cases} \tag{A.12}$$

equals to the c-coordinate of the vertex. That leads to a coupling of two arbitrary chosen inputs i and l by

$$a^*_{t,i} = a^*_{t,j} \left(\frac{s_{t,j}}{s_{t,i}}\right)^2 \quad i, j \in \{1, 2, \ldots, m\}. \tag{A.13}$$

Based on that the intersection points of the level sets, and thus the change of the minimal level value, can be determined depending on c as

$$\begin{aligned} 0 &= a^*_{t,j} \frac{s_{t,j}}{s_{t,i}} \left(c - s_{t,i}\right)^2 - a^*_{t,j} \left(c - s_{t,j}\right)^2, \\ 0 &= c \Big[c \cdot \left(s^2_{t,j} - s^2_{t,i}\right) - 2 \left(s_{t,i} \cdot s^2_{t,j} - s_{t,j} \cdot s^2_{t,i}\right)\Big], \\ \Rightarrow c_1 &= 0, \quad c_2 = 2\frac{s_{t,i} \cdot s_{t,j}}{s_{t,i} + s_{t,j}}. \end{aligned} \tag{A.14}$$

The first intersection point $c_1 = 0$ confirms that $\partial\mathcal{X}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}})$ is a common level set. Assuming, without loss of generality, $s_{t,i} \leq s_{t,j}$ leads to

$$c_2 \geq s_{t,i}, \quad \text{as } s_{t,j} \geq \frac{s_{t,i} + s_{t,j}}{2} \tag{A.15}$$

for the second intersection point. In other words, c_2 is never on the left-hand side of the smallest vertex $\tilde{s}_t = \min_i \left(s_{t,i}\right)$. Consequently, the input that belongs to $\tilde{s}_t$ defines the smallest level set for all $c \in [0, 1]$ whereby the proof is concluded.

The quintessence of that is sketched in Fig. A.1 considering a system with three inputs u_i, $i \in \mathbb{N}_{1:3}$. Shown is the in c quadratic function (A.9), (A.11) and one of its possible

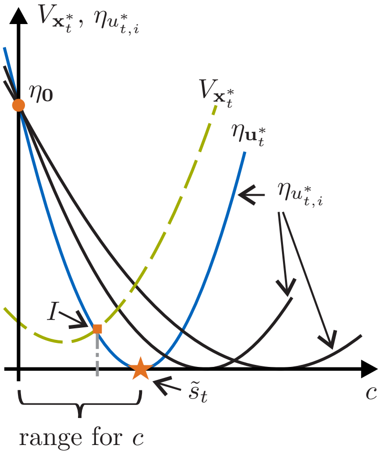


Figure A.1: Sketched curves for determining c if $\mathbf{u}_r^* \neq \mathbf{u}_t^* \neq \mathbf{u}_d^*$.

solutions I: Both functions, $V_{\mathbf{x}_t^*}$ as well as $\eta_{u_t^*,i}$ are not allowed to become negative. Thus $V_{\mathbf{x}_t^*}$ intersects with the level set corresponding to $\tilde{s}_t$ first (marked with a star). The operating range of $c \in [0, 1]$ has to end before $\tilde{s}_t$ is reached as here $|\mathbf{u}_t^*| = \mathbf{u}_{max}$ and thus the constraint in (6.14b) is violated. Consequently, $\eta_{\mathbf{u}_t^*}$ equals to the level value $\eta_{u_{t,i}^*}$ that belongs to a fixed and non-changing input i (in fact the one belonging to $\tilde{s}_t$) whereby the proof is concluded.

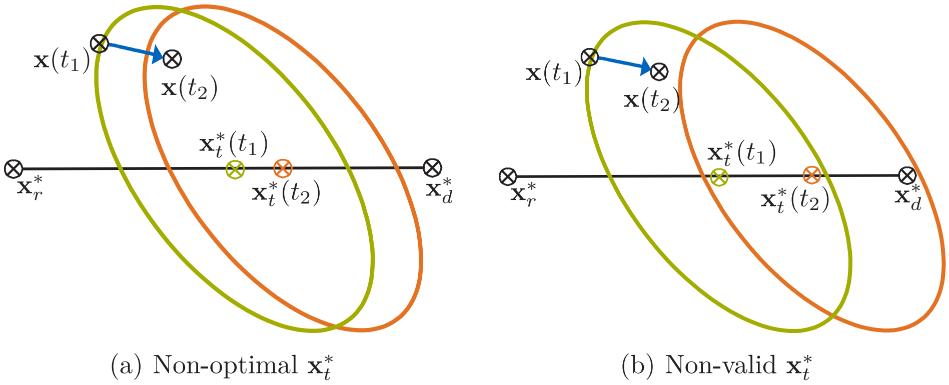
A.3 Proof of Theorem 6.4.3

Due to the fact that a solution can be obtained which is non-valid or at least not the optimal one, which is given by Theorem 6.3.1, we have to distinguish between four possible cases:

Case A.3.1 ($\mathbf{x}_t^* = \mathbf{x}_d^*$)**.** *The GINA controller is inactive and the asymptotic stability of $\mathbf{x}_d^*$ is ensured by case A.1.1 of Theorem 6.3.1 (see Section A.1).*

Case A.3.2 (optimal $\mathbf{x}_t^*$)**.** *If the temporary equilibrium of the iterative implementation equals to the optimal solution according to Theorem 6.3.1 then the asymptotic stabilization of $\mathbf{x}_d^*$ is guaranteed by case A.1.2 of Theorem 6.3.1 (see Section A.1).*

Case A.3.3 (non-optimal $\mathbf{x}_t^*$)**.** *Fig. A.2(a) illustrates a non-optimal but valid updating for the temporary equilibrium from a time t_1 to $t_2 = t_1 + \delta t$, with $\delta t << 1$, meaning that $\mathbf{x}(t_2)$ is within the DA of $\mathbf{x}_t^*(t_2)$ but non on the corresponding bounding level value. Depicted is the reference equilibrium $\mathbf{x}_r^*$, the desired set point $\mathbf{x}_d^*$, two subsequently calculated and activated temporary equilibria $\mathbf{x}_t^*(t_1)$ and $\mathbf{x}_t^*(t_2)$ as well as the trajectory piece of the state vector in-between $t \in [t_1, t_2]$. The value of the Lyapunov-like function* (A.1)



(a) Non-optimal $\mathbf{x}_t^*$

(b) Non-valid $\mathbf{x}_t^*$

Figure A.2: Visualization of possible non-optimal solutions.

decrease within the considered time-span as V_{e_d} as well as $\frac{\eta_{\mathbf{0}}}{\eta_{\mathbf{u}_t^*}} V_{\mathbf{x}_t^*}$ become smaller. Consequently, as at each time instance an optimal solution which asymptotically stabilize $\mathbf{x}_d^*$ generally exists a non-optimal solution leads to a decrease of the Lyapunov-like function (compared to the corresponding optimal solution of the prior time step). Non-optimal solutions asymptotically stabilize $\mathbf{x}_d^*$ as well.

Case A.3.4 (non-valid $\mathbf{x}_t^*$)**.** *Fig. A.2(b) illustrates a non-valid updating of $\mathbf{x}_t^*$, meaning $\mathbf{x}(t_2)$ is not within the DA of $\mathbf{x}_t^*(t_2)$. According to Algorithm 6.4.1, $\mathbf{x}_t^*(t_1)$ is kept active whereby it becomes a valid non-optimal solution (see Case A.3.3).*

Due to the universal approximation theorem, as detailed in [81], each nonlinear function can be represented with required accuracy. At the case at hand that means: finding a new valid temporary equilibrium can be ensured by refining (increasing) the basic rule base of Table 6.1. Therefore, the number of membership functions and linguistic variables has to be enlarged. Consequently, the asymptotic stability of the GINA controller extended close-loop system is ensured due to the Cases A.3.1 to A.3.4 whereby the proof is concluded.

A.4 Proof of Lemma 10.2.1

The bounding level value of the DA (10.5) is given by transferring (6.9) into the error domain:

$$\eta_{\mathbf{0}}(\mathbf{u}_T(t)) = \eta_{\mathbf{0}}(\mathbf{u}_{T,\mathbf{p}}(t)) = \min_i \left(\underbrace{\eta_{\mathbf{0}} \cdot \frac{(\overbrace{u_{max,i} - |u_{T,i}(t)|}^{u_{e,max,i}(t)})^2}{u_{max,i}^2}}_{\eta_{\mathbf{0}}(u_{T,i}(t))} \right), \quad \forall\, t \in \mathcal{T}. \tag{A.16}$$

The nominal bounding level value $\eta_{\mathbf{0}}$ is estimated based on LMIs (analogous to the set point case) setting $\mathbf{u}_T(t) = \mathbf{0}$, meaning $\mathbf{u}_{e,max}(t) = \mathbf{u}_{max}$ based on (10.4). The index $i \in \{1, 2, \ldots, m\}$ denotes the elements of the input vector and the index vector $\mathbf{p}$ summarizes the index of the elements $u_{T,i}$ that lead to the smallest level value at a certain time, e.g. if $\eta_{\mathbf{0}}(\mathbf{u}_{T,\mathbf{p}}(t_1)) = \eta_{\mathbf{0}}(u_{T,2}(t_1)) = \eta_{\mathbf{0}}(u_{T,4}(t_1))$ then $\mathbf{p} = [2, 4]$.

The time derivative of (A.16) is

$$\dot{\eta}_{\mathbf{0}}(u_{T,i}(t)) = \frac{2\eta_{\mathbf{0}}\dot{u}_{T,i}(t)(|u_{T,i}(t)| - u_{max,i})}{u_{max,i}^2}\,\mathrm{sgn}(u_{T,i}(t)) \tag{A.17}$$

whereby the signum function occurs due to the weak differentiability of the absolute value function. Due to Assumption 10.2.1, the derivative $\dot{u}_{T,i}(t)$ exists. The smallest level value $\eta_{\mathbf{0}}(\mathbf{u}_T(t))$ that decreases most (or increases least) of all of them is

$$\dot{\eta}_{\mathbf{0}}(\mathbf{u}_T(t)) = \min_p \left(\dot{\eta}_{\mathbf{0}}(\mathbf{u}_{T,p}(t))\right). \tag{A.18}$$

Suppose that a tracking error $\mathbf{e}(t_1) \in \mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\mathbf{u}_T(t_1)))$ occurs at a time $t_1 \in \mathcal{T}$. Now, for guaranteeing that $\mathbf{e}(t)$ stays within $\mathcal{E}_{\mathbf{0}}(\mathbf{P}, \eta_{\mathbf{0}}(\mathbf{u}_T(t)))$ for each $t > t_1$, $t \in \mathcal{T}$, the tracking error hast to decrease fast enough. More precisely, the value of the Lyapunov function $V_{\mathbf{0}}$ has to decrease faster than the critical level value of the related DA (10.5). Claiming that based on a decay rate $\alpha_{\mathbf{u}} > 0$ of $V_{\mathbf{0}}$ results in

$$\dot{V}_{\mathbf{0}} \leq -\alpha_{\mathbf{u}} V_{\mathbf{0}} \leq \dot{\eta}_{\mathbf{0}}(\mathbf{u}_T(t)) \tag{A.19a}$$

$$\leq -\alpha_{\mathbf{u}}\eta_{\mathbf{0}}(\mathbf{u}_T(t)) \leq \dot{\eta}_{\mathbf{0}}(\mathbf{u}_T(t)), \quad \forall\, t \in \mathcal{T} \tag{A.19b}$$

whereby $V_{\mathbf{0}} = \eta_{\mathbf{0}}(\mathbf{u}_T(t))$ denotes the most critical case (the error is on critical level value). Reformulating (A.19b) leads to

$$\alpha_{\mathbf{u}} \geq \underbrace{\frac{-\dot{\eta}_{\mathbf{0}}(\mathbf{u}_T(t))}{\eta_{\mathbf{0}}(\mathbf{u}_T(t))}}_{\alpha_{\mathbf{u}}(t)}, \quad \forall\, t \in \mathcal{T}, \tag{A.20a}$$

$$\alpha_{\mathbf{u}} \geq \max\left(\beta, \max_{t \in \mathcal{T}}\left(\alpha_{\mathbf{u}}(t)\right)\right), \tag{A.20b}$$

where $\beta \ll 1$ denotes a small positive scalar which ensures that $\dot{V}_{\mathbf{0}} < 0$ at least, e.g. for the non-critical case that $\alpha_{\mathbf{u}}(t) < 0$. Consequently, the proof is completed.

Appendix B

Discrete Reference Equilibrium

Beside the analytical calculation of a valid reference equilibrium, it can be also derived in a numerical way. The task of selecting a valid reference equilibrium (line 3 and 18 of Algorithm 6.5.1) depends on the available data storage and the computational power. In the following, we summarize two approaches based on a discretized functional relation of allowed system's equilibria which is either directly given by (6.3) or by discretizing (6.1.1). In Section B.1, the state space within the DA of each possible reference equilibrium is discretized and in Section B.2 the DA is approximated with a polytope whereby the required data storage can be further reduced.

B.1 Tabularization

In the first approach, we discretize the state space within the DA of each possible reference equilibrium $\mathbf{x}_r^*(k)$ along the functional relation of allowed system's equilibria $\hat{\mathbf{f}}(\mathbf{x}(k))$. This procedure is illustrated in Fig. B.1 concerning two reference equilibria $\mathbf{x}_r^*(k)$ with $k \in \mathbb{N}_{1:2}$. The state space within the bounding level value $\eta_r^*(1)$ is discretized (black crosses in Fig. B.1) and stored in a look-up table (state belonging to $\mathbf{x}_r^*(1)$). Due

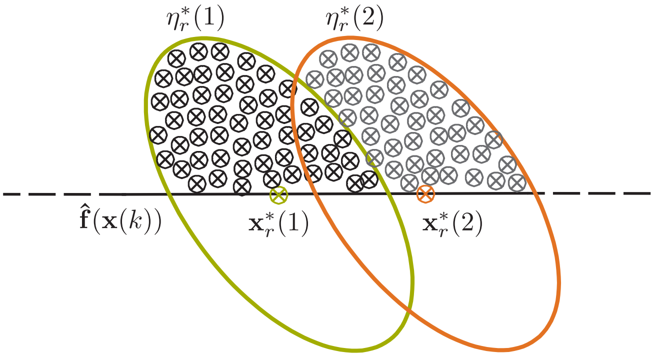


Figure B.1: Tabularization of the state space within $\eta_r^*(k)$.

to the point symmetry of the ellipsoidal DAs it is enough to discretize the upper half plane of the state space. Concerning the subsequent DA bounded by $\eta_r^*(2)$, we only have to store discrete points in the state space which are already within $\eta_r^*(1)$ (gray crosses in Fig. B.1).

Now, if the reference equilibrium needs to be recalculated in real-time then the current state vector is quantized with the discrete state values (stored in the look-up table).

B.2 Polytope within Ellipsoid

In order to reduce the required data storage of the tabularization approach we approximate the DA of each reference equilibrium by a polytope. This procedure is illustrated in Fig. B.2 based on two reference equilibria (analogous to Fig. B.1). Each DA is approximated by an inner polytope. In Fig. B.2, they exemplarily consist of five nodes. Note that the amount of nodes can be adjusted based on the size of the state space and the available data storage. Due to the point symmetry of the ellipsoidal DAs it is enough to consider their upper halves. The polytopes can be shaped in such a way that they do not intersect. In other words, the polytope concerning a reference equilibrium is constructed such that no intersection with the polytope of another reference equilibrium exist. For instance, the green polytope belongs to $\mathbf{x}_r^*(1)$ and the red one to $\mathbf{x}_r^*(2)$. Similar to explicit MPC [9, 126], a decision tree needs to be build up which finally detects within which polytope the actual state vector is. Consequently, the corresponding reference equilibrium has to be activated.

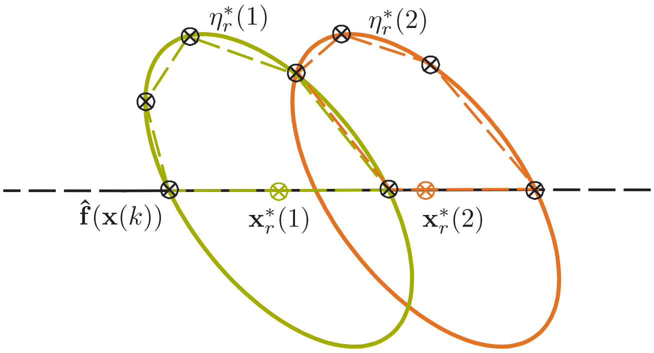


Figure B.2: Polytopic bounding of the domain of attraction.

Appendix C

Parameters of the Test Rigs

In this chapter, we summarize the parameters of the experimental systems. If a parameter range is given, e.g. input amplitude saturation, then the actual considered value is separately defined in each experiment.

C.1 Inverted Pendulum Test Rig

The parameters of the inverted pendulum system (7.1) are summed up in Tables C.1. The motor can be considered as an ideal rate limiter ($\tau \to \infty$) and its rate saturation is adjusted by a rate limiter in each experiment.

Table C.1: Parameters of the Pendulum Test Rig

description	symbol	value	unit
pendulum (rod) center of gravity	a	0.1925	m
rod mass	m_p	0.146	kg
cart mass	m_w	5.9	kg
total mass	$m_g = m_w + m_p$	6.046	kg
moment of inertia	$\Theta = \frac{4}{3}ma^2$	$1.8 \cdot 10^{-3}$	kgm^2
friction constant	d_x	843	Ns/m
motor constant	c_m	24.95	N/V
motor inverse time constant	τ	$\to \infty$	1/s
input amplitude range	u_{max}	$[0, 45]$	V
input rate	$\dot{u}_{max}$	user defined	V/s
gravitation constant	g	9.81	N/kg

The matrices (3.12) for deriving the SE-NL T-S model (3.11) are

$$\hat{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \theta_1\theta_2\theta_3 3m_p g & 0 & -\theta_1 4d_x & -\theta_1\theta_4 4m_p a & \theta_1 4c_m \\ \theta_1\theta_2\frac{3m_g g}{a} & 0 & -\theta_1\theta_3\frac{3d_x}{a} & -\theta_1\theta_3\theta_4 3m_p & \theta_1 f_3\frac{3}{a} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \hat{\mathbf{B}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{C.1}$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 & 0 & -\tau \end{bmatrix}.$$

C.2 Ballbot Test Rig

The parameters of the Ballbot test rig are summarized in Tables C.2. The parameter called "ext. frame" is the combination of the original frame F, the motors M_i, the clips C_i and the omniwheels W_i with $i \in \mathbb{N}_{1:3}$.

In [92], where the equations of motion of the Ballbot are detailed, the torque moment of the DC motors operating on the ball are expressed by

$$M_i = -\frac{k_E k_M i_G^2 \eta_M \eta_G}{R_M} \cdot \dot{\delta}_i + \frac{k_M i_G \eta_M \eta_G}{R_M} \cdot u_i, \quad i \in \mathbb{N}_{1:3}, \tag{C.2}$$

whereby the actual motor parameters are given in Table C.3. However, an experimental validation of the equation (C.2) has shown that an additional nonlinear damping function $d_{rot}(u_i)$ depending on the rotation speed of the omniwheels is required. The

Table C.2: Parameters of the Bodies of the Ballbot Test Rig

description	symbol	value	unit
ext. frame mass	m_F	7.839	kg
ext. frame moment of inertia: x-direction	$\Theta_x^{(F)}$	0.335	kgm^2
ext. frame moment of inertia: y-direction	$\Theta_y^{(F)}$	0.335	kgm^2
ext. frame moment of inertia: z-direction	$\Theta_z^{(F)}$	0.085	kgm^2
ext. frame center	l	0.300	m
ball mass	m_B	2.319	kg
ball moment of inertia	$\Theta_{x,y,z}^{(B)}$	0.024	kgm^2
ball radius	r_B	0.125	m
ball spin friction	d	0.170	Nms
omniwheel work angle	η	$\frac{1}{4}\pi$	rad

Table C.3: Parameters of the Motors and the Gears of the Ballbot Test Rig

description	symbol	value	unit
torque constant	k_M	19.1	$\frac{\text{mNm}}{\text{A}}$
voltage constant	k_E	2.0	$\frac{\text{mV}}{\text{rpm}}$
efficiency factor	η_M	0.83	–
terminal resistance	R_M	0.41	Ω
amplitude range	u_{max}	$[0, 12]$	V
gear ratio	i_G	43	–
gear efficiency factor	η_G	0.70	–
rotational damping in PCU	$d_{rot,PCU}$	0.8	Nms
rotational damping in VCU	$d_{rot,VCU}$	0.1	Nms

final nonlinear motor equation is

$$M_i = -\frac{k_E k_M i_G^2 \eta_M \eta_G}{R_M} \cdot \dot{\delta}_i + \frac{k_M i_G \eta_M \eta_G}{R_M} \cdot u_i - d_{rot}(u_i)\dot{\delta}_i, \quad i \in \mathbb{N}_{1:3}. \tag{C.3}$$

Fig. C.1 shows the identified damping function. It can be seen that the rotational damping coefficient highly varies with the motors' voltage. Consequently, we consider individual rotational damping coefficients for position (PCU) and velocity (VCU) control. The corresponding values ($d_{rot,PCU}$ and $d_{rot,VCU}$) are given in Table C.3.

The modeling of the Ballbot is detailed in [92]. This together with the updated Ballbot parameters (Table C.2 and Table C.3) and the new motor equation (C.3)) lead

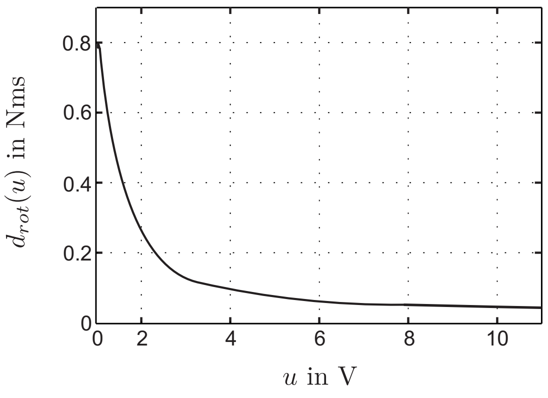


Figure C.1: Identified rotational damping coefficient $d_{rot}(u)$.

to the final state space model

$$
{}_p\mathbf{A} = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & -7.5217 & -82.0290 & 0 & 0 & 17.8823 \\
0 & 0 & 7.5217 & 0 & 0 & -82.0290 & -17.8823 & 0 \\
0 & 0 & 38.4979 & 0 & 0 & -271.7613 & -59.2440 & 0 \\
0 & 0 & 0 & 38.4979 & 271.7613 & 0 & 0 & -59.2440
\end{bmatrix}, \tag{C.4}
$$

$$
{}_p\mathbf{B} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & -2.7525 & 2.7525 \\
3.1784 & -1.5892 & -1.5892 \\
10.5299 & -5.2649 & -5.2649 \\
0 & 9.1192 & -9.1192
\end{bmatrix} \tag{C.5}
$$

concerning the PCU and for the VCU the matrices are:

$$
{}_v\mathbf{A} = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & -7.5217 & -49.3257 & 0 & 0 & 10.7530 \\
7.5217 & 0 & 0 & -49.3257 & -10.7530 & 0 \\
38.4979 & 0 & 0 & -163.4155 & -35.6246 & 0 \\
0 & 38.4979 & 163.4155 & 0 & 0 & -35.6246
\end{bmatrix}, \tag{C.6}
$$

$$
{}_v\mathbf{B} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & -2.7525 & 2.7525 \\
3.1784 & -1.5892 & -1.5892 \\
10.5299 & -5.2649 & -5.2649 \\
0 & 9.1192 & -9.1192
\end{bmatrix}. \tag{C.7}
$$

Appendix D

Recurrent Fuzzy switching rule base for the Inverted Pendulum

Table D.1 summarizes the rule base of the recurrent fuzzy switching controller considered in the experimental example 7.1.2. The linguistical characteristics (7.7) are abbreviated by "S", "M" and "L", respectively. The notation of the Table is as follows: The currently active subsystem $\Sigma^i(k)$ is written in the first column and the linguistical characteristics of the input $\tilde{\mathbf{u}}(k) = [\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k)]^T$ is placed in the first row. Thus, the first row and column represent the rules' premises. The rest of the Table represents the rules' conclusions, meaning which subsystem $\Sigma^i(k+1)$ is activated at the next time step. For instance, the second rule of the first row is given by

$$\text{If } \tilde{x}(k) \text{ is } \Sigma^1 \text{ and } [\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k)]^T \text{ is } [S, S, M]^T \text{ then } \tilde{x}(k+1) \text{ is } \Sigma^2. \tag{D.1}$$

It can be seen that some rules' conclusions have an additional upper index "$*$" which denotes that the mulit Lyapunov switching (condition *ii*) in Theorem 4.3.1) needs to be fulfilled before activating the corresponding closed-loop system. In other words, if the corresponding system should be activated due to the rule but the multi Lyapunov switching condition is not fulfilled yet then the actual system is kept active until the condition is satisfied. If the switching condition is fulfilled then a smooth blending happens such that only a single change of the active Lyapunov function occurs.

Table D.1: Rule base of the recurrent fuzzy switching pendulum controller

Σ^i	$[S,S,S]^T$	$[S,S,M]^T$	$[S,S,L]^T$	$[S,L,S]^T$	$[S,L,M]^T$	$[S,L,L]^T$	$[L,S,S]^T$	$[L,S,M]^T$	$[L,S,L]^T$	$[L,L,S]^T$	$[L,L,M]^T$	$[L,L,L]^T$
Σ^1	$\check{\Sigma}^{1,*}$	Σ^2	Σ^2	Σ^1	Σ^2	Σ^2	Σ^1	Σ^2	Σ^2	Σ^1	Σ^2	Σ^2
Σ^2	$\Sigma^{1,*}$	Σ^2	$\Sigma^{3,*}$	$\Sigma^{1,*}$	Σ^2	$\Sigma^{3,*}$	$\Sigma^{1,*}$	Σ^2	$\Sigma^{3,*}$	$\Sigma^{1,*}$	Σ^2	$\Sigma^{3,*}$
Σ^3	Σ^2	Σ^2	Σ^3	Σ^2	Σ^2	Σ^3	Σ^2	Σ^2	Σ^3	Σ^2	Σ^2	Σ^3
$\check{\Sigma}^1$	$\check{\Sigma}^4$	$\Sigma^{1,*}$	$\Sigma^{1,*}$	$\check{\Sigma}^1$	$\Sigma^{1,*}$	$\Sigma^{1,*}$	$\check{\Sigma}^{1,*}$	$\Sigma^{1,*}$	$\Sigma^{1,*}$	$\check{\Sigma}^1$	$\Sigma^{1,*}$	$\Sigma^{1,*}$
$\check{\Sigma}^4$	$\check{\Sigma}^4$	$\check{\Sigma}^1$	$\check{\Sigma}^1$	$\check{\Sigma}^4$	$\check{\Sigma}^1$	$\check{\Sigma}^1$	$\check{\Sigma}^1$	$\check{\Sigma}^1$	$\check{\Sigma}^1$	$\check{\Sigma}^1$	$\check{\Sigma}^1$	$\check{\Sigma}^1$

Appendix E

Flat output based on a T-S Formulation

Due to the difficulty of systematically finding a flat output of a nonlinear system in general, one may think about exploiting the T-S framework such that the problem can be eased to the linear case. For instance, this thought has been recently addressed for continuous-time recurrent fuzzy systems (see Section 2.1.1) in [42]. However, interpolating the flat outputs of the T-S subsystems does not necessarily yield to a flat output of the nonlinear system. Hence, the following problem statement rises naturally:

Problem E.0.1. Given a system in SN-NL T-S notation (2.6)(or a LO T-S model (2.15) without affine terms). Then the problem is to systematically find a flat output of the original nonlinear system based on the linear subsystems of the T-S model.

In the following we briefly summarize first results concerning a nonlinear system with a single input which hopefully will help to solve this problem in further research. We state our results in terms of two theorems which can be automatically checked. Consequently, in our opinion it is worth to be tried to find a flat output of a nonlinear system:

Theorem E.0.1. *Let a nonlinear system with a single input be given in T-S notation which consists of r subsystems without affine terms. Then the flat output of the original system equals to the flat outputs* $\mathbf{y}_f = \mathbf{y}_{f,i}$, $i \in \mathbb{N}_{1:r}$ *of the T-S subsystems if all subsystems are controllable and a common and constant transformation matrix*

$$\mathbf{\Psi} = \mathbf{\Psi}_i, \quad i \in \mathbb{N}_{1:r}, \tag{E.1}$$

into the flat coordinates (2.62) *exist.*

Proof: If all linear subsystems of the T-S representation are controllable then each of systems is differentially flat. If in addition, the transformation matrices into the flat coordinates fulfill (E.1) then the flat output subspace of each subsystem is identical. Hence, the flat output is not affected by the convex interpolation $\sum_{i=1}^{r} h_i(\mathbf{z}_s)$ of the T-S model which concludes the proof. ■

If however, the T-S subsystems does not share a common and constant transformation matrix (E.1) then it depends on system nonlinearities $\theta_k(\mathbf{z}_s)$, $k \in \mathbb{N}_{1:\frac{r}{2}}$. Hence, the convex interpolation is affected by

$$\boldsymbol{\Psi}(\boldsymbol{\theta}(\mathbf{z}_s)) = \sum_{i=1}^{r} h_i(\mathbf{z}_s)\boldsymbol{\Psi}_i. \tag{E.2}$$

Concerning that case, we state the following Theorem:

Theorem E.0.2. *Let a nonlinear system with a single input be given in T-S notation which consists of r subsystems without affine terms. If no common and constant transformation matrix* (E.1) *for all subsystems exist then* (E.2) *is a valid transformation into flat coordinates if* $\operatorname{rank}(\boldsymbol{\Psi}_i) = n$ *is fulfilled for each* $\boldsymbol{\Psi}_i$ *and if a nonlinearity* $\theta_k(\mathbf{z}_s)$, $k \in \mathbb{N}_{1:\frac{r}{2}}$*, that is located in the* $j \leq n$ *is* $(n-j)+1$ *times differentiable before the input* u *occurs.*

Proof: Due to the nonlinear transformation matrix the controllability of each linear subsystem of the T-S representation (as in Theorem E.0.1) is not enough to ensure flatness of the original nonlinear system. The postulated rank condition is required for the whole operating region (UoD) of the T-S system in order to guarantee the existence of an unique transformation into flat coordinates in general. For proving that a valid transformation has been detected, the location of nonlinearities $\theta_k(\mathbf{z}_s)$, $k \in \mathbb{N}_{1:\frac{r}{2}}$ within $\boldsymbol{\Psi}(\boldsymbol{\theta}(\mathbf{z}_s))$ has to checked: A nonlinearity that is located in the $j \leq n$ row the transformation matrix is according to (2.63) either part of the flat output y_f or of its derivatives. Hence, the nonlinearity occurs in the flat coordinates. Consequently, $\theta_k(\mathbf{z}_s)$ has to be $(n-j)+1$ times differentiable in order to obtain the highest derivative $y_f^{(n)}$ which is required for describing the input signal u depending on the flat output (see (2.65)) and the proof is concluded. ■

In order to clarify the established Theorems, we recall Example 2.2.3:

Example E.0.1 (Example 2.2.3 cont'd)**.** The systems controllability matrix is given by

$$\mathbf{Q}_s = \begin{bmatrix} 0 & 1 \\ 1 & f_1 \end{bmatrix}. \tag{E.3}$$

From the inverse of (E.3) if follows that a flat output of the system is $y_f = x_1$ and thus a common transformation matrix

$$\mathbf{\Psi} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{E.4}$$

is obtained for both subsystems as stated in Theorem E.0.1.

Example E.0.2. If the nonlinearity in Example E.0.1 is shifted such that the new system is

$$\dot{\mathbf{x}} = \begin{bmatrix} f_1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \sigma(u) \tag{E.5}$$

then a non-constant transformation matrix consisting of the subsystems

$$\mathbf{\Psi}_1 = \begin{bmatrix} 1 & 0 \\ \underline{f}_1 & 1 \end{bmatrix}, \mathbf{\Psi}_2 = \begin{bmatrix} 1 & 0 \\ \overline{f}_1 & 1 \end{bmatrix}, \tag{E.6}$$

is obtained. However, both subsystems share the same flat output $y_{f,i} = x_1$, $i \in \mathbb{N}_{1:2}$ which is equivalent to the flat output of the original nonlinear system due to Theorem E.0.2.

Bibliography

[1] Bossa nova robotics: http://www.bnrobotics.com, October 2013.

[2] Rezero Ballbot: http://www.rezero.ethz.ch, April 2014.

[3] Segway: http://www.segway.com, April 2014.

[4] Yalmip wiki: http://users.isy.liu.se/johanl/yalmip/, January 2015.

[5] J. Adamy and A. Flemming. Soft variable-structure controls: a survey. *Automatica*, 40:1821–1844, 2004.

[6] J. Adamy and R. Kempf. Regularity and chaos in recurrent fuzzy systems. *Fuzzy Sets Syst.*, 140(2):259–284, 2003.

[7] J. Adamy and R. Kempf. A survey on fuzzy systems with inherent dynamics. *at-Automatisierungstechnik*, 52(10):456–469, 2004.

[8] A. Ailon. Control for autonomous vtol aircraft with restricted inputs. In *Proc. IEEE Mediterranean Conference on Control and Automation*, pages 1569–1574, 2009.

[9] A. Alessio and A. Bemporad. A survey on explicit model predictive control. In L. Magni, D. M. Raimondo, and F. Allgöwer, editors, *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 345–369. Springer, 2009.

[10] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright. Nonlinear predictive control and moving horizon estimation - an introductory overview. In P. M. Frank, editor, *Advances in Control, Highlights of ECC'99*, pages 391–449. Springer, 1999.

[11] F. Allgöwer, R. Findeisen, and Z. K. Nagy. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs.*, 35(3):299–315, 2004.

[12] B. D. O. Anderson and J. B. Moor. Linear system optimization with prescribed degree of stability. *Proc. IEEE*, 116(12):2083–2087, 1969.

[13] A. Bateman and Z. Lin. An analysis and design method for linear systems under nested saturation. *Systems and Control Letters*, 48:41–52, 2003.

[14] O. Begovich, E. N. Sanchez, and M. Maldonado. Takagi-Sugeno fuzzy scheme for real-time trajectory tracking of an underactuated robot. *IEEE Trans. Contr. Syst. Technol.*, 10(1):14–20, 2002.

[15] S. Bezzaoucha, B. Marx, D. Maquin, and J. Ragot. Model reference tracking control for nonlinear systems described by Takagi-Sugeno structure. In *Proc. IEEE Conf. on Fuzzy Systems*, pages 1–8, 2013.

[16] A. Boccia, L. Grüne, and K. Worthmann. Stability and feasibility of state constrained MPC without stabilizing terminal constraints. *Systems and Control Letters*, 72:14–21, 2014.

[17] F. Borrelli, P. Falcone, J. Pekar, and G. Stewart. Reference governor for constrained piecewise affine systems. *J. Process Contr.*, 19(8):1229–1237, 2009.

[18] S. Boyd, L. E. Ghaouli, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory.* SIAM, 1994.

[19] M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Automat. Contr.*, 43(4):475–482, 1998.

[20] M. Buhl and B. Lohmann. Control with exponentially decaying Lyapunov function and its use for systems with input saturation. In *Proc. European Control. Conf.*, pages 3148–3153, 2009.

[21] M. Buhl and B. Lohmann. Lyapunov function based set point generator. *at-Automatisierungstechnik*, 10:499–504, 2009.

[22] R. E. Burkard and U. T. Zimmermann. *Einführung in die Mathematische Optimierung.* Springer, 2013.

[23] Y.-Y. Cao and Z. Lin. Robust stability analysis and fuzzy-scheduling control for nonlinear systems subject to actuator saturation. *IEEE Trans. Fuzzy. Syst.*, 11(1):57–67, 2003.

[24] J. M. G. da Silva, S. Tarbouriech, and G. Garcia. Local stabilization of linear systems under amplitude and rate saturating actuators. *IEEE Trans. Automat. Contr.*, 48(5):842–847, 2004.

[25] J. M. Gomes da Silva, M. Z. Oliveira, D. Coutinho, and S. Tarbouriech. Static anti-windup design for a class of nonlinear systems. *Int. J. Robust Nonlinear Control*, 24:793–810, 2014.

[26] Sergio Delgado and Klaus J. Diepold. Abschätzung des Einzugsbereichs unter Stellgrößenbeschränkung für die passivitätsbasierte Regelung mittels Takagi-Sugeno. In Günter Roppenecker and Boris Lohmann, editors, *Methoden und Anwendungen der Regelungstechnik*, pages 30–39. Shaker Verlag, 2015.

[27] S. Delprat, P.F. Toulotte, and T.M. Guerra J. Boonaert. Robust pole placement in a LMI region for Takagi-Sugeno models: Application to vehicle spacing control. In *Proc. IFAC Triennial World Congress*, pages 390–395, 2005.

[28] K. J. Diepold, A. Albers, and T. Guggemos. Flatness-based trajectory optimization along waypoints for a linear Ballbot system. *at-Automatisierungstechnik*, 62(12):842–850, 2014.

[29] K. J. Diepold and K. Albert. Local stability analysis of Takagi-Sugeno systems subject to their universe of discourse. *at-Automatisierungstechnik*, 62(10):687–697, 2014.

[30] K. J. Diepold and B. Lohmann. Transient probabilistic recurrent fuzzy systems. In *Proc. IEEE Conf. on Systems Man and Cybernetics*, pages 3529–3536, 2010.

[31] K. J. Diepold and S. J. Pieczona. Ein Ansatz für ein Fuzzy-Führungsfilter zur Berücksichtigung von Stellgrößenbeschränkungen beim Arbeitspunktwechsel. In *CI Workshop, Dortmund*, pages 87–98, 2012.

[32] K. J. Diepold and S. J. Pieczona. Recurrent Takagi-Sugeno fuzzy interpolation for switched linear systems and hybrid automata. In *Proc. IEEE Conf. on Fuzzy Systems*, pages 1–8, 2012.

[33] K. J. Diepold and S. J. Pieczona. Abschätzung des Einzugsbereiches für T-S Systeme: Beachtung des Gültigkeitssektors. In *CI Workshop, Dortmund*, pages 2090–2096, 2013.

[34] K. J. Diepold and S. J. Pieczona. Tracking control with adaptively allocated maximum input amplitudes and enlarged domain of attraction for linear systems. In *Proc. IEEE Conference on Decision and Control*, pages 370–384, 2013.

[35] K. J. Diepold, J. Schmidt-Colinet, B. Lohmann, and B. Vogel-Heuser. Intelligent probabilistic recurrent fuzzy control of human-mashine systems. In *Proc. IFAC Word Congress*, pages 4857–4862, Milano, Italy, 2011.

[36] H. Du, N. Zhang, J. C. Ji, and W. Gao. Robust fuzzy control of an active magnetic bearing subject to voltage saturation. *IEEE Trans. Contr. Syst. Technol.*, 18(1):164–169, 2010.

[37] A. Ellouze, F. Delmotte, J. Lauber, M. Chtourou, and M. Ksantini. Nonquadratic Lyapunov function for continuous TS fuzzy models through their discretization. *Fuzzy Sets Syst.*, 253:64–81, 2014.

[38] J. A Farrell and M. M. Polycarpou. *Adaptive approximation based control: unifying neural, fuzzy and traditional adaptive approximation approaches.* John Wiley & Sons, 1st edition, 2006.

[39] G. Feng. *Analysis and synthesis of fuzzy control systems.* CRC Press, Taylor & Francis, 2010.

[40] E. Feron, V. Balakrishnan, S. Boyd, and L. El Ghaoui. Numerical methods for H_2 related problems. In *Proc. IEEE American Control Conference*, pages 2321–2922, 1992.

[41] Michel Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *Int. J. Control*, 61(6):1327–1361, 1995.

[42] S. Gering and J. Adamy. Feedforward tracking control of flat recurrent fuzzy systems. *In Proc. J. Phys.: Conf. Ser. (ACD 2014)*, 570:022001, 2014.

[43] S. Gering and J. Adamy. Fuzzy control of continuous-time recurrent fuzzy systems. *Fuzzy Sets Syst.*, pages 126–141, 2014.

[44] S. Gering and J. Adamy. Synthese von Zustands- und Ausgangsrückführungen für rekurrente Fuzzy-Systeme. *at-Automatisierungstechnik*, 62(10):708–719, 2014.

[45] H. Ghorbel, A. El Hajjaji, M. Souissi, and M. Chaabane. Fault-tolerant trajectory tracking control for Takagi-Sugeno systems with unmeasurable premise variables: Descriptor approach. *Circuits Syst. Signal Process*, 33:1763–1781, 2014.

[46] V. Gorrini and H. Bersini. Recurrent fuzzy systems. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 193–198, Orlando, USA, 1994.

[47] A. Grancharova and T. A. Johansen. *Explicit nonlinear model predictive control*, volume 429 of *Lecture Notes in Control and Information Sciences*. Springer, 2012.

[48] L. Grüne and J. Pannek. *Nonlinear model predictive control: theory and algorithms*. Springer, 2011.

[49] T. Gußnera, M. Jost, and J. Adamy. Controller design for a class of nonlinear systems with input saturation using convex optimization. *Systems and Control Letters*, 61:258–265, 2012.

[50] W. M. Haddad, V. Chellaboina, and S. G. Nersesov. *Impulsive and hybrid dynamical systems: stability, dissipativity, and control*. Princeton University Press, 2006.

[51] J. Hauser, S. Sastry, and G. Meyer. Nonlinear control design for slightly non-minimum phase systems: Application to v/stol aircraft. *Automatica*, 28:665–679, 1992.

[52] T.A. Henzinger. The theory of hybrid automata. In *Proc. IEEE Annual IEEE Symposium on Logic in Computer Science*, pages 278–292, USA, 1996.

[53] P. Hippe. *Windup in control: its effects and their prevention*. Springer, 2006.

[54] P. Hippe. Stable and unstable systems with amplitude and rate saturation. In S. Tarbouriech, G. Garcia, and A. H. Glattfelder, editors, *Advanced Strategies in Control Systems with Input and Output Constraints*, pages 31–60. Springer, 2007.

[55] R. Hollis. Ballbots. *Scientific American*, 295(4):72–77, 2006.

[56] I. M. Horowitz. Synthesis of feedback systems. *Academic Press*, 1963.

[57] T. Hu and Z. Lin. *Control systems with actuator saturation: analysis and design*. Birkhäuser, 2001.

[58] T. Hu, Z. Lin, and B. M. Chenb. An analysis and design method for linear systems subject to actuator saturation and disturbance. *Automatica*, 38:351–359, 2002.

[59] A. Jadbabaie, M. Jamshidi, and A. Titli. Guaranteed-cost design of continuous-time Takagi-Sugeno fuzzy controllers via linear matrix inequalities. In *Proc. IEEE Conf. on Fuzzy Systems*, pages 268–273, 1998.

[60] U. Kalabic, C. Vermillion, and I. Kolmanovsky. Reference governors design for computationally efficient attitude and tether tension constraint enforcement on a lighter than air wind energy system. In *Proc. IEEE American Control Conference*, pages 1004–1010, 2013.

[61] K. Kefferpütz. *Regelungen für Systeme unter Stellgrößen- und Stellratenbeschränkungen*. PhD thesis, TU Darmstadt, Germany, 2013.

[62] K. Kefferpütz, C. Ackermann, and J. Adamy. Two-degree-of-freedom control of linear systems subject to input amplitude and rate constraints. *at-Automatisierungstechnik*, 3(60):155–167, 2012.

[63] R. Kempf. *Rekurrente Fuzzy-Systeme*. PhD thesis, TU Darmstadt, Germany, 2004.

[64] R. Kempf and J. Adamy. Sequential pattern recognition employing recurrent fuzzy systems. *Fuzzy Sets and Systems*, 146:451–472, 2004.

[65] H. Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle River, 3rd edition, 2002.

[66] E. Kim and H. Lee. New approaches to relaxed quadratic stability condition of fuzzy control systems. *IEEE Trans. Fuzzy. Syst.*, 8(5):523–534, 2000.

[67] Euntai Kim, Chang-Hoon Lee, and Young-Wan Cho. Analysis and design of an affine fuzzy system via bilinear matrix inequality. *IEEE Trans. Fuzzy Syst.*, 13(1):115–123, Feb 2005.

[68] G. Koch, K. J. Diepold, and B. Lohmann. Multi-objective road adaptive control of an active suspension system. In H. Ulbrich and L. Ginzinger, editors, *Motion and Vibration Control*, pages 189–200. Springer, 2009.

[69] I. Kolmanovsky, E. Garone, and S. Di Cairano. Reference and command governors: A tutorial on their theory and automotive applications. In *Proc. European Control. Conf.*, pages 226–241, 2014.

[70] A. Kroll. *Computional Intelligence: Eine Einführung in Probleme, Methoden und technische Anwendungen*. Oldenburg, 2nd edition, 2016.

[71] K. Kugler, B. Stahl, J. Reif, B. Lohmann, and F. Brodbeck. Zyklische Teamprozesse und Innovationsleistung - eine auf Fuzzy Logik basierte Modellierung wechselseitiger Beziehungen verschiedener Prozessphasen. In *Kongress der Deutschen Gesellschaft für Psychologie*, 2014.

[72] M. Kumagai and T. Ochiai. Development of a robot balanced on a ball - first report, implementation of the robot and basic control. *Journal of Robotics and Mechatronics*, 22:348–355, 2010.

[73] H. Kwakernaak and R. Sivan. *Linear optimal control systems.* Wiley Interscience, 1972.

[74] H. G. Kwatny, J.-E. T. Dongmo, B.-C. Chang, G. Bajpai, M. Yasar, and C. Belcastro. Aircraft accident prevention: Loss-of-control analysis: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20090030517.pdf. Technical report, American Institute of Aeronautics and Astronautics, 2009.

[75] W.H. Kwon and S.H. Han. *Receding horizon control: model predictive control for state models.* Springer, 2005.

[76] J. Latombe. *Robot motion planning.* Kluwer Academic, 1991.

[77] D. H. Lee and D. W. Kim. Relaxed LMI conditions for local stability and local stabilization of continuous-time Takagi-Sugeno fuzzy systems. *IEEE Trans. Syst., Man Cybern.*, 44(3):394–405, 2014.

[78] Z. Lendek, T. M. Guerra, R. Babuška, and B. De Schutter. *Stability analysis and nonlinear observer design using Takagi-Sugeno fuzzy models*, volume 262 of *Studies in Fuzziness and Soft Computing.* Springer, 2011.

[79] Z. Li and C. Yang. Neural-adaptive output feedback control of a class of transportation vehicles based on wheeled inverted pendulum models. *IEEE Trans. Control Syst. Technol.*, 20(6):1583–1591, 2012.

[80] D. Liberzon. *Switching in systems and control.* Birkhäuser, 2003.

[81] H. Liu and R. Fen. The sufficient condition of T-S fuzzy systems as universal approximators. In Y. Liu, G. Chen, and M. Ying, editors, *Fuzzy Logic, Soft Computing and Computational Intelligence*, pages 1082–1085. Springer, 2005.

[82] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proc. IEEE Symp. on Computer Aided Control Systems Design*, pages 284–289, 2004.

[83] B. Lohmann, K. J. Diepold, and B. Stahl. Analyse der Dynamik vernetzter Zyklen. In B. Vogel-Heuser, U. Lindemann, and G. Reinhart, editors, *Innovationsprozesse zyklenorientiert managen*, pages 63–76. Springer, 2014.

[84] B. Lohmann, B. Stahl, and K. J. Diepold. Systemtheoretische Grundlagen zyklengerechter Modellbildung. In B. Vogel-Heuser, U. Lindemann, and G. Reinhart, editors, *Innovationsprozesse zyklenorientiert managen*, pages 45–62. Springer, 2014.

[85] S. Macfarlane and E. A. Croft. Jerk-bounded manipulator trajectory planning: Design for real-time applications. *IEEE Trans. Automat. Contr.*, 19(11):42–52, 2003.

[86] P. Martin, S. Devasia, and B. Paden. A different look at output tracking: Control of a vtol aircraft. *Automatica*, 32:101–107, 196.

[87] MathWorks. How pattern search polling works: http://www.mathworks.de/de/help/gads/how-pattern-search-polling-works.html, May 2014.

[88] I. Maurović, M. Batorić, and I. Petrović. Explicit model predictive control for trajectory tracking with mobile robotics. In *Proc. IEEE/ASME Conf. on Advanced Intelligent Mechatronics*, pages 712–717, 2011.

[89] T. Nguyen and F. Jabbari. Output feedback controllers for disturbance attenuation with actuator amplitude and rate saturation. *Automatica*, 36:1339–1346, 2000.

[90] H. Noura, D. Theilliol, J.-C. Ponsart, and A. Chamseddine. *Fault-tolerant control systems: design and practical applications*. Springer, 2009.

[91] H. Ohtake, K. Tanaka, and H. O. Wang. Switching fuzzy controller design based on switching Lyapunov function for a class of nonlinear systems. *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, 36(1):13–23, 2006.

[92] E. Pellegrini, K. J. Diepold, R. Dessort, H. Panzer, and B. Lohmann. 3D-modeling of a robot balancing on a ball: http://mediatum.ub.tum.de/node?id=1081938. Technical report, Institute of Automatic Control, Technische Universität München, 2011.

[93] N. Peterfreund and Y. Baram. Convergence analysis of nonlinear dynamical systems by nested Lyapunov functions. *IEEE Trans. Automat. Contr.*, 43(8):1179–1184, 1998.

[94] K. B. Petersen and K. B. Petersen. The matrix cookbook, 2013.

[95] J. L. Pitarch and A. Sala andC. V. Ariño. Closed-form estimates of the domain of attraction for nonlinear systems via fuzzy-polynomial models. *IEEE Trans. Syst., Man Cybern.*, 44(4):526–538, 2014.

[96] M. Pivtoraiko, R. A. Knepper, and A. Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333, 2009.

[97] C. Plehn, J.Koch, K. J. Diepold, B. Stahl, B. Lohmann, G. Reinhart, and M. Zäh. Modeling and analyzing dynamic cycle networks in production planning. In *Proc. CIRP Global Web Conference on Production Engineering Research: Advancement beyond state of the art*, pages 149–154, 2014.

[98] N. Pletschen and K.J. Diepold. Nonlinear state estimation for suspension control applications: a Takagi-Sugeno kalman filtering approach. *Control Engineering Practice*, page in press: http://dx.doi.org/10.1016/j.conengprac.2016.05.013, 2016.

[99] A. Podelski and S. Wagner. Model checking of hybrid systems: From reachability towards stability. In J.P. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control (HSCC 2006)*, pages 507–521. Springer, 2006.

[100] A. Podelski and S. Wagner. A method and a tool for automatic verification of region stability for hybrid systems: http://domino.mpi-inf.mpg.de/internet/reports.nsf/c125634c000710d0c12560400034f45a /c4b734ea2697c2bdc1257296002a9d8e/$file/mpi-i-2007-2-001.pdf. Technical report, Max-Plack-Institute for Informatics, Saarbrücken, 2007.

[101] A. Podelski and S. Wagner. *Region stability proofs for hybrid systems*, volume 4763 of *Lecture Notes in Computer Science*, pages 320–335. Springer, 2007.

[102] S. Prajna, P. Parrilo, and A. Rantzer. Nonlinear control synthesis by convex optimization. *IEEE Trans. Automat. Contr.*, 49(2):310–314, 2004.

[103] C. Richter, A. Bry, and N. Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *Proc. International Symposium of Robotics Research*, 2013.

[104] A. Sala. On the conservativeness of fuzzy and fuzzy-polynomial control of nonlinear systems. *Annual Reviews in Control*, 33(1):48–58, 2009.

[105] H. Schulte and E. Gauterin. Input-to-state analysis of small wind turbines using LMI conditions. *at-Automatisierungstechnik*, 62(10):698–707, 2014.

[106] H. Sira-Ramírez and S.K. Agrawal. *Differentially flat systems.* Marcel Dekker, 2004.

[107] B. Stahl, K. J. Diepold, J. Pohl, J. Greitemann, C. Plehn, J.Koch, B. Lohmann, and G. Reinhart. Modeling cyclic interactions within a production environment using transition adaptive recurrent fuzzy systems. In *Proc. IFAC Conf. on Manufacturing Modelling, Management, and Control*, pages 2020–2025, 2013.

[108] G. Stein. Respect the unstable. *IEEE Control Systems Magazine*, 23(4):12–25, 2003.

[109] A. A. Stoorvogel and A. Saberi. Output regulation of linear plants with actuators subject to amplitude and rate constraints. *Int. Journal of Robust and Nonlinear Control*, 9:631–657, 1999.

[110] S. Summers, D. M. Raimondo, C. N. Jones, J. Lygeros, and M. Morari. Fast explicit nonlinear model predictive control via multiresolution function approximation with guaranteed stability. In *Proc. IFAC Symposium on Nonlinear Control Systems*, pages 533–538, Bologna, Italy, 2010.

[111] H. Surmann and M. Maniadakis. Learning feed-forward and recurrent fuzzy systems: A genetic approach. *Journal of Systems Architecture*, 47:649–662, 2001.

[112] T. Takei, R. Imamura, and S. Yuta. Baggage transportation and navigation by a wheeled inverted pendulum mobile robot. *IEEE Trans. Ind. Electron.*, 59(10):3985–3994, 2009.

[113] K. Tanaka and H. O. Wang. *Fuzzy control systems design and analysis: a linear matrix inequality approach.* John Wiley & Sons, 2001.

[114] S. Tarbouriech, G. Garcia, J. M. G. da Silva Jr., and I. Queinnec. *Stability and stabilization of linear systems with saturating sctuators.* Springer, 2011.

[115] S. Tarbouriech and M. Turner. Anti-windup design: an overview of some recent advances and open problems. *IET Control Theory and Appl.*, 3(1):1–19, 2009.

[116] V. Tsourapas, J. Sun, and A. Stefanopoulou. Incremental step reference governor for load conditioning of hybrid fuel cell and gas turbine power plants. *IEEE Trans. Contr. Syst. Technol.*, 17(4):756–767, 2009.

[117] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Application*, 19(2):499–533, 1998.

[118] S. Boyd L. Vandenberghe. *Convex optimization.* Cambridge Univ. Press, 2009.

[119] H. Wang, B. Chen, X. Liu, K. Liu, and C. Lin. Robust adaptive fuzzy tracking control for pure-feedback stochastic nonlinear systems with input constraints. *IEEE Trans. Syst., Man Cybern.*, 43(6):2093–2104, 2013.

[120] X.-P-Xie, Z.-W. Liu, and X.-L. Zhu. An efficient approach for reducing the conservatism of LMI-based stability conditions for continuous-time T-S fuzzy systems. *Fuzzy Sets Syst.*, 263:71–81, 2015.

[121] W. Xiang and J. Xiao. Switching PDC control for discrete-time T-S fuzzy systems: A membership function ranking approach. *Journal of the Franklin Institute*, 351(7):3526–3558, 2014.

[122] W. Xiang, J. Xiao, and M. N. Iqbal. h_∞ control for switched fuzzy systems via dynamic output feedback: Hybrid and switched approaches. *Communications in Nonlinear Science and Numerical Simulation*, 18(6):1499–1514, 2013.

[123] S.S.-D. Xu and Y.-K. Liu. Study of Takagi-Sugeno fuzzy-based terminal-sliding mode fault-tolerant control. *IET Control Theory and Applications*, 8(9):667–674, 2014.

[124] X. Xu and G. Zhai. Practical stability and stabilization of hybrid and switched systems. *IEEE Trans. Automat. Contr.*, 50(11):1897–1909, 2005.

[125] L. Zaccarian and A. R. Teel. *Modern anti-windup synthesis: control augmentation for actuator saturation.* Princeton Univ. Press, 2011.

[126] M. N. Zeilinger, C. N. Jones, and M. Morari. Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization. *IEEE Trans. Automat. Contr.*, 56(7):1524–1534, 2011.

[127] M. Zeitz. Differenzielle Flachheit: Eine nützliche Methodik auch für lineare SISO-Systeme. *at-Automatisierungstechnik*, 58(1):5–13, 2010.

[128] T. Zhang, G. Feng, H. Liu, and J. Lu. Piecewise fuzzy anti-windup dynamic output feedback control of nonlinear processes with amplitude and rate saturations. *IEEE Trans. Fuzzy. Systems*, 17(2):253–264, 2009.

[129] Y. Zhao and H. Gao. Fuzzy-model-based control of an overhead crane with input delay and actuator saturation. *IEEE Trans. Fuzzy. Systems*, 20(1):181–186, 2012.